Output-Only Techniques for Fault Detection

by

Adam John Brzezinski

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering)
in The University of Michigan
2011

Doctoral Committee:

Professor Dennis S. Bernstein, Co-Chair
Professor Jun Ni, Co-Chair
Professor Ilya V. Kolmanovsky
Professor Jeffrey L. Stein
Associate Professor Ella M. Atkins
Sunil L. Kukreja, NASA Dryden

Tiger got to hunt,
Bird got to fly;
Man got to sit and wonder, “Why, why, why?”

Tiger got to sleep,
Bird got to land;
Man got to tell himself he understand.

– Kurt Vonnegut, Jr., *Cat’s Cradle*

七転び八起き

– Japanese proverb

ACKNOWLEDGEMENTS

I would like to thank Prof. Dennis S. Bernstein and Prof. Jun Ni, my committee co-chairs, for their help and guidance. They have provided me with very valuable advice and direction; none of my work would have been possible without them. I would also like to thank Prof. Ilya V. Kolmanovsky, Prof. Jeffrey L. Stein, Prof. Ella M. Atkins, and Dr. Sunil L. Kukreja for serving on my doctoral committee and providing me with helpful suggestions on ways to improve my dissertation.

I am very grateful for my friends from the S.-M. Wu Manufacturing Research Center, especially Ahmad Almuhtady, Dr. Saumil Ambani, Kuen-Ren (Roland) Chen, Mo Chen, Wei Cheng, Prof. Deokki Choi, Prof. Dragan Djurdjanovic, Dr. Robert Dodde, Dr. Shichang Du, Shiming Duan, Dr. Jie Feng, Dr. Masahiro Fujiki, Xi Gu, Terri L. Gorowski, Kathleen S. Hoskins, Dr. Yan Jia, Li Jiang, Xiaoning Jin, Daniel D. Johnson, Dr. Hsin-Yu (Cindy) Kuo, Hoon Lee, Dr. Seungchul Lee, Dr. Yong Lei, Prof. Lin Li, Dr. Yongqing (Catcher) Li, Xinran Liang, Dr. Yi Liao, Dr. Jianbo Liu, Dr. Yang Liu, Carl S. McGill, Rifet Muminovic, Jae-Wook Oh, Chaoye Pan, Dr. Kwanghyun Park, Xianli (George) Qiao, Zheng Shen, Dr. David A. Stephenson, Li-Jung (Bruce) Tai, Dr. Jia Tao, Roberto Torres, Yong Wang, Chenjun Wei, Xin Weng, Grace L.-H. Wu, Li Xu, Li Xu, Xinwei Xu, Mingyi You, Hao Yu, Jing Yuan, Dr. Min Zhang, Dr. Jing Zhou, and Dr. Jie Zhu, as well as my other friends in Mechanical Engineering, especially Lindsay V. Allen, Dhanajay Anand, Julie DeFilippo, Jeffrey G. Fletcher, Serge L. L. H. Foo-Gregory, William S. Harrison III, Deepak Sharma, Janani Viswanathan, Seng K. Yeoh, and Ju-Il Yum. Your hard work and camaraderie

have helped me understand the value of dedication and effort.

I am very thankful for my friends in Professor Dennis S. Bernstein's lab, especially Asad A. Ali, Anthony M. D'Amato, Bojana Drincic, Matthew S. Fledderjohn, Prof. Jesse B. Hoagg, Matthew S. Holzel, Yan Jin, Kenneth S. Mitchell, Alexey V. Morozov, and Amy R. Wu, as well as my other friends in Aerospace Engineering, especially Serge F. Badiane, Rebecca L. Besser, Heejun Choi, Ryan D. Eubank, Dr. Eric D. Gustafson, Aaron Hoskins, Dr. Jennifer S. Hudson, Amir J. Matlock, Catherine McGhan, Dr. Amor Menezes, Ali Nasir, and Sara C. Spangelo. You have inspired and humbled me with your intelligence and insight.

I also greatly appreciate my friends from General Electric, especially Gerald Geverdt, Roger Lindle, Kevin W. Meyer, Li-Huan Peng, David T. Smither, Gary E. Sroka, and Howard Weaver, as well as my friends from the University of Cincinnati, especially Dr. Mohamed A. AbuAli, Patrick Brown, Yan Chen, Prof. Samuel H. Huang, Edzel R. Lapira, Prof. Jay Lee, David N. Siegel, and Tyler Skirtich. You have shown me the value of pragmatism and helped me to broaden my problem-solving perspective.

Finally, I would like to thank my family, especially my parents, John A. Brzezinski and Kathryn E. Brzezinski, my grandparents, Kenneth C. Macauley, Rosemary A. (DeMist) Macauley, John Brzeziński, and Helen (Madalińska) Brzezińska, and my wife, Makiko (Kawamura) Brzezinski. Words cannot express my gratitude for you. I hope I can someday support others as well as you have supported me.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

Appendix

CHAPTER I

Introduction

1.1 Background and Motivation

As sensing technology evolves, sensor costs decrease and sensing capability improves; hence, sensors are being installed in an increasing number of applications. Advances in data storage and computing capabilities also facilitate better and faster processing of sensor data. With the amount of sensor data and computing power available, new ways to process this data can therefore be developed to detect system faults.

Fault detection is essential to many disciplines [1, 2]. In aerospace engineering, structural health monitoring [3] and operational modal analysis [4] focus on detecting and localizing faults in wings, fuselages, and control surfaces. In manufacturing, process monitoring [5] and machine health monitoring [6, 7, 8] consider detection and characterization of anomalies in parts, machine tools, and material handling systems. In civil engineering, detection of cracks in bridges and buildings is widely researched [9, 10]. Mechanical engineering applications include engine fault detection [11] and pump fault detection [12].

Validated fault detection techniques provide multiple benefits. Downtime and faulty operation can be reduced by implementing fast, reliable fault detection. Maintenance costs and scrap can also be mitigated through fault detection. Furthermore, health and safety risks can be alleviated through automated fault detection, especially if it is applied to large structures that contain human occupants, such as aircraft and buildings.

In some applications, an analytical model for the system under study may be known [13]. In this case, model-based fault detection techniques may provide more accurate fault detection than output-only fault detection techniques [14]. However, an analytical model for the system under study is often unavailable or uncertain. In fact, output measurement data may be the only information known about the system. Furthermore, the excitation that induces the output data may be uncontrollable, uncommandable, unknown, or corrupted by noise and disturbances. Hence, techniques that use only output data for fault detection are needed.

1.2 Existing Techniques for Output-Only Fault Detection

1.2.1 Statistical Process Control

Statistical process control (SPC) techniques provide one way to detect faults by examining deviations of the output measurements from their nominal values. First, a control chart [15] is constructed using the output data. A Shewhart control chart [16] examines the raw output data, a cumulative sum (CUSUM) control chart [17] examines data that has been normalized by the mean and variance of all the data collected, and an exponentially-weighted moving average (EWMA) control chart [18] examines averaged data values in which the relative weight of each data point exponentially decreases with time. Second, a set of heuristic rules, such as the Westinghouse Electric Rules [19], is used to determine whether a fault has occurred because the data in the control chart is out-of-control. The Westinghouse Electric Rules state that the data in the control chart is out-of-control if (1) any point falls outside the 3σ limit, (2) two out of three consecutive points fall outside the 2σ limit, (3) four out of five consecutive points fall outside the σ limit, or (4) at least nine consecutive points fall on the same side of μ, where μ and σ are the mean and variance of the data in the control chart, respectively.

1.2.2 Pattern Recognition

Pattern recognition approaches for fault detection are typically divided into three main stages [20]. First, key characteristics, or features, are extracted from the output data. Second, a subset of the most relevant features are selected. Third, the values of the chosen features are combined together to produce an overall estimate of the condition of the system. A fault is detected if the estimated system condition does not correspond to a normal system condition.

Many features can be extracted from data, such as the mean and variance of the data [21]. Because some features provide better insight into whether a fault has occurred than others, a large number of time-domain, frequency-domain, and time-frequency-domain features are extracted from the data to ensure that no key information is lost [22]. Furthermore, application-specific features can be defined based on expert knowledge or observation of the data [23].

Fischer's criterion [24] is typically used to rank the relative importance of various features, but improved classification can be obtained by selecting the set of features that best span the feature space [25]. Hence, clustering and principal component analysis (PCA) can be applied [26]. However, because PCA separates the feature space using a linear combination of features, it is sub-optimal if the feature clusters cannot be separated using linear partitions. Hence, heuristic graph search techniques such as sequential forward selection have been developed to obtain a near-optimal set of features [27].

To classify a set of features, machine learning techniques are used to define boundaries between clusters of features. Boundaries can be linear (for example, using

Fisher's linear discriminant) or nonlinear (for example, using a support vector machine) [28] and are typically obtained by training using supervised learning techniques [29].

1.2.3 System Identification

Time-domain identification approaches can be used to estimate the parameters of a time-series model from the excitation to the output measurements [30, 14]. For example, estimates of the parameters $a_1, \dots, a_n, b_0, \dots, b_n$ of the linear time-series model

$$y(k) = a_1 y(k-1) + \cdots + a_n y(k-n) + b_0 u(k) + \cdots + b_n u(k-n)$$

can be used to detect faults, where $y(k)$ denotes the output measurement(s) at time step k and $u(k)$ denotes the excitation at time step k. For output-only identification, $u(k)$ is assumed to be a realization of a white random process [32].

Eigenstructure estimation approaches identify a linear state space model for the system [34, 35, 36]. For example, estimates of A and C of

$$x(k+1) = Ax(k) + n_1(k),$$

$$y(k) = Cx(k) + n_2(k),$$

can be used to detect faults, where $x(k)$ is the state of the system and $n_1(k)$ and $n_2(k)$ are white and uncorrelated. A related approach, which uses blind source separation to estimate the coefficients of the linear relationship between $y(k)$ and $x(k)$, can also be applied [37]. These approaches typically assume that the excitation is a realization of a white random process [33].

Frequency-domain identification techniques can be used to estimate the resonant frequencies and mode shapes of a structure [39]. For example, Fourier transforms can be used to compute the frequency spectrum of each output measurement, and modal characteristics inferred from the output spectra can be used to detect faults [38]. As with time-domain and eigenstructure estimation approaches, frequency-domain identification techniques typically assume that the excitation is a realization of a white random process [40].

In structural health monitoring, output-only system identification techniques are known as Operational Modal Analysis (OMA) [4]. In OMA, the dynamics between the excitation and output(s) are typically estimated, but the dynamics between sets of output measurements (transmissibilities) can also be estimated [41, 42]. If the excitation is sinusoidal, a frequency-domain approach can be used to identify the harmonic frequencies of the structure [43].

1.2.4 Limitations of Existing Output-Only Fault Detection Approaches

SPC techniques require training or expert knowledge to determine the threshold σ [44]. Furthermore, knowledge of the spectrum of the output data is required to design an SPC approach to minimize the number of missed detections and false alarms [15]. Finally, SPC techniques cannot detect changes in the dynamics of the system that do not result in a change in one of the first few statistical moments of the output data.

Feature extraction requires expert knowledge to define key metrics that characterize the data. Machine learning requires a large amount of training data and may need significant computational time to complete the training and classification steps [45, 46]. Furthermore, training must be completed for each excitation, and unexpected changes in the excitation can cause the output data to be incorrectly classified [47]. Finally, artificial neural networks (ANNs) require expert knowledge and/or extensive testing to determine the optimal number of layers, number of nodes, and types of activation functions in each layer [48].

Although many system identification approaches assume that the excitation signal is a realization of a white random process, this assumption cannot be verified unless the excitation can be measured. If the excitation signal is not a realization of a white random process, consistent parameter estimates can be achieved only for very specialized cases, namely, the white equation error and finite impulse response cases [49, 50]. Furthermore, disturbances and nonzero initial conditions can corrupt the output measurements and result in inaccurate parameter estimates [51, 52].

1.3 Research Objectives

The research presented in this thesis has three main objectives. First, we want to develop a technique to detect an abrupt change in a noisy signal in real-time. This technique should provide comparable or better performance than existing approaches in terms of detection speed, number of missed detections, and number of false alarms. Second, we want to define key features and choose a feature selection and classification approach that is able to classify multiple states accurately while while requiring short training and classification time. This technique should provide better performance than a benchmark approach using principal component analysis and a multi-layer perceptron. Third, we want to define a technique to detect changes in the dynamics of a linear system. This technique should require no knowledge or measurement of the excitation, allow an arbitrary nonzero and unknown initial system state, and degrade gracefully in the presence of output noise. Furthermore, this technique should be applicable to systems that have colored, non-unit-rank excitation as well as colored, correlated output noise.

1.4 Outline of the Dissertation

The remainder of this dissertation is organized as follows.

Chapter II explains the AMWPD algorithm for real-time abrupt change detection. The AMWPD algorithm calculates the mean and variance of the data in adjacent moving windows, compares the means of the data in the two windows, and returns a fault if the mean of the data in the most recent window differs from the mean of the data in the older window by more than $k\sigma_{\text{old}}$, where σ_{old} denotes the standard deviation of the data in the older window and $k > 0$ is chosen by experience or data-based training. We demonstrate techniques that reduce the number of operations required to compute the mean and variance of the data in the two windows, thereby reducing the computational requirements of the proposed algorithm. For a grinder-dresser crash detection application, the AMWPD algorithm is compared with existing techniques and shown to provide comparable detection speed and fewer false alarms. This chapter is based on work described in [53].

Chapter III discusses techniques for shaving tool wear classification (including tooth breakage detection) using indirect sensing. Feature extraction, feature dimension reduction, and classification are considered. PCA and modified tabu search (mTS) with long-term memory are compared for feature dimension reduction. A multi-layer perceptron (MLP) ANN is compared with a probabilistic neural network (PNN) for classification. The approach using modified TS and a PNN (mTS + PNN) is shown to achieve more accurate classification in less time than the approach using PCA and an MLP. This chapter is based on work described in [54, 55].

Chapter IV defines a pseudo transfer function (PTF) from one output measurement to another. The single-input-single-output (SISO) PTF order and relative degree are characterized, and proofs and numerical examples are provided to justify the results. SISO PTFs are estimated consistently in the presence of output noise using quadratically-constrained least squares (QCLS), a technique described in [50]. Furthermore, SISO PTFs are used to detect faults in a simulated example. This chapter is based on work described in [56, 57, 58].

Chapter V extends the work in Chapter IV by introducing multiple-input-multiple-output (MIMO) PTFs for systems with multiple excitations. Conditions for which a MIMO PTF is defined are presented, as well as an upper bound on the order of each entry of the MIMO PTF. A μ-Markov MIMO QCLS algorithm is developed and used to estimate the MIMO PTF consistently in the presence of output noise. This chapter is based on work described in [57, 59].

Finally, Chapter VI presents conclusions, contributions, and future work.

CHAPTER II

Real-Time Peak Detection in a Noisy Signal

2.1 Introduction

Recent research has focused on improving monitoring and control techniques for processes, tools, and machines to make manufacturing less expensive, more efficient, and safer. To accomplish this task, sensors can be used to collect information about the condition of the monitored system. Researchers have also developed techniques for processing sensor signals to extract features that highlight the system condition. Generally, the feature extraction methods can be classified as time domain, frequency domain, time-frequency domain, or model-based approaches [21].

Once characteristic features are extracted from the signal, various methods can be used to decide when the change in a feature or set of features necessitates maintenance action. Some decision-making strategies include thresholding [60], statistical process control (SPC) [15], and clustering based on pattern recognition [29].

Although this serial flow of information from sensor data to extracted features to decisions has provided significant improvements in manufacturing quality and efficiency, "more development work is needed to 'ruggedize' monitoring algorithms so that they can be used reliably on the shop floor" [21]. For example, although time-frequency signal processing using wavelets provides insight into the system condition (see [61] and [62]), the associated computational requirements makes this approach infeasible for some embedded applications. Furthermore, the computational time required for time-frequency signal processing can limit its application in real-time monitoring, where processing time must be minimized.

Hence, we develop an algorithm for peak detection that includes signal processing and decision-making to minimize computational requirements and computing time. The proposed algorithm is applied to the signal collected from a horsepower sensor on a dresser spindle motor to detect contact between a dresser and grinder in real-time. If contact is detected, the proposed algorithm provides feedback to stop machine motion before damage occurs. The performance of the proposed algorithm is compared with other contact detection methods and shown to provide fewer false alarms than existing approaches.

The rest of this chapter is organized as follows. Section 2.2 provides details on existing signal processing and decision-making methods. Section 2.3 explains the proposed algorithm. Section 2.4 discusses implementation of the proposed algorithm in an industrial production environment and compares the results of the proposed algorithm with existing techniques for peak detection. Finally, Section 2.5 presents a summary and some future research directions.

2.2 Signal Processing and Decision-Making Methods

Time domain signal processing methods extract statistical characteristics from the signal; these characteristics are then used to determine whether a change in the signal has occurred [63]. The main advantage of time domain techniques is that they generally require less computational time than frequency domain or time-frequency domain signal processing methods. However, time domain methods may be sensitive to noise and do not provide as much insight into the process dynamics as other types of signal processing methods.

Frequency domain signal processing methods extract information about the spectrum of the sensor data using Discrete Fourier Transforms (DFTs) or Hilbert Transforms [64]. Key spectral content is reflected in characteristic features such as the NA4 index [23]. Frequency domain techniques require more computational time than time domain techniques but may provide additional insight into the properties of the signal, for example, by estimating natural frequencies. However, frequency domain methods do not provide an estimate of the time when a specific frequency component begins or ends, which could help users pinpoint the time when a change in the system characteristics began.

Time-frequency domain signal processing methods extract features present in both the time and frequency domains using short-time Fourier transforms (STFT) [65], wavelet transforms [66], or a time-frequency kernel such as the Wigner-Ville [67] or Choi-Williams [68] distribution. Time-frequency domain methods describe the signal's characteristics more completely than the time domain or frequency domain methods, but time-frequency domain methods require significant computational time. Hence, it is often infeasible to apply time-frequency signal processing methods for embedded, real-time fault detection.

Model-based signal processing methods build a model for the nominal system dynamics using expert knowledge or input-output data [13]. Sensor data is then used to construct a model for the present system dynamics; the present system dynamics are compared to the nominal dynamics and a fault is detected if the estimated dynamics differ significantly from the nominal dynamics. ARMA model residuals [69], parity equations [70], virtual sensing [71], and parameter estimation [72] are some model-based signal processing methods. For systems with unknown nonlinear dynamics, construction of a model for the system dynamics can be time-consuming and computationally intensive, which limits the applicability of model-based signal processing techniques in embedded, real-time fault detection.

Decision making methods for abrupt change detection are either parametric or non-parametric [73]. Parametric change detection techniques consider changes in estimated parameters or features, while non-parametric change detection techniques consider changes in parameters and model structure simultaneously. If sensor noise is known or estimated, parametric approaches are more accurate than non-parametric approaches [74]. However, a Shewhart or control chart [15], which is a type of parametric change detection technique, cannot differentiate between abrupt and gradual changes. If the sensor output gradually changes over time due to degradation or process changes [75], a control chart may falsely indicate that an abrupt change has occurred. Hence, a moving window, such as an Exponentially Weighted Moving Average (EWMA), a cumulative sum (CUSUM), or a finite moving average control chart [19], can be introduced so that only abrupt changes in the signal are detected. However, these algorithms compare a single incoming data point with the baseline data to determine whether an abrupt change has occurred. If the incoming data is noisy, direct application of these methods can yield many false alarms.

2.3 Description of Proposed Algorithm

In the proposed Adjacent Moving Window Peak Detection (AMWPD) algorithm, we assume that no abrupt changes occur in the signal until two adjacent first-in-first-out buffers are filled with data. Hence, we compare the new data in the small buffer SB with the old data in the large buffer LB. For $K \in \{L, S\}$, the mean of the data in buffer K at iteration i is given by μ_{KB_i}, while the standard deviation of the data in buffer K at iteration i is given by σ_{KB_i}. We detect an abrupt change if $\mu_{SB_i} \gg \mu_{LB_i}$. We note that the presence of LB reduces the sensitivity of the proposed approach to false detection of gradual changes. We apply the proposed approach to a sample time series in Figure 2.1.

For each iteration i of the AMWPD algorithm, the newest sensor measurement y is assigned to first element of SB, while the last element of SB in iteration $i - 1$ is

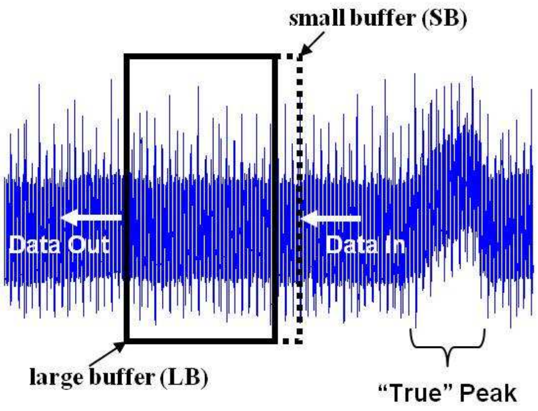


Figure 2.1: Illustration of the two adjacent buffers in the Adjacent Moving Window Peak Detection (AMWPD) algorithm.

assigned to the first element of LB. Hence,

$$SB_i[1] = y \tag{2.1}$$

and

$$LB_i[1] = SB_{i-1}[N_{SB}], \tag{2.2}$$

where the number in square brackets denotes the element of the associated buffer, N_{KB} is the number of data points in buffer KB, and $K \in \{L, S\}$.

N_{SB} and N_{LB} are chosen to maximize detection accuracy and robustness. If N_{SB} is too small, the algorithm will not be sufficiently robust to noise; conversely, if N_{SB} is too large, the algorithm will take too long to detect an abrupt change in the signal. To ensure that N_{SB} and N_{LB} are properly chosen, we move a test data set through a single buffer and choose N_{SB} to be the length of the shortest buffer whose mean and variance do not change significantly as the data is passed through the buffer. We choose $N_{LB} > N_{SB}$ but small enough to reduce computation time. The choices of N_{SB} and N_{LB} for the dresser contact application are discussed in Section 2.4.2.

Once both buffers are full with data, μ_{SB_i}, μ_{LB_i}, and σ_{LB_i} are given by

$$\mu_{KB_i} \triangleq \frac{1}{N_{KB}} \sum_{q=1}^{N_{KB}} KB_i[q] \tag{2.3}$$

and

$$\sigma^2_{KB_i} \triangleq \frac{1}{N_{KB}} \sum_{q=1}^{N_{KB}} \left(KB_i[q] - \mu_{KB_i}\right)^2, \tag{2.4}$$

where $q \in \{1, \ldots, N_{KB}\}$ and $K \in \{L, S\}$.

When a new data point y enters SB, (2.3) implies that, for time $i+1$,

$$\mu_{SB_{i+1}} = \frac{y}{N_{SB}} + \frac{1}{N_{SB}} \sum_{q=1}^{N_{SB}-1} SB_i[q]. \tag{2.5}$$

For time i, (2.3) also implies that

$$\frac{1}{N_{SB}} \sum_{q=1}^{N_{SB}-1} SB_i[q] = \mu_{SB_i} - \frac{SB_i[N_{SB}]}{N_{SB}}. \tag{2.6}$$

Substituting (2.6) into (2.5), we have

$$\mu_{SB_{i+1}} = \frac{y - SB_i[N_{SB}]}{N_{SB}} + \mu_{SB_i}. \tag{2.7}$$

Similarly, from (2.2), we have

$$\mu_{LB_{i+1}} = \frac{SB_i[N_{SB}] - LB_i[N_{LB}]}{N_{LB}} + \mu_{LB_i}. \tag{2.8}$$

Hence, the number of operations to calculate the mean is reduced from N_{AB}-1 additions and 1 division to 1 addition, 1 subtraction, and 1 division.

The standard deviation of the large buffer at time $i+1$ is also updated dynamically.

For time $i+1$, (2.4) implies

$$\sigma^2_{LB_{i+1}} = \frac{\left(SB_i[N_{SB}] - \mu_{LB_{i+1}}\right)^2}{N_{LB}} + \frac{\left(\sum_{q=1}^{N_{LB}-1} LB_i[q] - \mu_{LB_{i+1}}\right)^2}{N_{LB}}. \tag{2.9}$$

For time i, (2.4) also implies

$$\frac{\left(\sum_{q=1}^{N_{LB}-1} LB_i[q] - \mu_{LB_i}\right)^2}{N_{LB}} = \sigma^2_{LB_i} - \frac{\left(SB_i[N_{SB}] - \mu_{LB_i}\right)^2}{N_{LB}}. \tag{2.10}$$

Next, we assume $\mu_{LB_{i+1}} \approx \mu_{LB_i}$. This approximation is justified because we assume that the mean of the signal in LB does not change significantly as each new data point enters SB. Hence, (2.9) and (2.10) imply

$$\sigma^2_{LN_{i+1}} = \frac{\left(SB_i[N_{SB}] - \mu_{LB_i}\right)^2 - \left(LB_i[N_{LB}] - \mu_{LB_i}\right)^2}{N_{LB}} + \sigma^2_{LB_i}. \tag{2.11}$$

Hence, the number of operations required to calculate the standard deviation is reduced from N_{LB} subtractions, N_{LB} multiplications, N_{LB}-1 additions, 1 division, and 1 square root to 3 subtractions, 2 multiplications, 1 addition, 1 division, and 1 square root.

We set a threshold so that, if the mean of SB is significantly larger than the mean of LB, an abrupt change is detected. The abrupt change detection criterion is given by

$$Peak = \begin{cases} 0, & \mu_{SB_{i+1}} < \mu_{LB_{i+1}} + k\sigma_{LB_{i+1}}, \\ 1, & else, \end{cases} \tag{2.12}$$

where k is the user-defined threshold that determines how large the change must be in order to be detected. If the data is normally distributed, $k \approx 3$ is a common threshold choice based on the size of the confidence interval needed to describe normal process variations [76, 77]. The choice of k for the dresser contact detection application is further discussed in Section 2.4.2.

A flowchart of the AMWPD algorithm is shown in Figure 2.2. Although the AMWPD algorithm is used to detect changes in mean, a similar approach could be used to detect changes in variance.

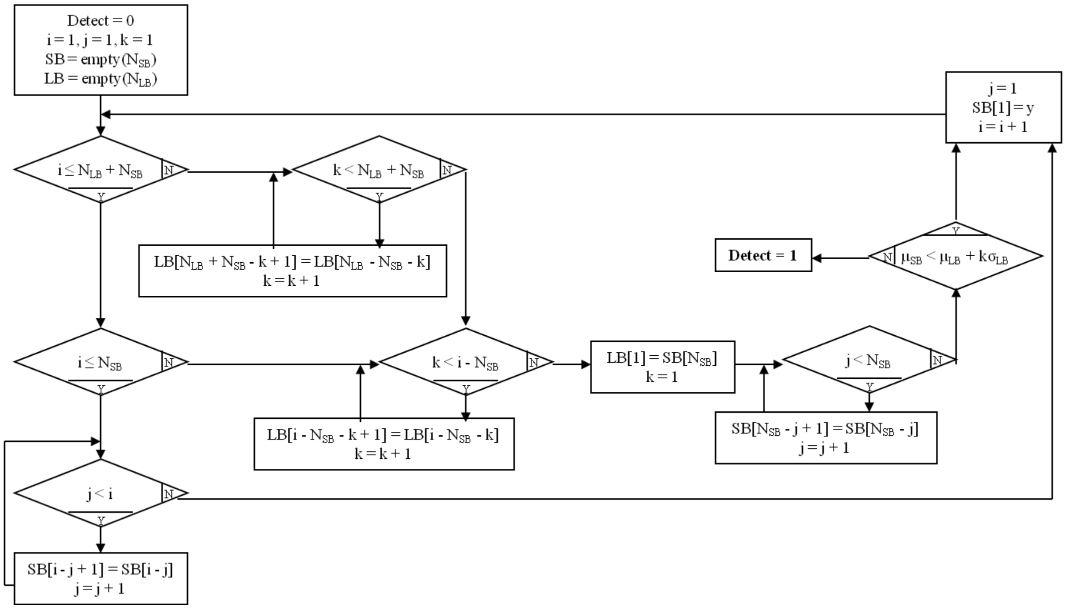


Figure 2.2: Flowchart of the AMWPD algorithm.

2.4 Industrial Application

2.4.1 Application Description

Grinding is an abrasive finishing process that can produce smooth surfaces with tight geometric tolerances while maintaining high material removal rates [78]. To facilitate grinding, the dressing process removes dull grains, exposes new, sharp grains, and re-shapes the grinding wheel. Because improper dressing can cause the grinding wheel to fail catastrophically, the dressing process must be carefully monitored.

Recent advances in numerical control, probing, and sensing have improved the dressing process [79, 80]. However, proposed techniques for detecting dresser contact using Acoustic Emission (AE) sensors are prohibitively expensive for shop-floor implementations and require extremely high data sampling rates [81, 82]. Hence, other sensors and processing techniques must be investigated to detect dresser-grinder contact.

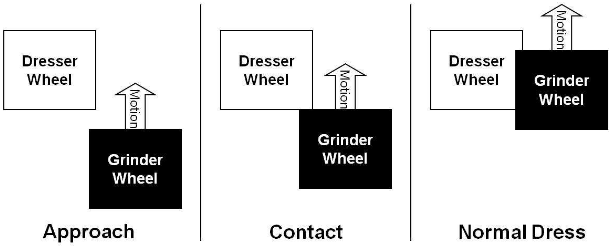


Figure 2.3: A schematic drawing of the dressing process.

We apply the AMWPD algorithm to detect contact between a cylindrical dresser and 4-axis cylindrical grinder. The AMWPD algorithm interfaces with the machine controller to immediately stop grinder motion if improper contact occurs between the grinder and dresser to ensure that catastrophic damage is avoided. A schematic drawing of the proper dressing process is shown in Figure 2.3, while a drawing of some of the possible dresser failure modes is shown in Figure 2.4.

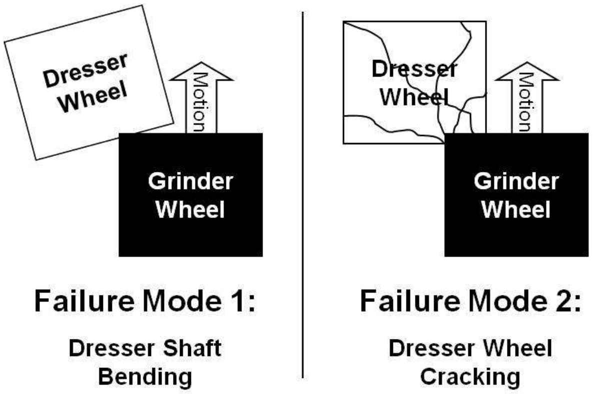


Figure 2.4: Two possible dresser failure modes.

A picture of the grinder where the data collection, data processing, and decision making system is installed is shown in Figure 2.5. Data from a horsepower sensor is used to determine whether contact between the dresser and grinder has occurred. Because the horsepower sensor only needs access to the power cables on the spindle motor, this sensor is very unintrusive to the process [79]. Other types of sensors used for dresser contact detection, like AE [81] or force [83] sensors, must be installed very close to or embedded in the grinding wheel and therefore may adversely affect the grinding process. Furthermore, the horsepower sensor is less expensive and requires a much lower sampling rate than other sensor types. Hence, although AE sensors can detect grinder-dresser contact faster than horsepower sensors [84], we use a horsepower sensor to detect grinder-dresser contact.

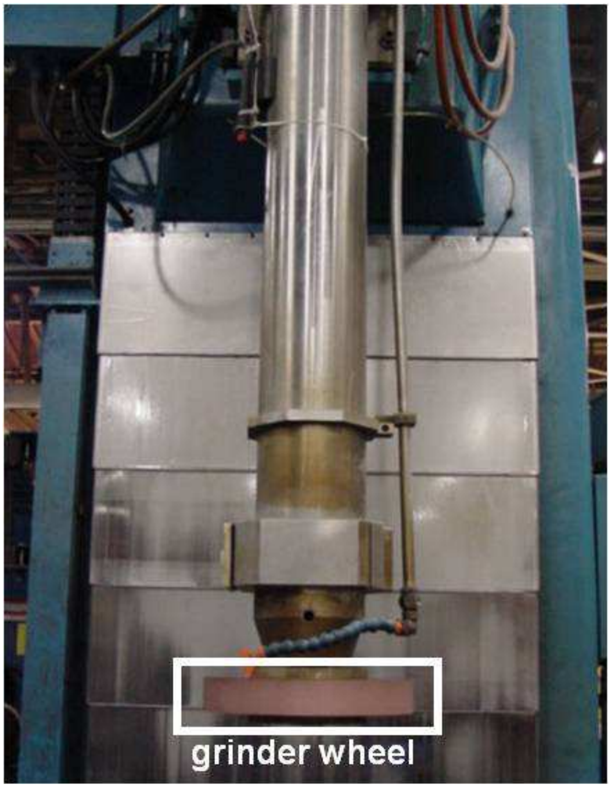


Figure 2.5: Picture of the grinder system where the AMWPD algorithm is applied.

The hardware for this system consists of an analog input module, a solid-state relay output module, and a real-time controller. To avoid aliasing, the signal from

k	N_{LB}	N_{SB}	Missed Detections	False Alarms
0.55	500	100	0/21	0
0.45	500	100	0/21	2
0.65	500	100	4/21	0
0.55	250	100	1/21	0
0.55	500	50	0/21	10
0.55	500	150	3/21	0

Table 2.1: Dependence of the AMWPD algorithm on N_{SB}, N_{LB}, and k.

the horsepower sensor is sampled at 100 Hz (the sensor has a time constant of 0.05 seconds). The AMWPD algorithm is embedded to facilitate real-time operation. If contact is detected, the output module closes the solid state relay. The machine control, which monitors the solid state relay, stops the grinder motion if the solid state relay is closed. The real-time controller is connected to the local intranet so that engineers can check the performance and output of the algorithm.

2.4.2 Application Results

We evaluate the performance of the AMWPD algorithm for various choices of N_{SB}, N_{LB}, and k in Table 2.1. To obtain the results in Table 2.1, we apply the AMWPD algorithm to analyze signals collected from a dressing process in which the grinder contacts the dresser 21 times. By increasing k and holding all other parameters constant, we see in Table 2.1 that the number of false alarms decreases while the number of missed detections increases.

Hence, the parameter choices in the first row of Table 2.1 provide sufficiently low rates of missed detections and false alarms. With this choice of parameters, a peak is detected less than 0.5 seconds after it occurs, as shown in Figure 2.6.

As Figure 2.7 illustrates, the motion of the grinder is stopped less than 0.55 seconds after contact occurs. During this time, the grinding wheel does not feed far enough into the dresser to damage the dresser or grinder wheel.

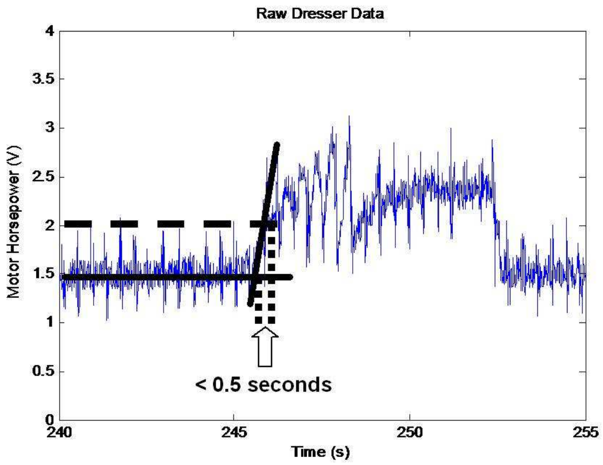
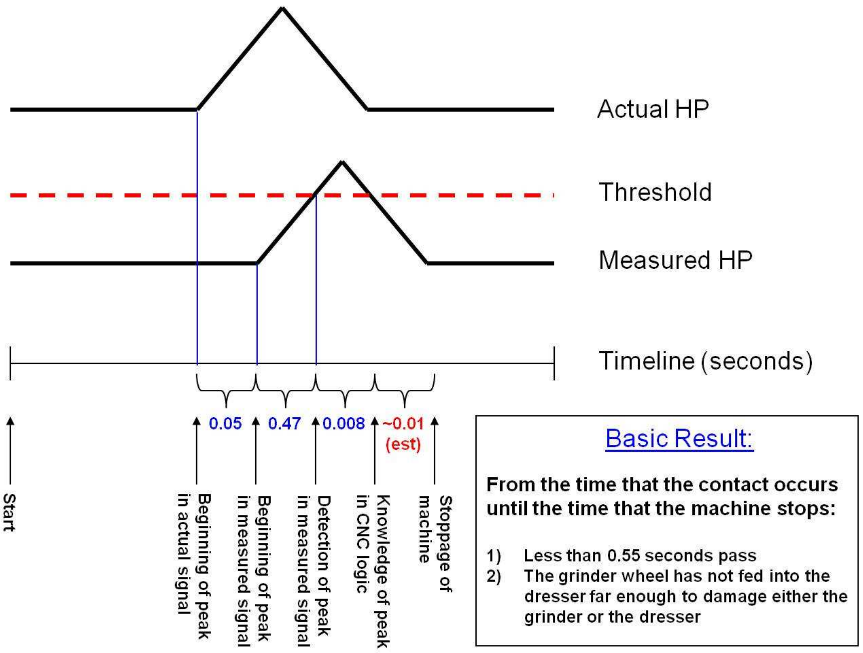


Figure 2.6: Illustration of the peak detection time of the AMWPD algorithm.


Actual HP
Threshold
Measured HP
Timeline (seconds)
0.05
0.47
0.008
~0.01
(est)
Start
Beginning of peak in actual signal
Beginning of peak in measured signal
Detection of peak in measured signal
Knowledge of peak in CNC logic
Stoppage of machine

Basic Result:

From the time that the contact occurs until the time that the machine stops:

1) Less than 0.55 seconds pass
2) The grinder wheel has not fed into the dresser far enough to damage either the grinder or the dresser

Figure 2.7: Schematic drawing characterizing the time required for the grinder to stop moving after the grinding wheel has touched the dresser.

2.4.3 Algorithm Comparison

We compare the speed and accuracy of the AMWPD algorithm with the performance of existing techniques for abrupt change detection. We use the raw dresser horsepower data shown in Figure 2.8. We note that two large peaks are present in the signal.

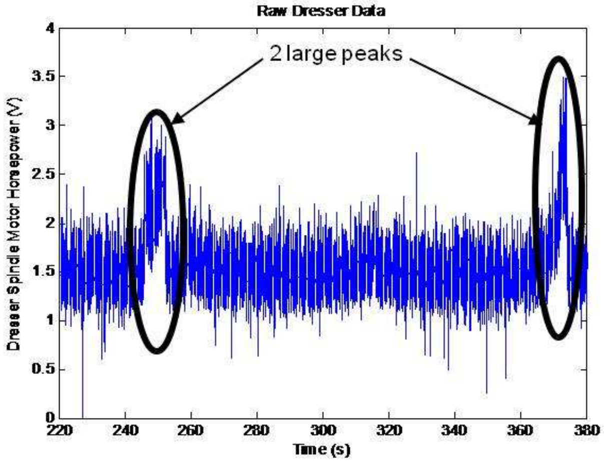


Figure 2.8: Unfiltered dresser horsepower data.

When the data is passed through a third-order Butterworth filter with a cutoff frequency of 2 Hz, we see that the signal actually contains 4 peaks, as shown in Figure 2.9. In the raw data signal in Figure 2.9, the two small peaks cannot be distinguished from the signal baseline due to the overwhelming presence of noise in the signal. Hence, these two small peaks could be missed by a peak detection algorithm.

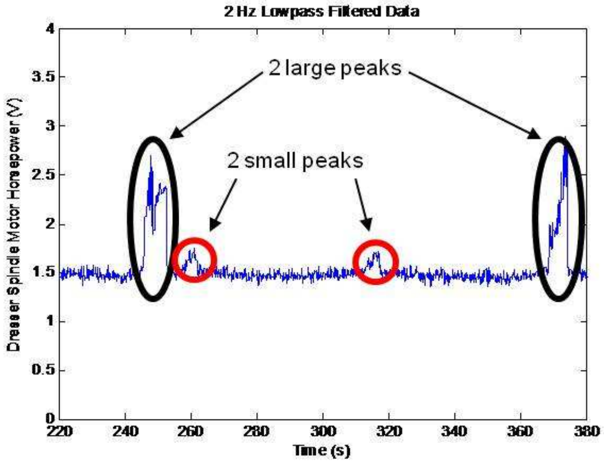


Figure 2.9: Filtered dresser horsepower data.

The AMWPD algorithm is compared with an online Shewhart control chart [16,

Algorithm	λ	k	N_{SB}	N_{LB}	Missed Detections	False Alarms	Avg. detection delay (s)
AMWPD Alg.		0.55	100	500	0/4	0	0.47
Shewhart		4		500	0/4	17	0.78
EWMA	0.025	4		500	0/4	9	0.22

Table 2.2: Comparison of AMWPD algorithm with online Shewhart control chart.

85] as well as an online EWMA control chart [18]. The EWMA control chart compares each new data point with a weighted average of past data, where recent data has higher relative weight than old data. Because the weighted average smooths the old data, the EWMA control chart is more robust to noise than the Shewhart control chart. Furthermore, the EWMA control chart gives more importance to the most recent measurements, which provides faster detection speed than the Shewhart control chart.

We use the data in Figure 2.8 to compare the performance of the AMWPD algorithm with the performance of the Shewhart and EWMA detection methods. The optimal values of k and λ for the Shewhart and EWMA detection methods are obtained by exhaustively searching for the sets of parameters that provide the minimum number of false alarms and missed detections. Using the optimal parameter values, the performance of the three algorithms is compared in Table 2.2.

2.4.4 Discussion

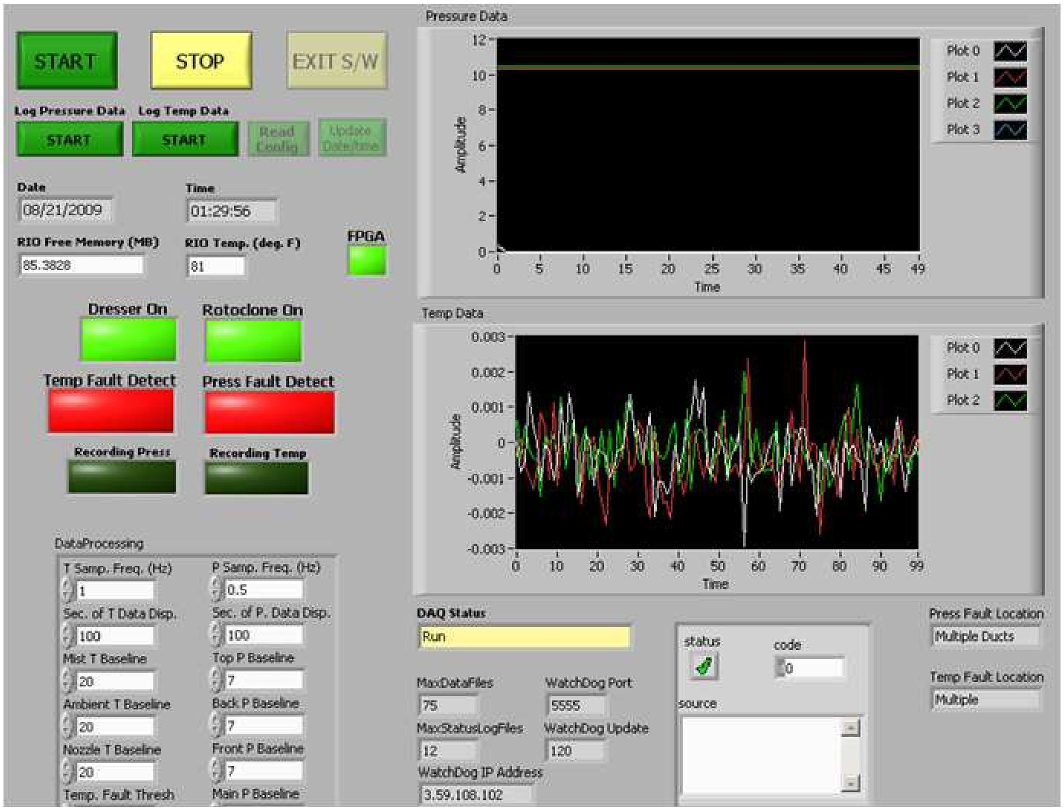
Based on the results shown in Table 2.2, the AMWPD algorithm provides fewer false alarms (Type II errors) than the Shewhart and EWMA algorithms. Although all the algorithms can be tuned to provide zero missed detections (Type I errors), only the AMWPD algorithm simultaneously provides zero missed detections and zero false alarms. Furthermore, although the EWMA algorithm provides a faster response time than the AMWPD algorithm, the AMWPD algorithm has a faster response time than the Shewhart algorithm.

We note that λ and k are typically chosen to be 0.2 and 3, respectively, to provide minimal false alarm and missed detection rates [18, 19]. However, Table 2.2 shows that the false alarm and missed detection rates for this application are minimized by unconventional choices of λ and k. The nominal λ and k may differ from expected values because the sensor noise may not be white, normally distributed, or stationary.

2.5 Conclusions and Future Work

This chapter has presented a new algorithm for real-time dresser contact detection using a horsepower sensor connected to the dresser spindle motor of an industrial grinder. The proposed AMWPD algorithm provides advantages over existing methods for abrupt change detection in terms of detection speed, reliability, and robustness. Furthermore, the AMWPD algorithm has been implemented and validated for dresser contact detection on an industrial grinding machine. A picture of the interface to the AMWPD algorithm, which is embedded using a LabVIEW real-time operating system, is provided in Figure 2.10.

The next phase of this research will be to leverage the results of the AMWPD algorithm and hardware interface. Hence, the robustness of the AMWPD algorithm will be improved and automatic methods for choosing k, N_{LB}, and N_{SB} will be investigated so that the AMWPD algorithm can be applied to prevent crashes on similar grinding machines. Also, additional work will be done to determine how well the AMWPD algorithm can detect peaks in signals from other manufacturing applications such as drilling, milling, and turning.



Date: 08/21/2009

Time: 01:29:56

RIO Free Memory (MB): 85.3828

RIO Temp. (deg. F): 81

T Samp. Freq. (Hz): 1

P Samp. Freq. (Hz): 0.5

Sec. of T Data Disp.: 100

Sec. of P. Data Disp.: 100

Mist T Baseline: 20

Top P Baseline: 7

Ambient T Baseline: 20

Back P Baseline: 7

Nozzle T Baseline: 20

Front P Baseline: 7

Temp. Fault Thresh

Main P Baseline

DAQ Status: Run

code: 0

MaxDataFiles: 75

WatchDog Port: 5555

MaxStatusLogFiles: 12

WatchDog Update: 120

WatchDog IP Address: 3.59.108.102

Press Fault Location: Multiple Ducts

Temp Fault Location: Multiple

Figure 2.10: User interface for embedded AMWPD algorithm.

CHAPTER III

Multi-State Classification Using Heuristic Feature Selection

3.1 Introduction

Shaving is a gear finishing process which reduces gear tooth surface roughness and dimensional inaccuracies [86]. During the shaving process, the tool engages the workpiece so that the tool teeth mesh with the workpiece teeth and the tool drives the rotation of the workpiece. Because of a small angle between the tool and workpiece axes of rotation, the sides of the tool teeth press against the sides of the workpiece teeth as the tool and workpiece rotate. The sides of the tool teeth contain many sharp ridges, as shown in Figure 3.1, which remove workpiece material through the cutting and forging processes. Although researchers have explored various aspects of the shaving process, including offline estimation of shaving cutting tool wear using optical inspection techniques [87], mathematical modeling of the shaving process to reduce machining errors [88], and new cutter designs to improve workpiece surface finish [89], only one previous work addresses shaving tool wear classification using indirect measurement techniques [54]. However, the approach in [54] provides low classification accuracy and does not consider the evolution of tool wear for a single tool; instead, different tools with different wear states are considered.

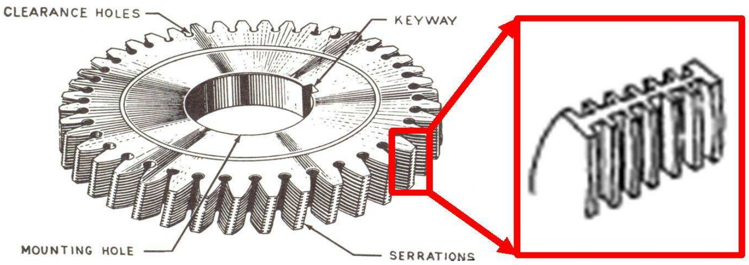


Figure 3.1: Shaving cutting tool with cutting edges enhanced in inset.

Shaving tool wear classification using indirect measurements, such as accelerometer signals, is difficult for various reasons. First, the cutting force, which has been shown to be correlated with tool wear in turning, drilling, and milling [90], is very small for the shaving process. Second, since multiple tool and workpiece teeth are engaged at any instant during the shaving process, the change in the cutting force due to the breakage or wear of a single ridge on a single tooth is extremely small. Third, the indirect sensor signal may only contain a small contribution from the cutting force due to the long transmission path from the cutting force to the sensor. However, because shaving tools are very expensive, fast and accurate shaving tool wear classification can provide significant maintenance cost savings.

In this chapter, we employ a standard approach for tool wear classification, which consists of data collection, feature extraction, feature selection, and classification [20]. We extract many well-known features in the time, frequency, and joint time-frequency domains, including root mean square [21], NA4 index [91], and time-frequency entropy [92]. Because the key features required for shaving tool wear classification are unknown, we also define new features, discussed in Section 3.2.3, which are designed to improve the classification result. To select key features and obtain a classification result, we employ a recently-developed heuristic technique [93]. For some applications, the heuristic technique developed in [93] has been shown to have higher accuracy and lower computational cost than conventional approaches, such as sequential forward/backward selection [27] and genetic algorithms [94]. However, the heuristic technique developed in [93] has never been applied to estimate tool wear condition.

The remainder of this chapter is organized as follows. Section 3.2 explains the methods and techniques used to conduct the experiments and subsequent data analysis. Section 3.3 provides our results and discussion on tool tooth breakage detection as well as tooth wear estimation. Finally, Section 3.4 provides conclusions.

3.2 Methodology

3.2.1 Experimental Setup

We conducted two experiments on shaving tool wear classification. First, we consider 5 different shaving cutter tools. Of these 5 tools, one tool is freshly re-ground. Another tool has an intermediate wear condition (given a part-based re-grind schedule to re-grind after X workpieces are machined, this tool has machined X/2 workpieces). A third tool has an advanced wear condition (it has machined X workpieces since the previous re-grind). The fourth and fifth tools have multiple broken teeth.

Second, we conduct a run-to-failure test on a freshly sharpened shaving tool with no broken teeth. Data is collected during the processing of three workpieces at the beginning of every shift, which corresponds to an interval of about 200 workpieces cut between measurements. As teeth break, the measurement interval is reduced so that there are only about 50 workpieces cut between measurements. The run-to-failure test is terminated when the tool has 11 total broken teeth and 3 broken teeth adjacent to each other (as shown in Figure 3.2).

Three-axis accelerometers are installed in locations 1 and 2 on the shaving machine shown in Figure 3.3. Accelerometers were chosen for this application because they are simple to install, do not affect the process under study, and are relatively inexpensive. Furthermore, accelerometers have been shown to provide good sensitivity when used

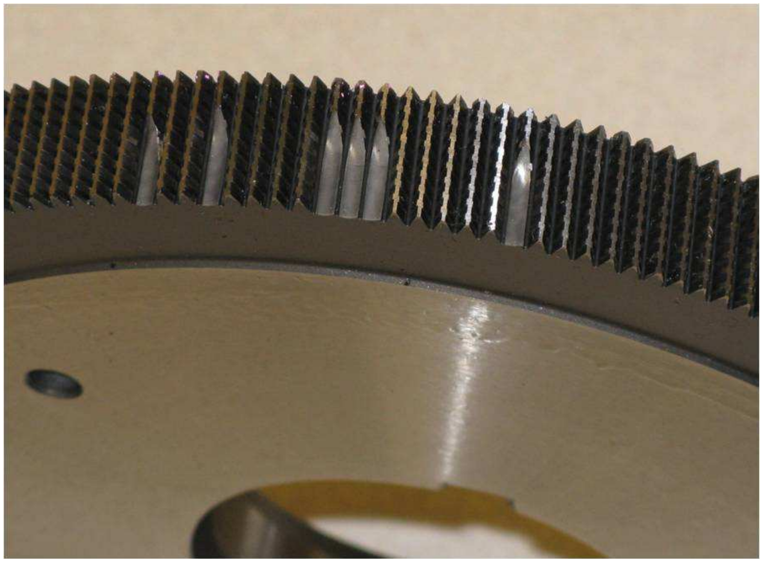


Figure 3.2: Shaving tool with broken teeth.

for gearbox CM [95].

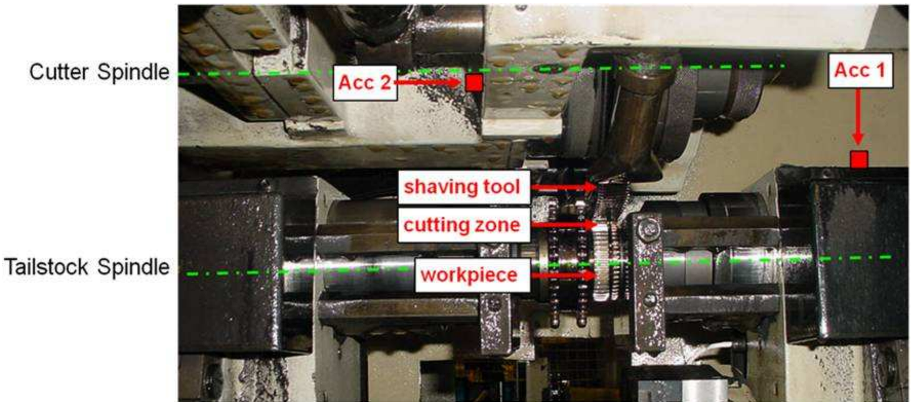


Figure 3.3: Accelerometer locations on machine.

The shaving process, which lasts roughly 30 seconds, can be separated into three key steps, or “cuts.” During the first cut, the shaving cutting tool rotates clockwise and engages the workpiece. During the second cut, the shaving cutting tool rotates counter-clockwise and feeds into the workpiece. During the third cut, the shaving cutting tool rotates clockwise, performs a finishing operation, and retracts from the workpiece. Accelerometer data is sampled at 25 kHz throughout the shaving process.

3.2.2 Data Pre-Processing

A total of 6 time series are obtained for each shaving cutting cycle (one time series for each axis of each accelerometer). A sample of one time series is shown in Figure 3.4.

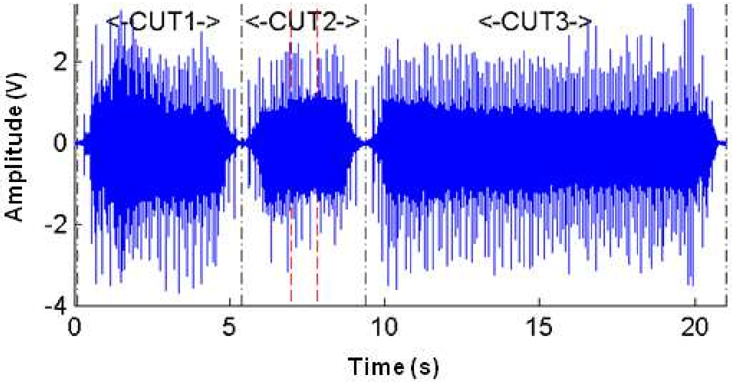


Figure 3.4: Sample accelerometer data from shaving one workpiece.

As labeled in Figure 3.4, the time series is separated into three "cuts." To estimate tool condition, the data collected during one cut in Figure 3.4 is examined independently from the data collected during another cut. Hence, data from Cut 2 is only compared to data from Cut 2, not data from Cut 1 or Cut 3. This procedure ensures that only data collected under the same cutting conditions is compared.

We only present the results of analysis of data from Cut 2 in this study. Data from Cut 2 is chosen because, unlike Cut 1 and Cut 3, Cut 2 does not involve any disengagement of the tool and workpiece. Note that analysis of the data from Cut 1 or Cut 3 does not yield better classification results than analysis of the data from Cut 2.

3.2.3 Feature Extraction

We extract the 18 metrics listed in Table 3.1 from each segment of data from Cut 2. Since we use two 3-axis accelerometers to collect vibration data, a feature consists of a metric extracted from data from one axis of one accelerometer. Hence, we extract

Time Domain	Freq. Domain	Time-Freq. Domain
Mean	MF1 (Mag. of Meshing Freq.)	EtrpSTFT (Entropy of STFT)
PPV (Peak-to-Peak Value)	MF2 (1st Overtone of MF1)	EtrpCW (Entropy of Choi-Williams Kernel)
RMS (Root Mean Square)	MF3 (2nd Overtone of MF1)	EtrpWavelet (Entropy of Wavelet Kernel)
CF (Crest Factor)	MF4 (3rd Overtone of MF1)	
Kurtosis	SumMF1 (Sideband Mag. of MF1)	
ModKurt (Modified Kurtosis)	SumMF2 (Sideband Mag. of MF2)	
RecSk (Rectified Skewness)	SumMF3 (Sideband Mag. of MF3)	
Entropy	SumMF4 (Sideband Mag. of MF4)	

Table 3.1: Variables used for breakage detection and tool condition estimation.

a total of 108 features (18 metrics, 3 axes, and 2 sensors) from the vibration data. The feature labeling convention is $Metric/Axis/Sensor$. For example, the feature $RMS/Y/2$ corresponds to the Root Mean Square of data collected from the Y-axis of Sensor 2.

Definitions of many of the metrics in Table 3.1 are provided in [21]. For completeness, we provide definitions of these metrics. Let $\{x(k)\}_{k=1}^{l}$ denote a time series of measurement data. Then the mean of x is given by

$$\mu \triangleq \frac{1}{l}\sum_{i=1}^{l} x(i), \tag{3.1}$$

the peak-to-peak value of x is given by

$$PPV \triangleq \max_i(x(i)) - \min_i(x(i)), \tag{3.2}$$

the root mean square (or standard deviation) of x is given by

$$RMS \triangleq \sqrt{\frac{1}{l}\sum_{i=1}^{l} (x(i)-\mu)^2}, \tag{3.3}$$

the kurtosis of x is given by

$$Kur \triangleq \frac{\frac{1}{l}\sum_{i=1}^{l}(x(i)-\mu)^4}{RMS^4}, \tag{3.4}$$

and the entropy of x is given by

$$Entropy \triangleq -\sum_{i=1}^{l}\text{Prob}(x(i))\text{log}(\text{Prob}(x(i))). \tag{3.5}$$

The sideband magnitude of the N^{th} harmonic of the meshing frequency $SumMFN$ quantifies the noise in the frequency domain data near the meshing frequency and its overtones. Examining the FFT of the data from Cut 2, shown in Figure 3.5, we see that the magnitude of the sidebands near the meshing frequency and its overtones is larger if the tool is broken.

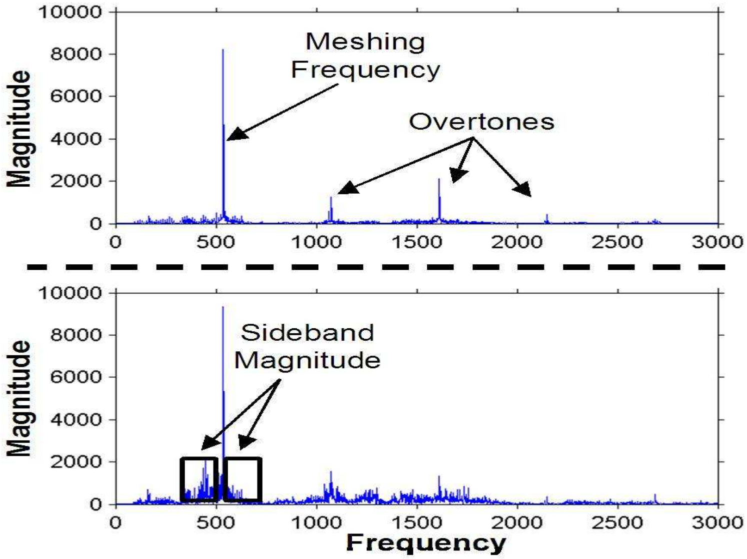


Figure 3.5: FFT of data from new tool (top) and broken tool (bottom).

To obtain an expression for $SumMFN$, we let

$$X_k = \sum_{j=1}^{l} x(j)e^{-\frac{2\pi i}{l}jk}, \tag{3.6}$$

where X_k is the discrete Fourier transform (DFT) of x at frequency k, $i^2 = -1$, and $k = 0, 1, \ldots, l-1$. We define

$$\mathcal{K}_N \triangleq \{v\colon 0 < |v - NM| < 0.5NM\}, \tag{3.7}$$

where M is the meshing frequency, $N = 1, 2, \ldots$, and $v \in \mathbb{R}$. Then

$$SumMFN = \sum_{k \in \mathcal{K}_N} \|X_k\|, \tag{3.8}$$

where $\|\cdot\|$ denotes the Euclidean norm.

Modified kurtosis attempts to provide a more accurate estimate of the *peakedness* of a bi-modal distribution than standard kurtosis (which is based on a unimodal, symmetric distribution). The modified kurtosis of x is given by

$$ModKurt = \frac{K_+ + K_-}{RMS^4}, \tag{3.9}$$

where, for $* \in \{+, -\}$,

$$K_* \triangleq \frac{\sum_{j \in J_*} (x(j) - \mu)^4}{|J_*|}, \tag{3.10}$$

$$J_+ \triangleq \{j : x(j) \geq \mu\} \subset \{1, \ldots, l\}, \tag{3.11}$$

and

$$J_- \triangleq J_+^{\mathrm{C}} \cap \{1, \ldots, l\}. \tag{3.12}$$

Because the shaving process consists of periodic impacts between the shaving cutting tool and workpiece gear, the accelerometer data exhibit a sinusoidal trend. Hence, the data histogram is bi-modal, where one mode corresponds to data collected during meshing and the other mode corresponds to data collected during transitions between meshing. Figure 3.6 shows how the histogram of the accelerometer signal changes as the tool wear increases.

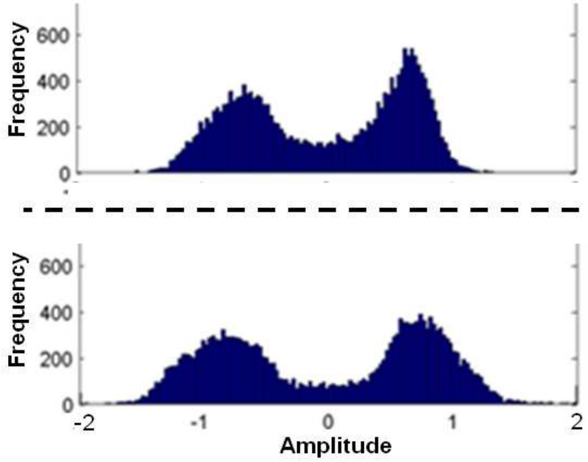


Figure 3.6: Histogram of data from new tool (top) and severely worn tool (bottom).

3.2.4 Feature Selection and Tool Wear Classification

3.2.4.1 Feature dimension reduction and classification using principal component analysis and a multi-layer perceptron

Because of the large number of features extracted and the inability for a single feature to correctly and reliably classify the tool wear state, we consider combining multiple features together to achieve a better classification result. We consider reduction of the dimension of the feature space using principal component analysis (PCA), which extracts key dimensions from the feature space [96], and classification using a multi-layer perceptron (MLP) type of artificial neural network (ANN). We call this the PCA + MLP approach.

To explain PCA, let l be the number of data sets collected and let r be the number of features extracted from each data set. For $j \in \{1, \ldots, r\}$, we define the j^{th} feature vector

$$g_j \triangleq \begin{bmatrix} g_j(1) \\ \vdots \\ g_j(l) \end{bmatrix} \in \mathbb{R}^l, \tag{3.13}$$

where $g_j(k)$ is the value of feature j associated with data set k, and $k \in \{1, \dots, l\}$. Then we define

$$A \triangleq \begin{bmatrix} g_1^{\mathrm{T}} \\ \vdots \\ g_r^{\mathrm{T}} \end{bmatrix}. \tag{3.14}$$

The singular value decomposition [97] of A is given by

$$A = W \Sigma V^{\mathrm{T}}, \tag{3.15}$$

where the columns of W are the eigenvectors of AA^{T} and Σ is a diagonal matrix with nonnegative diagonal entries. Arranging the diagonal entries of Σ in descending order, the reduced-dimension representation A_{red} is constructed by selecting the first $1 \leq q < r$ singular vectors of A so that

$$A_{\mathrm{red}} = W_{[1:q,1:q]}^{\mathrm{T}} A. \tag{3.16}$$

Details on the choice of q for the tool wear classification example are provided in Section 3.3.1.

A sample of a two-dimensional case in which PCA provides feature dimension reduction is shown in Figure 3.7, where the two input features $x1$ and $x2$ can be combined into the single output feature $pc1$ without losing the ability to differentiate between the two classes (dots and circles).

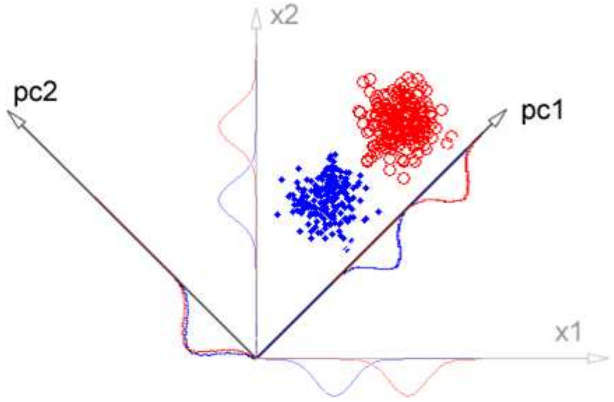


Figure 3.7: Illustration of PCA for feature dimension reduction.

We consider the MLP because of the success it has achieved in previous classification applications [98]. The MLP architecture consists of a feed-forward ANN with three layers, shown in Figure 3.8. The first (input) layer of the MLP inputs the principal components extracted using PCA, while the third (output) layer outputs the classification result. From testing, we achieved the best classification performance when the second (hidden) layer contains 27 neurons. Each neuron in the second layer has a tan-sigmoid activation function, while the neuron in the third layer has a linear activation function [29].

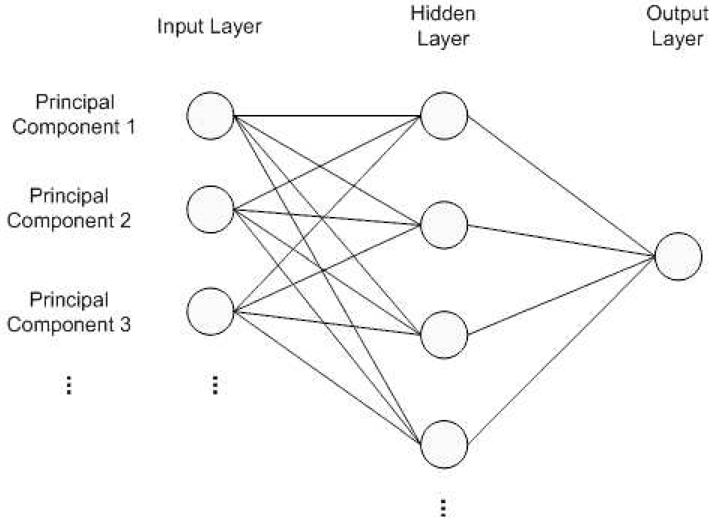


Figure 3.8: Multi-layer perceptron artificial neural network structure used in shaving tool wear classification.

Back-propagation is used to train the MLP network to fix the values of the weights between the layers. Hence, we divided the data into training and testing groups. To improve the accuracy of the weights, we generated additional training data sets by constructing a moving window, as shown in Figure 3.9, that has the same length as the amount of time required for the shaving cutting tool to rotate once. Then the moving window is applied to the center of the data in Cut 2. By advancing the moving window through a single data set in increments of 0.1 s, 14 separate data sets can be generated.

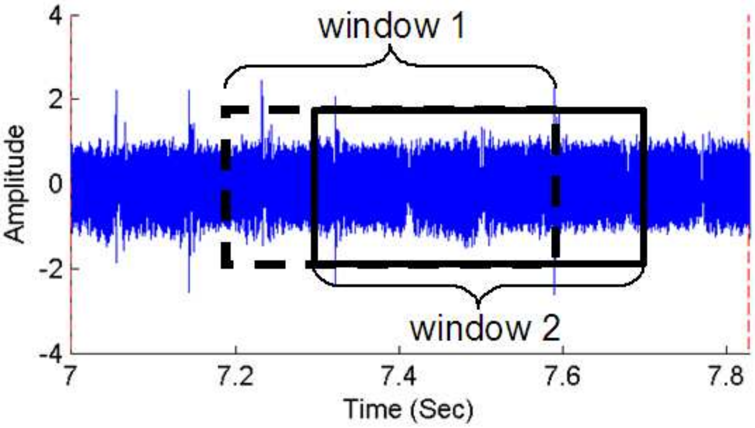


Figure 3.9: Moving windows to increase the amount of training data.

3.2.4.2 Feature selection and classification using modified tabu search and a probabilistic neural network

We also consider modified tabu search with long-term memory (mTS). The long-term memory provides advantages over existing approaches for heuristic feature selection, such as sequential forward selection [27], sequential forward floating selection [99], and tabu search with a short-term memory [100]. First, unlike TS with short-term memory, mTS does not require training to determine the optimal memory length. Second, the long-term memory helps to reduce the probability of the TS converging to a premature local optima, a problem for sequential selection techniques. An overview of the proposed feature selection approach using mTS is shown in Figure 3.10.

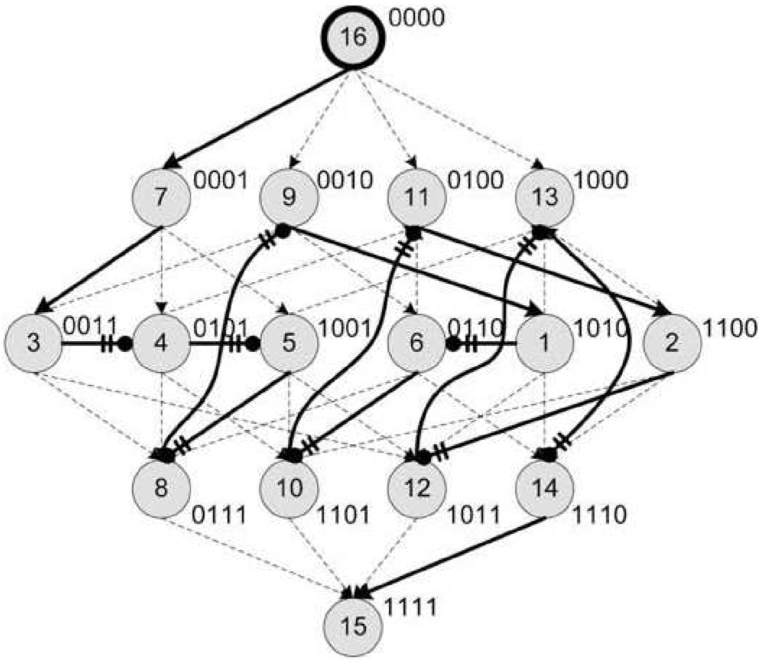


Figure 3.10: Overview of proposed tabu search with long-term memory (mTS).

We use a probabilistic neural network (PNN) to classify the tool condition approach. A PNN, originally proposed in [101], is chosen instead of a back-propagation neural network because a PNN requires less training time and is analytically tractable [102]. The PNN contains an input layer, pattern layer, summation layer, and output layer, as shown in Figure 3.11. Since the number of layers in the PNN architecture is fixed and all the synaptic weights are directly assigned using training samples, training can be completed in a single epoch without requiring subsequent error correction. Furthermore, a PNN converges to a Bayesian classifier given sufficient training data [101].

In the training stage, a set of N_L data samples $\{X_1, X_2, \ldots, X_{N_L}\}$ is used to teach the PNN. Each input sample has m dimensions and each dimension corresponds to a selected feature. We write $X_j = \left[\begin{array}{cccc} x_1^{(j)} & x_2^{(j)} & \cdots & x_m^{(j)} \end{array} \right]^{\mathrm{T}}$, where $j \in \{1, 2, \ldots, N_L\}$. The pattern layer has N_L nodes and the synaptic weight between the j^{th} neuron in the pattern layer and the i^{th} component of the input vector is the value of the i^{th}

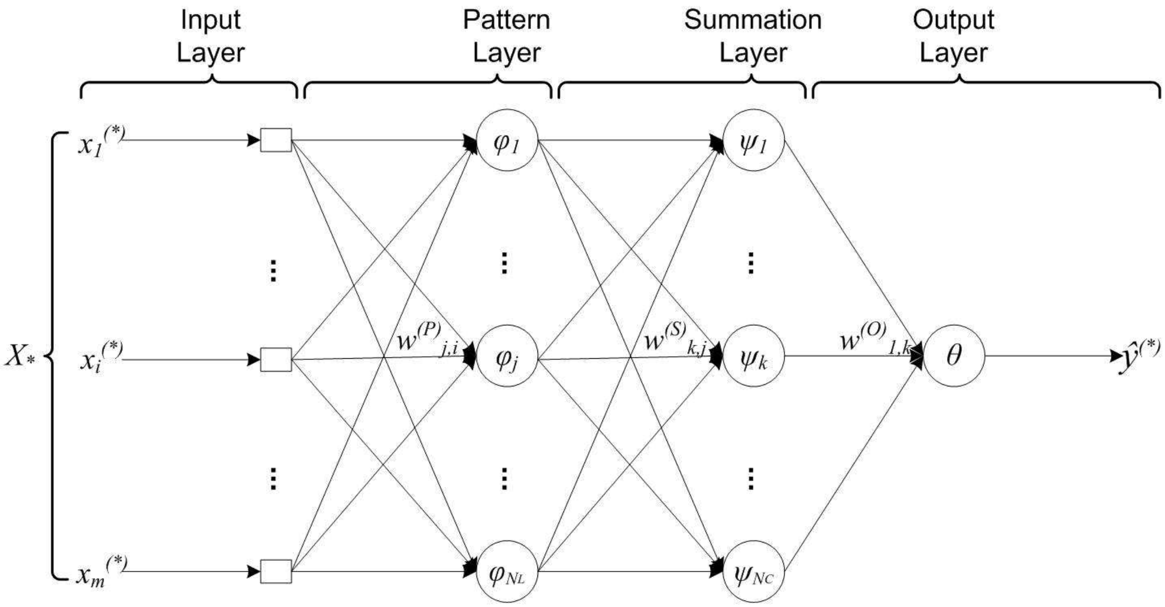


Figure 3.11: Structure of a probabilistic neural network (PNN).

component of the j^{th} training sample, $w^p_{j,i} = x_i^{(j)}$. The summation layer has N_C nodes, where N_C is the number of classes. The synaptic weight between the k^{th} neuron of the summation layer and the j^{th} neuron of the pattern layer is given by

$$w^{(S)}_{k,j} = \begin{cases} 1, & X_j \in \text{class}\, k, \\ 0, & \quad else. \end{cases} \tag{3.17}$$

The synaptic weight between the output neuron and the k^{th} neuron in the summation layer is given by $w^{(O)}_{1,k} = 1$. Using these conventions, a PNN is trained to achieve zero error in classifying the training samples.

After training, a test set of N_T data samples, which do not belong to the training set, are used to evaluate the classification accuracy of the PNN. When a test sample $X_* = \left[\begin{array}{cccc} x_1^{(*)} & x_2^{(*)} & \cdots & x_m^{(*)} \end{array} \right]^{\text{T}}$ is input into the PNN, the output of the j^{th} activation function in the pattern layer is

$$\phi_j = e^{-\frac{\sum_{i=1}^{m}\left(w_{j,i}^{(P)}-x_i^{(*)}\right)^2}{2\sigma_j^2}}, \tag{3.18}$$

where σ_j is a smoothing parameter. Hence, pattern layer neurons with weight vectors different from the input vector X_* output values near zero, while neurons with weight vectors similar to X_* output values near one. For each class, the summation layer combines and normalizes the outputs of the pattern layer to produce a vector of probabilities. The output of the k^{th} neuron in the summation layer is given by

$$\psi_k = \frac{\sum_{j=1}^{N_L} w_{k,j}^{(S)}\phi_j}{\sum_{j=1}^{N_L} w_{k,j}^{(S)}}. \tag{3.19}$$

In the output layer, the outputs of the summation layer are compared and the label corresponding to the class with the maximum probability

$$\hat{y}^{(*)} = \arg\max_k \, \psi_k \tag{3.20}$$

is chosen as the output of the PNN. If $\hat{y}^{(*)}$ equals the true class label $y^{(*)}$ that corresponds to X_*, then the input sample has been correctly classified. Classification accuracy is defined as the percentage of correctly classified samples out of all N_T test samples.

3.3 Results and Discussion

3.3.1 Classification of 5 Different Tools Using the PCA + MLP Approach

We see from Figure 3.12 that a single feature can be used to differentiate wear condition 4 (a tool with multiple broken teeth) from the other 3 wear conditions (tools with no broken teeth, but various amounts of wear). However, Figure 3.12 also shows that it is difficult to differentiate between the wear conditions of unbroken tools.

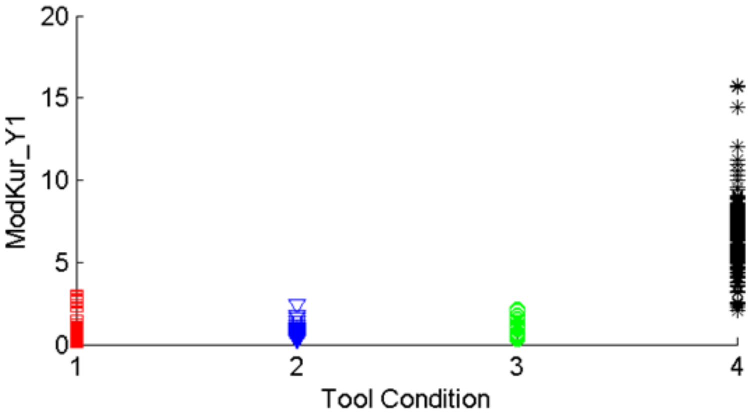


Figure 3.12: Dependence of a single feature on the tool condition.

To consider multiple features and improve the classification accuracy using PCA, we first normalize the feature vectors to have zero mean and unity variance. Then we compute the contribution rate and cumulative contribution rate of each eigenvalue of AA^{T}. As shown in Figure 3.13, the contribution rate decreases and the cumulative contribution rate increases as the principal component index increases. Both trends are consistent with expected trends for PCA. Since the cumulative contribution rate of the first 7 principal components is greater than 90% we use the first 7 principal components as inputs into the MLP.

We see from Figure 3.14(A) that the clusters in the MLP training data set are closely-packed and easily-differentiable. Figure 3.14(B) shows that the clusters in the MLP testing data set are also closely-packed and easily-differentiable. Furthermore, by comparing Figure 3.14(A) with Figure 3.14(B), we see that the boundaries used to classify the training data set can also be used to classify the testing data set.

Using the 7 principal components from the training data set as inputs to the MLP, the training goal of the MLP (a mean squared error $\leq 10^{-5}$) is achieved in 12 iterations using the Levenberg-Marquardt algorithm [103]. After training, the performance of the trained MLP is tested using the testing data set. Figure 3.15 shows that all the testing samples are correctly identified and that the 4 different tool wear conditions

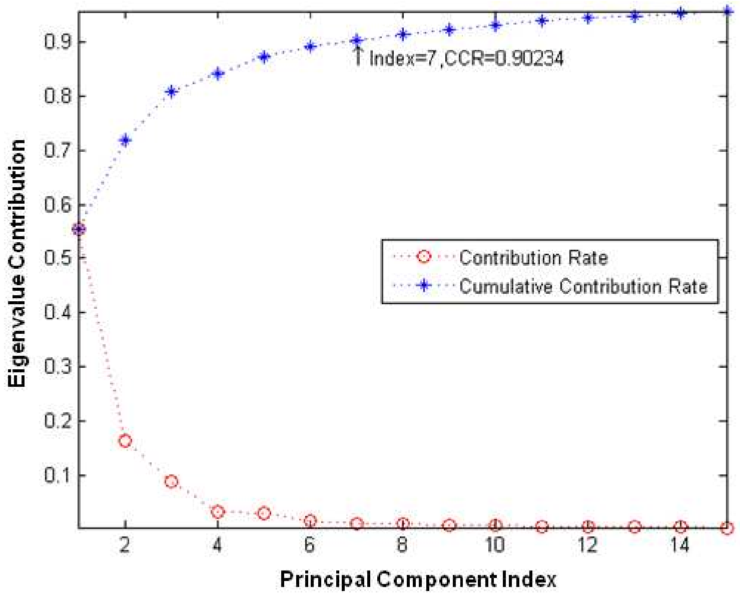


Figure 3.13: Contribution and cumulative contribution rate curves of principal components.

can be correctly classified using multiple features.

3.3.2 Run-To-Failure Test: Tooth Breakage Detection Using the mTS + PNN Approach

During shaving, tools with at least one broken tooth can deteriorate workpiece quality and increase scrap. We investigate whether the proposed technique can be applied to detect the tool tooth breakage after proper training. For tooth breakage detection, the run-to-failure data is divided into two classes, shown in Figure 3.16, which provides the entire history of a single feature ($CF/Y/1$). The first class includes all the data from when the tool has zero broken teeth. The second class includes all the data from when the tool has at least one broken tooth. We randomly choose 50% of the data from each class for training; we use the remaining data for testing. Using the testing data, we evaluate how accurately we can distinguish between unbroken and broken tools using a single feature. Table 3.2 shows the 6 features that provide classification accuracy greater than 90%. Note that 4 of these features include

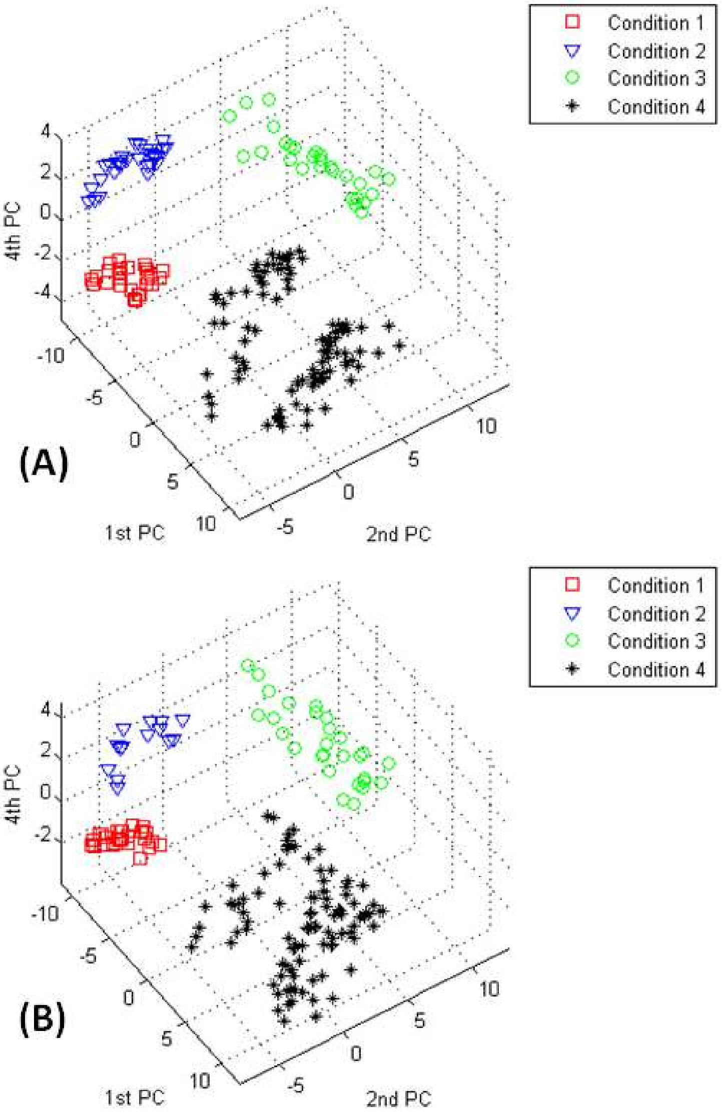


Figure 3.14: Combinations of the 1st, 2nd, and 4th principal components for the training set (A), and the testing set (B).

$SumMFN$, while the remaining features are extracted in the time-frequency domain using a wavelet distribution.

We also use the testing data to evaluate how accurately we can distinguish between unbroken and broken tools using the features extracted from a single sensor. Table 3.3 shows that we can achieve 99.92% classification accuracy using 12 features from Sensor 1 and 99.82% classification accuracy using 23 features from Sensor 2. Therefore, if only a single sensor can be used to detect breakage, Sensor 1 should be used because

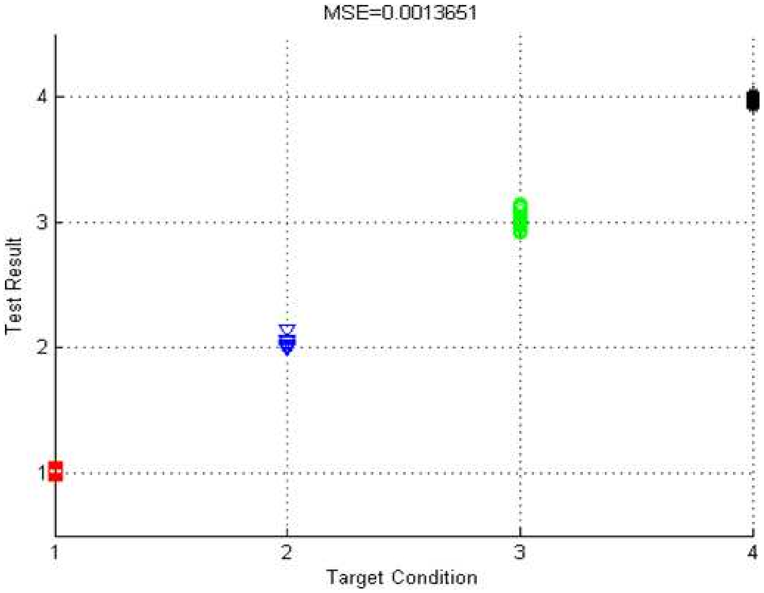


Figure 3.15: Test of PCA + MLP classification result.

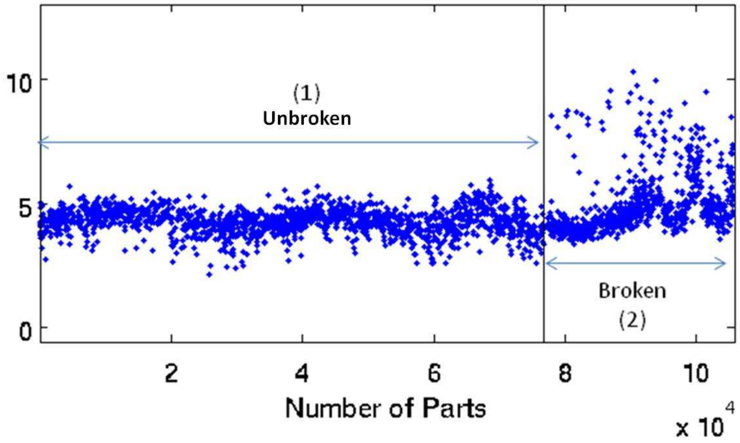


Figure 3.16: Division of the $CF/Y/1$ data into two classes - unbroken (1) and broken (2).

it provides higher classification accuracy using fewer features.

Finally, we use the testing data to evaluate how accurately we can distinguish between unbroken and broken tools using data from both sensors. Table 3.4 shows that we can achieve 99.95% classification accuracy using 17 features.

Feature Name	**Accuracy (%)**
$SumMF1/X/1$	94.37
$EtrpWavelet/X/2$	93.53
$EtrpWavelet/Z/2$	92.81
$SumMF1/X/2$	91.25
$SumMF3/Y/1$	90.88
$SumMF1/Z/1$	90.19

Table 3.2: Breakage detection results using a single feature.

Sensor 1 Only (99.92 % Accuracy)		**Sensor 2 Only (99.82 % Accuracy)**			
$CF/Y/1$	$PPV/Y/1$	$CF/Z/2$	$EtrpWavelet/Z/2$	$MF2/Y/2$	$RecSk/X/2$
$Etrp/Y/1$	$RecSk/Y/1$	$Etrp/X/2$	$Kur/Y/2$	$MF2/Z/2$	$RMS/Y/2$
$EtrpCW/Y/1$	$SumMF1/X/1$	$Etrp/Y/2$	$Kur/Z/2$	$MF3/Y/2$	$SumMF1/Y/2$
$EtrpSTFT/Z/1$	$SumMF1/Y/1$	$EtrpCW/X/2$	$MF1/X/2$	$MF4/X/2$	$SumMF1/Z/2$
$EtrpWavelet/X/1$	$SumMF1/Z/1$	$EtrpSTFT/X/2$	$MF1/Y/2$	$MF4/Z/2$	$SumMF3/X/2$
$Kur/Y/1$	$SumMF2/Z/1$	$EtrpWavelet/X/2$	$MF2/X/2$	$PPV/Y/2$	

Table 3.3: Best features for breakage detection using a single sensor.

Both Sensors (99.95 % Accuracy)		
$Etrp/Y/2$	$MF1/Z/2$	$RecSk/Y/2$
$EtrpCW/X/2$	$MF2/Y/2$	$RMS/X/1$
$EtrpCW/Y/1$	$MF3/Y/1$	$SumMF1/X/1$
$EtrpSTFT/Z/2$	$MF3/Z/2$	$SumMF1/Z/1$
$Kur/Y/1$	$MF4/X/2$	$SumMF2/Z/1$
$Kur/Y/2$	$PPV/Y/2$	

Table 3.4: Best features for breakage detection using both sensors.

3.3.3 Run-to-Failure Test: Tooth Wear Condition Estimation Using the mTS + PNN Approach

We also investigate whether the proposed technique can be applied to estimate the wear condition of the shaving tool after proper training. We divide the data into four classes (normal, worn, broken, and severely broken), as shown in Figure 3.17, which provides the entire history of the feature $CF/Y/1$. The "normal" class includes all data collected before 12,000 workpieces have been cut, since this is the manufacturer's specification for the tool regrind interval. The "worn" class includes all data collected after 12,000 workpieces have been cut and before any teeth have broken. The "broken" class includes all data collected after one tooth has broken but before multiple adjacent teeth have broken. The "severely broken" class includes all data collected after at least two adjacent teeth have broken. After collecting the data, we apply the procedure described in Section 3.3.2 to classify the data. For training, we randomly choose 50% of the data from each of the 4 classes; we use the remaining data for testing.

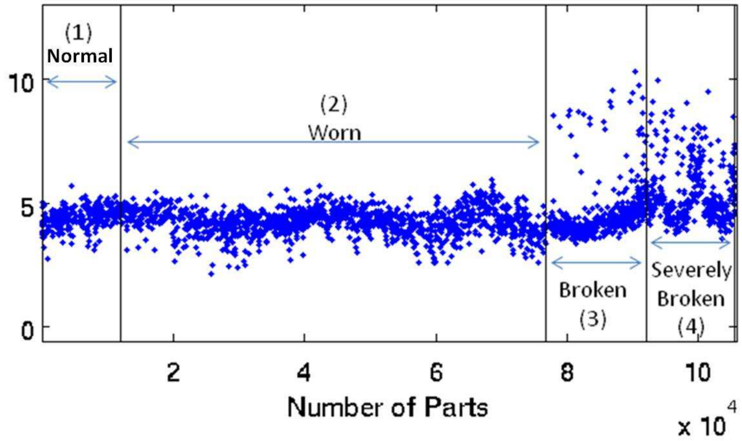


Figure 3.17: Division of the $CF/Y/1$ data into 4 classes - normal (1), worn (2), broken (3), and severely broken (4).

Using the testing data, we evaluate how accurately we can identify the tool condition using a single feature. Table 3.5 shows the 4 features that provide the highest

Feature Name	**Accuracy (%)**
$EtrpWavelet/Z/2$	80.44
$EtrpWavelet/X/2$	79.38
$SumMF1/X/1$	77.84
$SumMF1/X/2$	74.55

Table 3.5: Tool condition estimation results using a single feature.

Sensor 1 Only (97.69% Accuracy)			**Sensor 2 Only (96.25% Accuracy)**		
$CF/Y/1$	$MF1/Z/1$	$SumMF1/Y/1$	$Etrp/X/2$	$MF3/X/2$	$SumMF1/X/2$
$EtrpSTFT/Y/1$	$MF2/X/1$	$SumMF2/Y/1$	$EtrpCW/Z/2$	$MF3/Y/2$	$SumMF1/Z/2$
$EtrpSTFT/Z/1$	$MF2/Y/1$	$SumMF2/Z/1$	$EtrpWavelet/X/2$	$MF3/Z/2$	$SumMF2/X/2$
$EtrpWavelet/X/1$	$MF2/Z/1$	$SumMF3/Y/1$	$EtrpWavelet/Z/2$	$MF4/Z/2$	$SumMF2/Y/2$
$Kur/Z/1$	$MF3/X/1$	$SumMF3/Z/1$	$Mean/Y/2$	$RecSk/X/2$	$SumMF3/Z/2$
$MF1/X/1$	$MF4/Y/1$	$SumMF4/Y/1$	$MF1/Z/2$	$RMS/Y/2$	$SumMF4/Z/2$
$MF1/Y/1$	$SumMF1/X/1$		$MF2/Y/2$	$RMS/Z/2$	

Table 3.6: Best features for tool condition estimation using a single sensor.

classification accuracy. Note that all the features in Table 3.5 are included in Table 3.2.

We also use the testing data to evaluate how accurately we can identify the tool condition using the features extracted from a single sensor. Table 3.6 shows that we can achieve 97.69% classification accuracy using 20 features from Sensor 1 and 96.25% classification accuracy using 20 features from Sensor 2. Therefore, if only a single sensor can be used to classify the tool condition, Sensor 1 should be used since it provides higher classification accuracy than Sensor 2 while using the same number of features. This result is consistent with the result in Section 3.3.2 for breakage detection.

Finally, we use the testing data to evaluate how accurately we can identify the tool condition using data from both sensors. Table 3.7 shows that we can achieve 98.41% classification accuracy using 34 features.

3.3.4 Comparative Study

We compare the PCA + MLP approach with the mTS + PNN approach for classification of tool condition given the run-to-failure data. Both algorithms are

Both Sensors (98.41% Accuracy)			
$Etrp/X/1$	$Mean/Y/2$	$MF3/Y/1$	$SumMF1/Z/2$
$Etrp/Y/2$	$Mean/Z/1$	$MF4/Y/1$	$SumMF2/Y/1$
$EtrpCW/Z/1$	$MF1/X/1$	$RMS/Y/1$	$SumMF2/Z/1$
$EtrpSTFT/Y/1$	$MF1/X/2$	$RMS/Z/1$	$SumMF3/X/1$
$EtrpWavelet/X/1$	$MF2/X/1$	$RMS/Z/2$	$SumMF3/Y/1$
$EtrpWavelet/X/2$	$MF2/X/2$	$SumMF1/X/1$	$SumMF3/Y/2$
$EtrpWavelet/Y/1$	$MF2/Y/2$	$SumMF1/Y/2$	$SumMF3/Z/1$
$EtrpWavelet/Z/2$	$MF2/Z/1$	$SumMF1/Z/1$	$SumMF4/Y/1$

Table 3.7: Best features for tool condition estimation using both sensors.

	Breakage detection		**Tool condition estimation**	
	Accuracy (%)	Processing time (s)	Accuracy (%)	Processing time (s)
PCA + MLP	99.10	32.5	77.50	33.0
mTS + PNN	99.95	7.4	98.41	7.1

Table 3.8: Comparison of feature selection techniques.

implemented in Matlab on a desktop computer with an Intel Core2 CPU (Q8200, 2.33 GHz) and 2 GB of RAM. Because the mTS + PNN method uses 17 features for breakage detection and 34 features for tool condition estimation, we use choose 17 principal components for breakage detection classification and 34 principal components for tool condition using the PCA + MLP approach. We note that the principal components obtained using PCA are not individual features; they are linear combinations of the 108 features extracted from the sensor data. Hence, in the absence of prior knowledge, the PCA + MLP method must extract all 108 features before performing PCA. The mTS + PNN method, in contrast, needs to extract only the features listed in Tables 3.4 and 3.7 to obtain the classification result. Table 3.8 compares the performance of the two techniques when data from both sensors is used. In Table 3.8, "Processing Time" is the average amount of time required to classify one input data measurement. Table 3.8 shows that the classification accuracy and the processing time are both improved using the proposed method.

3.4 Conclusions

In this chapter, we developed a method for classifying the wear of a shaving tool using indirect measurements. We proposed a new metric, $SumMFN$, which helps to characterize the tool wear state. We also applied a recently-developed feature selection and classification technique, modified tabu search with a probabilistic neural network, to achieve high classification accuracy and short processing time. Through an experimental case-study, we showed that the proposed technique correctly classifies the shaving tool wear and provides faster, more accurate results than a conventional approach.

To justify the relevance of the proposed feature selection and wear classification approach, we estimated the potential cost-savings to a company sponsor if the proposed approach were implemented. Considering tooling costs, setup time, and scrap, we showed that the proposed feature selection and wear classification approach could save $> \$11,000$ per tool per machine. Production downtime and maintenance personnel costs, which may be significant, were not considered in this cost-savings estimate.

CHAPTER IV

SISO Pseudo Transfer Function Identification

4.1 Introduction

In many applications of system identification, the system is driven by external excitations that are not measured. Still, output measurements may be used to detect whether the dynamics of the system have changed. In particular, output-only system identification techniques have been used for damage detection in numerous applications [38, 31, 35, 36, 39, 34, 40, 33]. Since the input (excitation) signal is unknown, its statistical properties are often assumed to be known in order to compensate for the lack of knowledge of its time history. It is typically assumed that the input is white noise, and frequency-domain [38, 39, 40, 42], subspace-based [33, 34, 35, 36], and time-domain [31] system identification techniques are used to detect changes in the dynamics of the system. Blind source separation techniques can also be applied [37]. A related technique is used in [104] for blind channel identification in FIR systems.

In this chapter we develop an output-only identification technique that uses multiple outputs but requires neither measurements nor a statistical description of the external input signal. We designate one of the output signals as the pseudo input and another as the pseudo output. The resulting *pseudo transfer function* (PTF) from the pseudo input to the pseudo output thus provides a map between the output signals. The main contribution of this chapter is an analysis of the order and relative degree of the PTF between a pair of outputs. In particular, we prove that the order of the PTF is one less than the order of the system and the relative degree of the PTF is zero. Next, we apply system identification techniques to the PTF. For the case of noisy measurements, we use quadratically-constrained least squares (QCLS) [50] to identify the PTF. Assuming that the input is sufficiently persistent (see Definition B.0.3) and the output-noise autocorrelation is known up to a scale factor, QCLS achieves consistent parameter estimates for arbitrary noise types and input signals. We then use changes in the estimated PTF to detect system faults.

To compare a transmissibility [105] and a PTF, consider the system G in Figure 4.1. The transmissibility from y_1 to y_2 in Figure 4.1(a) assumes that y_1 is colocated with the excitation u. Hence, because the transfer function from u to y_1 is given by

$$y_1 = u,$$

and the transfer function from u to y_2 is given by

$$y_2 = \frac{N_2}{D}u,$$

the transmissibility from y_1 to y_2 is given by

$$y_2 = \frac{N_2}{D}y_1. \tag{4.1}$$

However, the PTF from y_1 to y_2 in Figure 4.1(b) does not assume that y_1 and u are colocated. Then the transfer function from u to y_1 is given by

$$y_1 = \frac{N_1}{D}u,$$

and the transfer function from u to y_2 is given by

$$y_2 = \frac{N_2}{D} u.$$

Hence, the PTF from y_1 to y_2 is given by

$$y_2 = \frac{N_2}{N_1} y_1. \tag{4.2}$$

We note that the transmissibility (4.1) contains pole information, while the PTF (4.2) does not. Furthermore, pole information appears in the output-to-output relationship only if the excitation and sensor measurement are colocated, and hence a transmissibility is a special type of PTF.

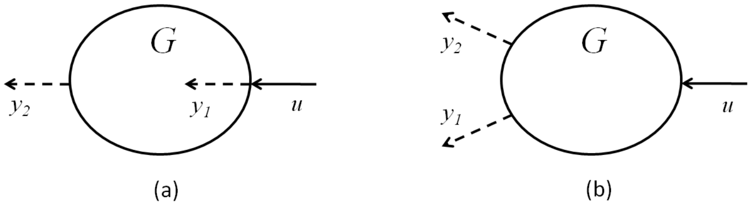


Figure 4.1: Difference between the transmissibility from y_1 to y_2 (a) and the pseudo transfer function (PTF) from y_1 to y_2 (b).

The rest of this chapter is organized as follows. Section 4.2 explains how transfer functions are used to model input-output relationships. Section 4.3 reviews the effect of sampling zeros on the order and relative degree of discrete-time transfer functions. Section 4.4 uses input-output transfer function models to obtain output-output PTF models. Section 4.5 provides simulation examples with noisy and noise-free measurements.

4.2 Problem Formulation

Consider the system $\mathscr{S}$ in Figure 4.2, which has a scalar input u and two scalar outputs y_1 and y_2. For $i = 1, 2$, the subsystem $\mathscr{S}_i$ has the input u and output y_i. Measurement noise corrupting y_1 and y_2 is denoted by w_1 and w_2, respectively. For $\bar{A} \in \mathbb{R}^{n\times n}$, $\bar{B} \in \mathbb{R}^n$, $C \triangleq \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \in \mathbb{R}^{2\times n}$, and $D \triangleq \begin{bmatrix} D_1 \\ D_2 \end{bmatrix} \in \mathbb{R}^2$, a state space representation of $\mathscr{S}$ is denoted by $(\bar{A}, \bar{B}, C, D)$, where

$$\dot{\bar{x}}(t) = \bar{A}\bar{x}(t) + \bar{B}\bar{u}(t), \quad \bar{x}(0) = x_0, \tag{4.3}$$

$$\bar{y}(t) = C\bar{x}(t) + D\bar{u}(t). \tag{4.4}$$

For $i = 1, 2$, a state space representation of $\mathscr{S}_i$ is given by $(\bar{A}, \bar{B}, C_i, D_i)$. We let G denote the continuous-time transfer function corresponding to $(\bar{A}, \bar{B}, C, D)$ and, for $i = 1, 2$, we let G_i denote the continuous-time transfer function corresponding to $(\bar{A}, \bar{B}, C_i, D_i)$.

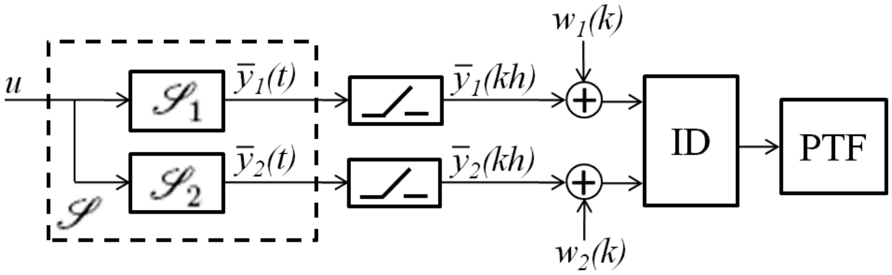


Figure 4.2: Method for identifying a pseudo transfer function (PTF).

For $k \in \{0, 1, \ldots\}$ and sample interval $h > 0$, we apply the variation of constants method [106] to solve (4.3) for $\bar{x}(t)$ from $t = kh$ to $t = kh + h$, which yields

$$\bar{x}(kh+h) = e^{\bar{A}h}\bar{x}(kh) + \int_{kh}^{kh+h} e^{\bar{A}(kh+h-\tau)}\bar{B}\bar{u}(\tau)d\tau. \tag{4.5}$$

We assume that $\bar{u}(t)$ changes sufficiently slowly that $\bar{u}(kh) \approx \bar{u}(kh+h)$. Hence, we rewrite (4.5) and (4.4) as

$$x(k+1) = Ax(k) + Bu(k) \tag{4.6}$$

and

$$y(k) = \begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = Cx(k) + Du(k), \tag{4.7}$$

where $x(k) \triangleq \bar{x}(kh)$, $u(k) \triangleq \bar{u}(kh)$, $A \triangleq e^{\bar{A}h}$, and

$$B \triangleq \int_0^h e^{\bar{A}\tau}d\tau\bar{B}.$$

For $i = 1, 2$, (4.7) implies

$$y_i(k) = C_i x(k) + D_i u(k). \tag{4.8}$$

For $i = 1, 2$, let

$$\begin{aligned} G_i(s) &= C_i\left(sI - \bar{A}\right)^{-1}\bar{B} + D_i \\ &= \frac{\bar{\beta}_{i,m_i}s^{m_i} + \bar{\beta}_{i,m_i-1}s^{m_i-1} + \cdots + \bar{\beta}_{i,0}}{s^n + \bar{\alpha}_{n-1}s^{n-1} + \cdots + \bar{\alpha}_0} \end{aligned}$$

denote the continuous-time transfer function corresponding to $(\bar{A}, \bar{B}, C_i, D_i)$, and let

$$\begin{aligned} G_{i,h}(z) &= C_i \left(zI - A\right)^{-1} B + D_i \\ &= \frac{\beta_{i,n} z^n + \beta_{i,n-1} z^{n-1} + \cdots + \beta_{i,0}}{z^n + \alpha_{n-1} z^{n-1} + \cdots + \alpha_0} \end{aligned} \tag{4.9}$$

denote the discrete-time transfer function corresponding to (A, B, C_i, D_i). We assume that (A, B, C_1, D_1) and (A, B, C_2, D_2) are minimal realizations of $G_{1,h}$ and $G_{2,h}$, respectively, and thus $G_{1,h}$ and $G_{2,h}$ each have order n. This assumption requires that sampling does not cause loss of observability or controllability [107].

To account for the possibly nonzero initial condition x_0 and the resulting free response, we write $G_{1,h}$ and $G_{2,h}$ in terms of the forward shift operator $\mathbf{q}$ [108] rather than in terms of the z-transform variable. For $i = 1, 2$, we obtain

$$y_i = G_{i,h}(\mathbf{q})u = \frac{\eta_i(\mathbf{q})}{\delta(\mathbf{q})}u, \tag{4.10}$$

where

$$\delta(\mathbf{q}) \triangleq \det(\mathbf{q}I - A)$$

and

$$\eta_i(\mathbf{q}) \triangleq C_i \mathrm{adj}(\mathbf{q}I - A)B + D_i\delta(\mathbf{q})$$

are polynomials in $\mathbf{q}$, and y_i and u denote time sequences, that is, $y_i = \{y_i(0), y_i(1), \ldots\}$ and thus $\mathbf{q}y_i = \{y_i(1), y_i(2), \ldots\}$. Note that (4.10) represents an ARMA time-series model rather than a relation between z-transforms. For $i = 1, 2$, since (A, B, C_i, D_i) is a minimal realization of $G_{i,h}$, the polynomials $\delta(\mathbf{q})$ and $\eta_i(\mathbf{q})$ are coprime.

It follows from (4.10) that

$$\eta_2(\mathbf{q})\delta(\mathbf{q})y_1 = \eta_2(\mathbf{q})\eta_1(\mathbf{q})u,$$
$$\eta_1(\mathbf{q})\delta(\mathbf{q})y_2 = \eta_1(\mathbf{q})\eta_2(\mathbf{q})u,$$

and thus

$$\eta_2(\mathbf{q})\delta(\mathbf{q})y_1 = \eta_1(\mathbf{q})\delta(\mathbf{q})y_2.$$

The PTF from y_1 to y_2 can thus be written as

$$y_2 = \frac{\delta(\mathbf{q})\eta_2(\mathbf{q})}{\delta(\mathbf{q})\eta_1(\mathbf{q})}y_1. \tag{4.11}$$

Note that (4.11) is independent of the input u. Because (4.11) is expressed in terms of the forward shift operator $\mathbf{q}$ and not the complex number z, (4.11) accounts for nonzero initial conditions.

Unlike common factors in the complex number z, common factors in the forward-shift operator $\mathbf{q}$ cannot always be cancelled. This point is illustrated by the following example.

Example 4.2.1 Consider the sequences

$$y_1 = \{y_1(0), y_1(1), \ldots\} = \{1, 2, 3, \ldots\}, \tag{4.12}$$

$$y_2 = \{y_2(0), y_2(1), \ldots\} = \{6, 7, 8, \ldots\}. \tag{4.13}$$

Operating on (4.12) and (4.13) with $\mathbf{q}-1$ yields

$$(\mathbf{q}-1)y_1 = \{2-1, 3-2, 4-3, \ldots\} = \{1, 1, 1, \ldots\},$$

$$(\mathbf{q}-1)y_2 = \{7-6, 8-7, 9-8, \ldots\} = \{1, 1, 1, \ldots\}.$$

Hence $(\mathbf{q}-1)y_1 = (\mathbf{q}-1)y_2$, whereas $y_1 \neq y_2$.

Despite Example 4.2.1, we show in Section 4.4 that the common factor $\delta(\mathbf{q})$ in (4.11) can be cancelled. This cancellation is possible because $\delta(\mathbf{q})$, $\eta_1(\mathbf{q})$, and $\eta_2(\mathbf{q})$ are obtained from minimal state-space realizations of $G_{1,h}$ and $G_{2,h}$ with the same initial condition x_0.

To identify a PTF, output data are collected from sensors. Although we make no explicit assumptions about the persistency of u, the pseudo-input y_1 must be sufficiently persistent for PTF identification. Estimation of the PTF from y_1 to y_2 can be classified as a functional errors-in-variables identification problem [109] because y_1 is an arbitrary (not necessarily white) signal.

4.3 Sampling Zeros

Discretization of a continuous-time system may yield a discrete-time system that has more zeros than the continuous-time system. The additional zeros are called sampling zeros [110]. For discrete-time models that arise from sampled-data systems, the following result shows that zero-order-hold sampling of G_i, $i = 1, 2$, yields the discrete-time transfer function $G_{i,h}$ given by (4.9), whose relative degree $d_i \triangleq n - m_i$ is either 0 or 1.

Proposition 4.3.1 Let $i = 1, 2$. If $n = m_i$, then $d_i = 0$ and $\beta_{i,n} \neq 0$. If $n > m_i$, then $d_i = 1$, $\beta_{i,n} = 0$, and $\beta_{i,n-1} \neq 0$. Furthermore, as $h \to 0$, $d_i - 1$ zeros of $G_{i,h}(\mathbf{q})$

approach the roots of $J_{d_i}(\mathbf{q})$, where

$$J_{d_i}(\mathbf{q}) \triangleq J_{d_i,1}\mathbf{q}^{d_i-1} + J_{d_i,2}\mathbf{q}^{d_i-2} + \cdots + J_{d_i,d_i},$$

and, for $k \in \{1, \ldots, d_i\}$,

$$J_{d_i,k} \triangleq \sum_{j=1}^{k} (-1)^{k-j} j^{d_i} \left(\begin{array}{c} d_i + 1 \\ k - j \end{array} \right).$$

Furthermore, the roots of $J_{d_i}(\mathbf{q})$ are distinct and negative.

Proof 4.3.1 *It is shown in Theorem 1 of [110] that* $d_i - 1$ *zeros of* $G_{i,h}(\boldsymbol{q})$ *approach the roots of* $J_{d_i}(\boldsymbol{q})$. *Theorem 2.1 of [111] shows that the roots of* $J_{d_i}(\boldsymbol{q})$ *are distinct and negative.* □

Note that $J_{d_i}(\mathbf{q})$ is the Euler-Frobenius polynomial [111]

$$J_{d_i}(\mathbf{q}) \triangleq \begin{cases} 1, & d_i = 1, \\ \frac{(1-\mathbf{q})^{d_i+1}}{\mathbf{q}} \left(\mathbf{q}\frac{\mathrm{d}}{\mathrm{d}\mathbf{q}}\right)^{d_i-1} \left(\frac{\mathbf{q}}{(1-\mathbf{q})^2}\right), & d_i \geq 2. \end{cases}$$

For example,

$$J_2(\mathbf{q}) = \mathbf{q} + 1,$$

with root -1,

$$J_3(\mathbf{q}) = \mathbf{q}^2 + 4\mathbf{q} + 1,$$

with roots $-2 \pm \sqrt{3}$, and

$$J_4(\mathbf{q}) = \mathbf{q}^3 + 11\mathbf{q}^2 + 11\mathbf{q} + 1,$$

with roots -1, $-5 \pm 2\sqrt{6}$. Finally, since $J_d(\mathbf{q})$ is a palindromic polynomial, it follows that -1 is a root of $J_{d_i}(\mathbf{q})$ with multiplicity 0 or 1, and, if λ is a root of $J_{d_i}(\mathbf{q})$, then $1/\lambda$ is a root of $J_{d_i}(\mathbf{q})$ [112]. Since, in addition, all of the roots of $J_{d_i}(\mathbf{q})$ are negative, the asymptotic zeros due to sampling appear at either -1 or in negative reciprocal pairs or both.

4.4 Output-Only Model

For $i = 1, 2$, we formulate an equivalent matrix representation of (4.11) by writing

$$\eta_i(\mathbf{q}) = \sum_{j=0}^{n} \beta_{i,j}\mathbf{q}^j, \quad \delta(\mathbf{q}) = \sum_{j=0}^{n} \alpha_j \mathbf{q}^j.$$

We define

$$v \triangleq NY \in \mathbb{R}^{l-n}, \tag{4.14}$$

where

$$N \triangleq \begin{bmatrix} N_2 & -N_1 \end{bmatrix} \in \mathbb{R}^{(l-n)\times 2l},$$

and, for $i = 1, 2$,

$$N_i \triangleq \begin{bmatrix} \beta_{i,0} & \dots & \beta_{i,n} & 0 & \dots & 0 \\ 0 & \beta_{i,0} & \dots & \beta_{i,n} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \beta_{i,0} & \dots & \beta_{i,n} \end{bmatrix} \in \mathbb{R}^{(l-n)\times l}.$$

Without loss of generality, we assume $\alpha_n = 1$ and express (4.11) as

$$\Delta v = 0, \tag{4.15}$$

where, for $l > 2n$ data points,

$$\Delta \triangleq \begin{bmatrix} \alpha_0 & \dots & \alpha_{n-1} & 1 & 0 & \dots & 0 \\ 0 & \alpha_0 & \dots & \alpha_{n-1} & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \alpha_0 & \dots & \alpha_{n-1} & 1 \end{bmatrix} \in \mathbb{R}^{(l-2n)\times(l-n)}.$$

Finally, we define

$$Y \triangleq \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} \triangleq \begin{bmatrix} y_1(0) \\ \vdots \\ y_1(l-1) \\ y_2(0) \\ \vdots \\ y_2(l-1) \end{bmatrix} \in \mathbb{R}^{2l}. \tag{4.16}$$

Combining (4.15) and (4.14) yields

$$\Delta N Y = 0, \tag{4.17}$$

which is an equivalent matrix formulation of (4.11).

From (4.16) we have for $i = 1, 2$,

$$Y_i = \Gamma_i x_0 + \mathcal{H}_i U, \tag{4.18}$$

where

$$\Gamma_i \triangleq \begin{bmatrix} C_i \\ \vdots \\ C_i A^{l-1} \end{bmatrix} \in \mathbb{R}^{l\times n}, \quad U \triangleq \begin{bmatrix} u(0) \\ \vdots \\ u(l-1) \end{bmatrix} \in \mathbb{R}^{l},$$

and

$$\mathcal{H}_i \triangleq \begin{bmatrix} D_i & 0 & \dots & 0 \\ C_iB & D_i & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ C_iA^{l-2}B & \dots & C_iB & D_i \end{bmatrix} \triangleq \begin{bmatrix} H_{i,0} & 0 & \dots & 0 \\ H_{i,1} & H_{i,0} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ H_{i,l-1} & \dots & H_{i,1} & H_{i,0} \end{bmatrix} \in \mathbb{R}^{l\times l}.$$

Then we define

$$Y_{i,\mathrm{free}} \triangleq \Gamma_i x_0$$

and

$$Y_{i,\mathrm{forced}} \triangleq \mathcal{H}_i U,$$

so that

$$Y_i = Y_{i,\mathrm{free}} + Y_{i,\mathrm{forced}}.$$

Furthermore,

$$Y = \Gamma x_0 + \mathcal{H} U,$$

where $\Gamma \triangleq \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \end{bmatrix}$ and $\mathcal{H} \triangleq \begin{bmatrix} \mathcal{H}_1 \\ \mathcal{H}_2 \end{bmatrix}$. Finally,

$$Y = Y_{\text{free}} + Y_{\text{forced}},$$

where

$$Y_{\text{free}} \triangleq \begin{bmatrix} Y_{1,\text{free}} \\ Y_{2,\text{free}} \end{bmatrix}$$

and

$$Y_{\text{forced}} \triangleq \begin{bmatrix} Y_{1,\text{forced}} \\ Y_{2,\text{forced}} \end{bmatrix}.$$

Hence, (4.17) can be written as

$$\Delta N(Y_{\text{free}} + Y_{\text{forced}}) = 0. \tag{4.19}$$

Now we show that $Y_{\text{free}} \in \mathcal{N}(N)$ and $Y_{\text{forced}} \in \mathcal{N}(N)$, and thus $Y \in \mathcal{N}(N)$, where $\mathcal{N}(N)$ is the null space of N.

Lemma 4.4.1 $N_2\Gamma_1 = N_1\Gamma_2$.

Proof 4.4.1 *See Appendix A.* □

Proposition 4.4.1 $NY_{\text{free}} = 0$.

Proof 4.4.2 *With $u(k) \equiv 0$, (4.18) implies*

$$N_2 Y_{1,\text{free}} = N_2\Gamma_1 x(0), \tag{4.20}$$

$$N_1 Y_{2,\mathrm{free}} = N_1 \Gamma_2 x(0). \tag{4.21}$$

Subtracting (4.21) from (4.20) and using Lemma 4.4.1, we have

$$\begin{aligned} NY_{\mathrm{free}} &= N_2 Y_{1,\mathrm{free}} - N_1 Y_{2,\mathrm{free}} \\ &= N_2 \Gamma_1 x(0) - N_1 \Gamma_2 x(0) \\ &= 0. \end{aligned}$$

□

Lemma 4.4.2 $N_2 \mathcal{H}_1 = N_1 \mathcal{H}_2$.

Proof 4.4.3 *See Appendix A.* □

Example 4.4.1 *Consider the IIR system*

$$y_1(k) = \frac{1}{\boldsymbol{q}^2 - b\boldsymbol{q} - a} u(k), \quad y_2(k) = \frac{\boldsymbol{q}}{\boldsymbol{q}^2 - b\boldsymbol{q} - a} u(k),$$

where

$$A = \begin{bmatrix} 0 & 1 \\ a & b \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C_1 = \begin{bmatrix} 1 & 0 \end{bmatrix},$$

$$C_2 = \begin{bmatrix} 0 & 1 \end{bmatrix}, \quad D_1 = D_2 = 0.$$

Finally, let $l = 5 > 2n = 4$. Then

$$N_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \quad N_2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

$$\Gamma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ a & b \\ ab & a+b^2 \\ a^2+ab^2 & 2ab+b^3 \end{bmatrix}, \quad \Gamma_2 = \begin{bmatrix} 0 & 1 \\ a & b \\ ab & a+b^2 \\ a^2+ab^2 & 2ab+b^3 \\ 2a^2b+ab^3 & a^2+3ab^2+b^4 \end{bmatrix},$$

$$\mathcal{H}_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ b & 1 & 0 & 0 & 0 \\ a+b^2 & b & 1 & 0 & 0 \end{bmatrix}, \quad \mathcal{H}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ b & 1 & 0 & 0 & 0 \\ a+b^2 & b & 1 & 0 & 0 \\ 2ab+b^3 & a+b^2 & b & 1 & 0 \end{bmatrix}.$$

Hence,

$$N_2\Gamma_1 = N_1\Gamma_2 = \begin{bmatrix} 0 & 1 \\ a & b \\ ab & a+b^2 \end{bmatrix},$$

which confirms Lemma 4.4.1. Furthermore,

$$N_2\mathcal{H}_1 = N_1\mathcal{H}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ b & 1 & 0 & 0 & 0 \end{bmatrix},$$

which confirms Lemma 4.4.2.

Proposition 4.4.2 $NY_{\text{forced}} = 0$.

Proof 4.4.4 *With $x_0 = 0$, (4.18) implies*

$$N_2Y_{1,\text{forced}} = N_2\mathcal{H}_1U, \tag{4.22}$$

$$N_1Y_{2,\text{forced}} = N_1\mathcal{H}_2U. \tag{4.23}$$

Subtracting (4.23) from (4.22) and invoking Lemma 4.4.2 yields

$$\begin{aligned} NY_{\text{forced}} &= N_2 Y_{1,\text{forced}} - N_1 Y_{2,\text{forced}} \\ &= N_2 \mathcal{H}_1 U - N_1 \mathcal{H}_2 U \\ &= 0. \end{aligned} \qquad \square$$

Combining Propositions 4.4.1 and 4.4.2 with (4.19) yields the following result, which is stronger than (4.17).

Theorem 4.4.1 $NY = 0$.

Theorem 4.4.1 is an equivalent matrix formulation of

$$y_2 = \frac{\eta_2(\mathbf{q})}{\eta_1(\mathbf{q})} y_1. \tag{4.24}$$

Comparing (4.24) with (4.11), Theorem 4.4.1 shows that cancellation of the $\delta(\mathbf{q})$ in the numerator and denominator of (4.11) is valid. The following result characterizes the order and relative degree of the PTF given by (4.24).

Proposition 4.4.3 The PTF in (4.24) obtained by sampling the one-input, two-output continuous-time system (4.3),(4.4) and using zero-order hold has relative degree 1 if $D_1 \neq 0$ and $D_2 = 0$; relative degree -1 if $D_1 = 0$ and $D_2 \neq 0$; and relative degree 0 otherwise. Furthermore, the PTF has order n if $D_1 \neq 0$ and order $n-1$ otherwise.

Proof 4.4.5 *Since $\eta_1(\boldsymbol{q})$ and $\eta_2(\boldsymbol{q})$ are the numerators of the discrete-time transfer functions from u to y_1 and u to y_2, respectively, Proposition 4.3.1 implies that, for $i = 1, 2$, $\eta_i(\boldsymbol{q})$ has degree n if the continuous-time system has relative degree 0 and degree $n-1$ otherwise.* □

Note that y_1 and y_2 are related by

$$y_2 = \frac{C_2 \mathrm{adj}(\mathbf{q}I - A)B + D_2 \det(\mathbf{q}I - A)}{C_1 \mathrm{adj}(\mathbf{q}I - A)B + D_1 \det(\mathbf{q}I - A)} y_1, \tag{4.25}$$

which shows that the PTF from y_1 to y_2 captures information about the zeros of $G_{1,h}$ and $G_{2,h}$.

4.5 Examples

Consider the mass-spring-damper structure in Figure 4.3, which has the equations of motion

$$M\ddot{q}(t) + C_\mathrm{d}\dot{q}(t) + Kq(t) = F(t), \tag{4.26}$$

where

$$q(t) = \begin{bmatrix} q_1(t) \\ q_2(t) \end{bmatrix}, \ M = \begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix}, \ C_\mathrm{d} = \begin{bmatrix} c_1 + c_2 & -c_2 \\ -c_2 & c_2 + c_3 \end{bmatrix},$$
$$K = \begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 + k_3 \end{bmatrix}, \ F(t) = \begin{bmatrix} f_1(t) \\ f_2(t) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(t). \tag{4.27}$$

We express (4.26) in state-space form (4.3), where $\bar{x}(t) = \begin{bmatrix} q_1(t) & q_2(t) & v_1(t) & v_2(t) \end{bmatrix}^\mathrm{T}$,

$$\bar{A} \triangleq \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{k_1+k_2}{m_1} & \frac{k_2}{m_1} & -\frac{c_1+c_2}{m_1} & \frac{c_2}{m_1} \\ \frac{k_2}{m_2} & -\frac{k_2+k_3}{m_2} & \frac{c_2}{m_2} & -\frac{c_2+c_3}{m_2} \end{bmatrix}, \ \bar{B} \triangleq \begin{bmatrix} 0 \\ 0 \\ \frac{1}{m_1} \\ 0 \end{bmatrix},$$

and where $m_1 = \frac{1}{2}m_2 = 1$ kg, $k_1 = \frac{4}{5}k_2 = \frac{2}{3}k_3 = 4$ N/m, $c_1 = \frac{5}{2}c_2 = \frac{5}{7}c_3 = 0.5$ kg-m/s, and $h = 0.5$ s.

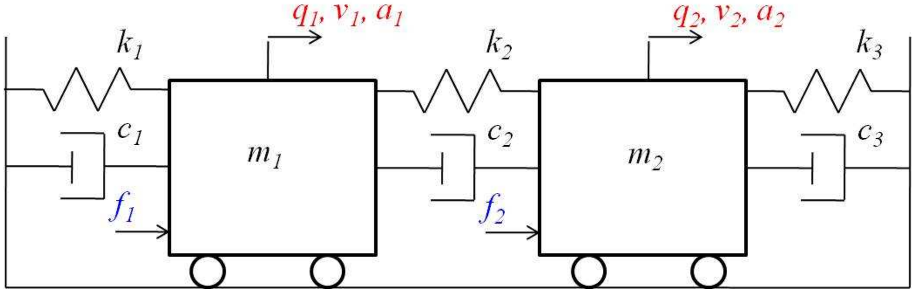


Figure 4.3: 2 DOF Mass-spring-damper structure.

We use Markov parameters to characterize (4.25) because Markov parameters can be estimated consistently under more general conditions than coefficients, poles, or zeros [113]. Hence, we consider the μ-Markov model structure

$$\begin{aligned} y_2(k) = &- a_\mu y_2(k-\mu) - \cdots - a_{\mu+n_{\rm mod}-1} y_2(k-\mu-n_{\rm mod}+1) \\ &+ H_0 y_1(k) + \cdots + H_{\mu-1} y_1(k-\mu+1) \\ &+ b_\mu y_1(k-\mu) + \cdots + b_{\mu+n_{\rm mod}-1} y_1(k-\mu-n_{\rm mod}+1). \end{aligned} \tag{4.28}$$

The order of this model is $n_{\rm mod}$ due to the cancellation of μ poles and zeros. The absence of terms involving $y_2(k-1), \ldots, y_2(k-\mu+1)$ is responsible for the explicit presence of the Markov parameters $H_0, \ldots, H_{\mu-1}$. Note that, if y_1 is a realization of a white random process, the Markov parameters $H_0, \ldots, H_{\mu-1}$ can be estimated consistently using least squares (LS) [113]. However, because y_1 is not a realization of a white noise process, we use QCLS to provide consistent estimates of the Markov parameters [50].

We investigate the accuracy of the estimates of the Markov parameters of a PTF for various u and x_0. We quantify the difference between the estimated and actual

Markov parameters by defining

$$\varepsilon_{\mathcal{T}} \triangleq \sigma_{\max}\left(\mathcal{T} - \hat{\mathcal{T}}\right),$$

where, for $\mu = 10$,

$$\mathcal{T} \triangleq \begin{bmatrix} H_0 & \cdots & 0 \\ \vdots & \ddots & 0 \\ H_9 & \cdots & H_0 \end{bmatrix}$$

is the truncated Toeplitz operator [114] and $\hat{\mathcal{T}}$ is an estimate of $\mathcal{T}$.

4.5.1 Effect of model order with noise-free measurements

We first investigate the effect of the order of the μ-Markov model on the accuracy of the estimated Markov parameters of the PTF. We consider three different input signals u, namely, a realization of a white Gaussian process with mean 0 and variance 1, a square wave with period 0.33 s, and a sine wave with period 0.33 s, as well as zero and nonzero x_0. We simulate (4.26) to obtain the position q_1 of the first mass and the velocity v_2 of the second mass. We then use LS with the μ-Markov model structure (4.28) of relative degree 0 to estimate the first 10 Markov parameters of the PTF from q_1 to v_2. For 50 realizations of 500 samples of each u, we construct $\varepsilon_{\mathcal{T}}$ and compute the average estimation error $\bar{\varepsilon}_{\mathcal{T}}$ over all u. Plotting $\bar{\varepsilon}_{\mathcal{T}}$ as a function of the μ-Markov model order $n_{\rm mod}$ in Figure 4.4 shows that the Markov parameters are correctly estimated if $n_{\rm mod} \geq n - 1 = 3$.

4.5.2 Consistency of the estimated PTF

We now investigate the effect of output noise on the accuracy of the identified PTF by adding noise to both the pseudo input and pseudo output. Since the pseudo input

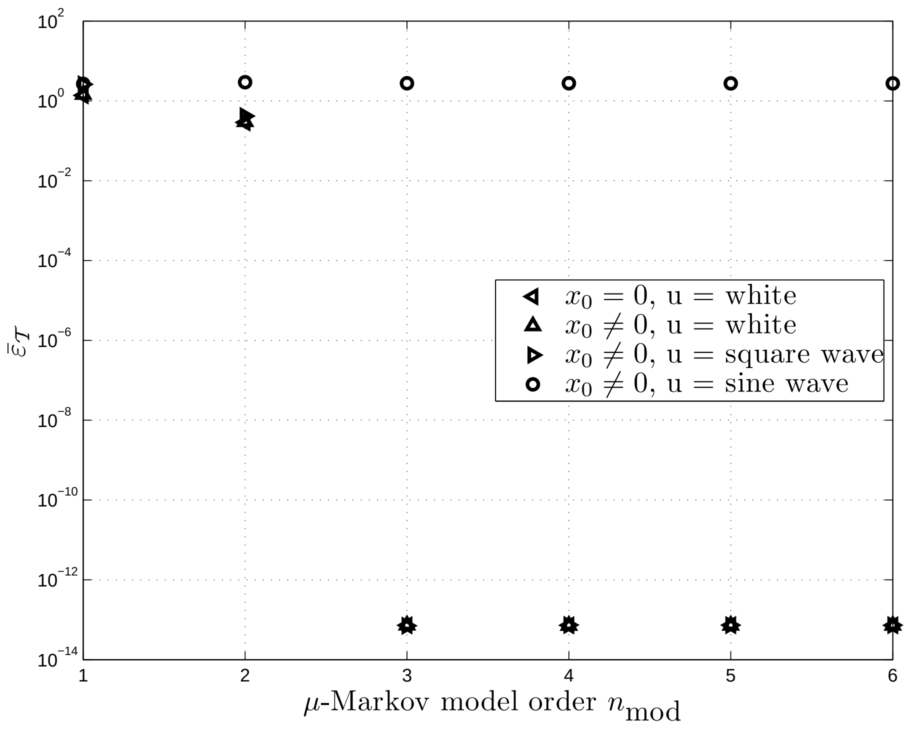


Figure 4.4: Error in the estimated Markov parameters of the PTF from q_1 to v_2 as the μ-Markov model order increases.

is not a realization of a white random process, LS with a μ-Markov model structure does not yield consistency of the Markov parameters. Hence, for comparison, we use both LS and QCLS [50] with the μ-Markov model structure (4.28) with $n_{\text{mod}} = 3$ and relative degree 0 to estimate the first $\mu = 10$ Markov parameters of the PTF from q_1 to v_2. To obtain consistent estimates, QCLS requires knowledge of the noise autocorrelation

$$R \triangleq \mathbb{E}\left[\psi(k)\psi^{\mathrm{T}}(k)\right] \in \mathbb{R}^{(2n_{\text{mod}}+\mu+1)\times(2n_{\text{mod}}+\mu+1)}$$

to within a scalar multiple, where $n_{\rm mod}+\mu-1\le k\le l$, l is the number of data samples,

$$\psi(k) \triangleq \left[\begin{array}{c} \psi_2(k) \\ -\psi_1(k) \end{array}\right],$$

and, for $i=1,2$,

$$\psi_i(k) \triangleq \left[\begin{array}{cccc} w_i(k) & w_i(k-\mu) & \cdots & w_i(k-\mu-n_{\rm mod}+1) \end{array}\right]^{\rm T},$$

Note that we form R by solving a discrete-time Lyapunov equation for the autocorrelation of the noise vector $\psi(k)$ [115].

We choose $x(0)\neq 0$ and $u\sim N(0,1)$ and, for $i=1,2$, we choose the standard deviation of w_i so that the signal-to-noise ratio (SNR) is 10. For LS, we let $\mu=10$ in (4.28) and estimate the first 10 Markov parameters of the PTF from q_1 to v_2. For QCLS, we let $\mu=1$ in (4.28) and estimate the coefficients of (4.28). We then impulse the identified model (4.28) to obtain an indirect estimate of the first 10 Markov parameters of the PTF from q_1 to v_2. We use the estimated Markov parameters from LS and QCLS along with the true Markov parameters to compute $\varepsilon_{\mathcal{T}}$, which we average over 50 noise sequences and 10 input sequences to obtain $\bar{\varepsilon}_{\mathcal{T}}$. Figure 4.5, shows that QCLS provides consistent estimates of the PTF, while LS does not. This result is expected since the pseudo-input y_1 is not white.

4.5.3 Damage detection through PTF estimation

Finally, we investigate whether changes in the identified PTF can be used to detect changes in system parameters. We choose $x(0)\neq 0$ and $u\sim N(0,1)$ and simulate (4.26)–(4.27). In this case, we consider the exactly proper PTF from the acceleration a_1 of the first mass to the acceleration a_2 of the second mass. Both continuous-time

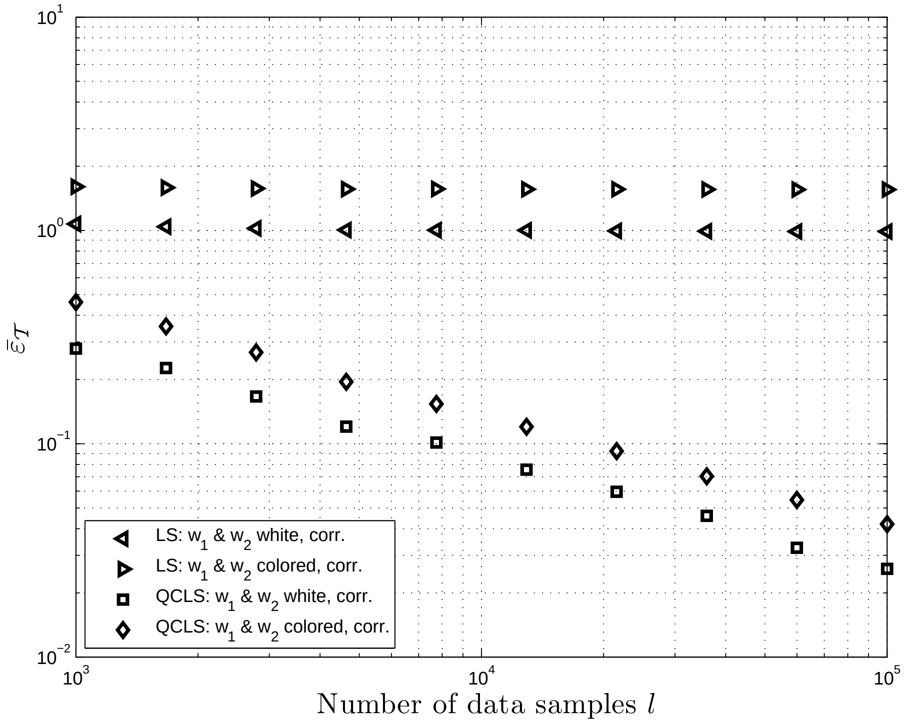


Figure 4.5: Error in the estimated PTF from q_1 to v_2, where zero-mean noise is added to both output measurements.

transfer functions G_1 and G_2 have 2 zeros at $s = 0$. Under sampling, these zeros map to the same locations, and thus the order of the PTF from y_1 to y_2 is $n = 2$. Adding white noise with a SNR of 100 to both y_1 and y_2, we use QCLS with a μ-Markov model structure (4.28) with $n_{\text{mod}} = n = 2$, $\mu = 1$ and relative degree 0 to estimate the model coefficients. We then impulse the identified model to estimate the first $\mu = 2$ Markov parameters of the PTF from a_1 to a_2.

At time step $k = 34$, the stiffnesses k_i are reduced by a factor of 2 and the damping ratios c_i are increased by a factor of 3. Figure 4.6 shows the change in the true and estimated Markov parameters after damage is introduced. At time step $k = 66$, we re-initialize the QCLS algorithm. Figure 4.6 shows that the Markov parameter estimates converge to the modified values after re-initialization. Note that the QCLS algorithm returns zero estimates until $l = 3n_{\text{mod}} + 2\mu - 1 = 9$ data samples have been collected.

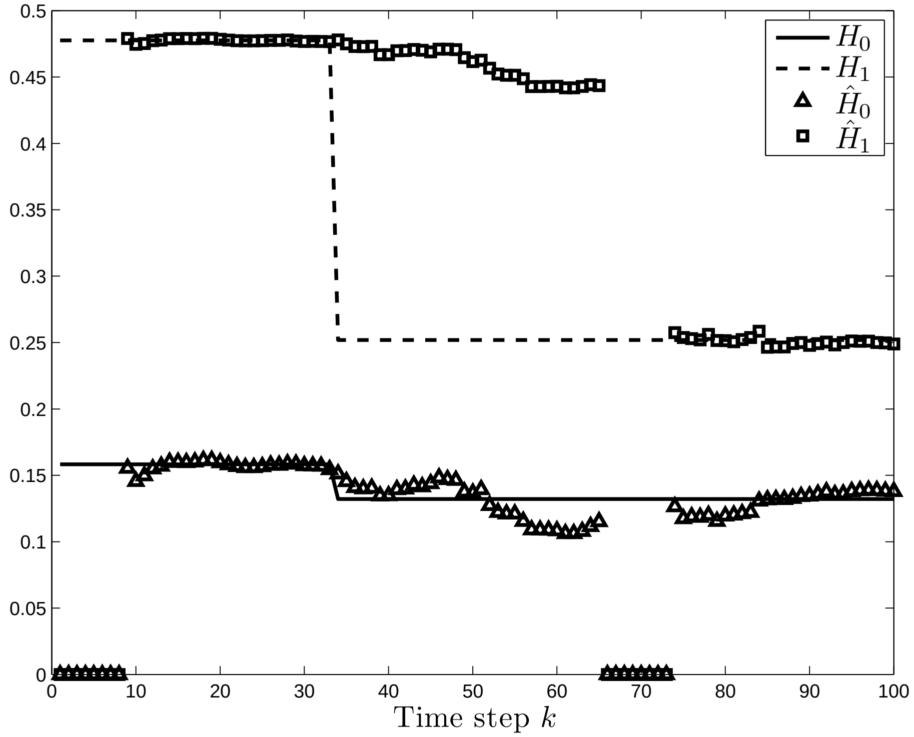


Figure 4.6: Damage detection using the PTF from a_1 to a_2, where white, zero-mean noise is added to both output measurements.

4.6 Conclusions

For SISO sampled-data systems, we defined the concept of a pseudo transfer function (PTF) from one output to another. A PTF, which contains information about the zeros of the system, does not depend on either the input to the system or its initial condition. We provided results to illustrate that a PTF can be estimated consistently in the presence of output noise if the autocorrelation of the output noise is known to within a scalar multiple. We also demonstrated how PTFs can be used to detect faults in a simulated mass-spring-damper system. Future research will address consistency of the PTF estimate when the output noise autocorrelation is unknown using an instrumental variable approach [116]. Investigation into damage localization using PTFs is also left for future research.

CHAPTER V

MIMO Pseudo Transfer Function Identification

5.1 Introduction

In some applications, the excitation may be unknown and thus output measurement data may be the only available information for system identification. In this case, it is typically assumed that the excitation is generated by a white random process, and frequency-domain [38, 39, 40, 42], subspace-based [33, 34, 35, 36], and time-domain [31] system identification techniques are used to detect changes in the dynamics of the system. Blind source separation techniques can also be applied [37].

As an alternative approach, pseudo transfer functions (PTFs) are used in [56, 58] to detect system changes under unknown excitation. SISO PTFs for single-input, two-output systems are characterized in [56, 58]. Sampling introduces additional zeros into a discrete-time input-output model if the relative degree of the continuous-time system is greater than 1 [110]. Hence, for a strictly proper continuous-time system, the order of the SISO PTF arising from a sampled-data application is $n-1$, where n is the order of the underlying system [56, 58]. Since a PTF is essentially a ratio of transfer functions, the information in a PTF consists of information about the zeros of the system from the unknown excitation to each of the outputs. For applications involving structural dynamics, PTFs can be viewed as an extension of transmissibilities, where PTFs do not require that one of the outputs be colocated with the prescribed displacement [42].

The results of [56, 58] assume that a single unknown excitation is applied to the system. For the case of multiple excitation signals, it is shown in [57] that additional outputs can be used to obtain a MIMO PTF that is independent of the excitation signals. For example, the system shown in Figure 5.1 is excited by both u_1 and u_2 so that the output measurements y_1, y_2, and y_3 contain contributions from both u_1 and u_2. Therefore, for $j = 1, 2, 3$, $y_{j,2}$, which is the effect of u_2 on the j^{th} output, appears as output noise corrupting measurement y_j. Furthermore, identification of the SISO PTF from y_1 to y_2 involves estimation in the presence of noise due to u_2. However, identification of the MIMO PTF from $[y_1\ y_2]^{\mathrm{T}}$ to y_3 is noise-free in the sense that, in the absence of additional noise sources, exact identification of the MIMO PTF is possible using finite data. In [57], recursive least squares is used to identify MIMO PTFs.

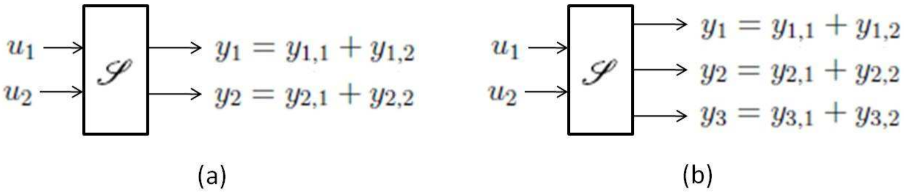

u_1
u_2
$\mathscr{S}$
$y_1 = y_{1,1} + y_{1,2}$
$y_2 = y_{2,1} + y_{2,2}$
(a)
u_1
u_2
$\mathscr{S}$
$y_1 = y_{1,1} + y_{1,2}$
$y_2 = y_{2,1} + y_{2,2}$
$y_3 = y_{3,1} + y_{3,2}$
(b)

Figure 5.1: Illustration of how unknown multiple excitation signals can cause an increase in output noise.

The contribution of this chapter is to analyze MIMO PTFs in terms of the conditions on the outputs under which a MIMO PTF can be defined. In particular, we consider the normal rank of the PTF as well as its order and relative degree. We also go beyond the results of [57] by considering the case in which the outputs are corrupted by noise that is not due to an excitation signal. Hence, we consider an errors-in-variables identification problem. To address this problem, we apply quadratically constrained least squares [50]. Since the results of [50] are confined to SISO systems, we apply the MIMO extension developed in Appendix B. In addition, we use the μ-Markov model structure rather than an ARMAX model structure since the former requires only a lower bound on the estimated model order.

5.2 Problem Formulation

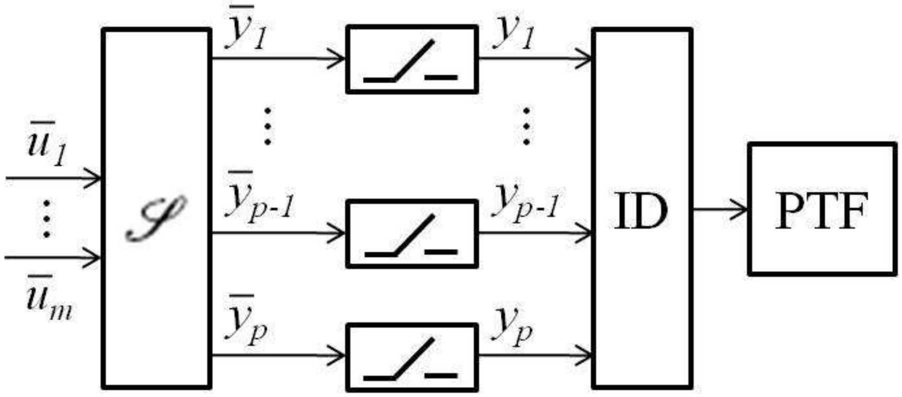


Figure 5.2: Pseudo transfer function (PTF) identification problem.

Consider the system $\mathscr{S}$ in Figure 5.2, which has unknown inputs $\bar{u}_1, \ldots, \bar{u}_m$ and outputs $\bar{y}_1, \ldots, \bar{y}_p$. We define $\bar{u} \triangleq \begin{bmatrix} \bar{u}_1 & \cdots & \bar{u}_m \end{bmatrix}^{\mathrm{T}}$ and $\bar{y} \triangleq \begin{bmatrix} \bar{y}_1 & \cdots & \bar{y}_p \end{bmatrix}^{\mathrm{T}}$. For $\bar{A} \in \mathbb{R}^{n\times n}$, $\bar{B} \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$, and $D \in \mathbb{R}^{p\times m}$, a state space representation of $\mathscr{S}$ is denoted by $(\bar{A}, \bar{B}, C, D)$, where

$$\dot{\bar{x}}(t) = \bar{A}\bar{x}(t) + \bar{B}\bar{u}(t), \quad \bar{x}(0) = x_0, \tag{5.1}$$

$$\bar{y}(t) = C\bar{x}(t) + D\bar{u}(t). \tag{5.2}$$

For $k \in \{0, 1, \ldots\}$ and sample interval $h > 0$, we apply the variation of constants method [106] to solve (5.1) for $\bar{x}(t)$ from $t = kh$ to $t = kh + h$, which yields

$$\bar{x}(kh + h) = e^{\bar{A}h}\bar{x}(kh) + \int_{kh}^{kh+h} e^{\bar{A}(kh+h-\tau)}\bar{B}\bar{u}(\tau)d\tau. \tag{5.3}$$

We assume that $\bar{u}(t)$ changes sufficiently slowly that $\bar{u}(kh) \approx \bar{u}(kh + h)$. Hence, we rewrite (5.3) as

$$x(k + 1) = Ax(k) + Bu(k), \tag{5.4}$$

where $x(k) \triangleq \bar{x}(kh)$, $u(k) \triangleq \bar{u}(kh)$, $A \triangleq e^{\bar{A}h}$, and

$$B \triangleq \int_0^h e^{\bar{A}\tau} d\tau \bar{B}.$$

We rewrite (5.4) in terms of the forward-shift operator $\mathbf{q}$ as

$$\delta(\mathbf{q})x = \operatorname{adj}\left(\mathbf{q}I - A\right) Bu, \tag{5.5}$$

where

$$\delta(\mathbf{q}) \triangleq \det(\mathbf{q}I - A)$$

and $\operatorname{adj}(\cdot)$ denotes the *adjugate* operator. Defining $y(k) \triangleq \bar{y}(kh)$, it follows from (5.2) that

$$y(k) = Cx(k) + Du(k). \tag{5.6}$$

For the remainder of this chapter, we assume $p > m$. Hence, substituting (5.5) into (5.6) yields

$$\delta(\mathbf{q})y = N(\mathbf{q})u, \tag{5.7}$$

where, for $U(\mathbf{q}) \in \mathbb{R}^{m\times m}[\mathbf{q}]$, $L(\mathbf{q}) \in \mathbb{R}^{(p-m)\times m}[\mathbf{q}]$, $C_U \in \mathbb{R}^{m\times n}$, $C_L \in \mathbb{R}^{(p-m)\times n}$, $D_U \in \mathbb{R}^{m\times m}$, and $D_L \in \mathbb{R}^{(p-m)\times m}$,

$$\begin{aligned} N(\mathbf{q}) &\triangleq C\mathrm{adj}\left(\mathbf{q}I - A\right) B + \delta(\mathbf{q})D \\ &= \begin{bmatrix} U(\mathbf{q}) \\ L(\mathbf{q}) \end{bmatrix} \\ &= \begin{bmatrix} C_U\mathrm{adj}\left(\mathbf{q}I - A\right) B + \delta(\mathbf{q})D_U \\ C_L\mathrm{adj}\left(\mathbf{q}I - A\right) B + \delta(\mathbf{q})D_L \end{bmatrix} \in \mathbb{R}^{p\times m}[\mathbf{q}]. \end{aligned} \tag{5.8}$$

We can also write (5.8) as

$$\begin{aligned} y &= G(\mathbf{q})u \\ &\triangleq \left[C\left(\mathbf{q}I - A\right)^{-1} B + D\right] u \\ &= \begin{bmatrix} G_U(\mathbf{q}) \\ G_L(\mathbf{q}) \end{bmatrix} u \\ &= \begin{bmatrix} C_U\left(\mathbf{q}I - A\right)^{-1} B + D_U \\ C_L\left(\mathbf{q}I - A\right)^{-1} B + D_L \end{bmatrix} u. \end{aligned}$$

5.3 Output-Only Model

For $1 \le i \le j \le p$, we define

$$y_{[i:j]} \triangleq \begin{bmatrix} y_i & \cdots & y_j \end{bmatrix}^{\mathrm{T}},$$

so that

$$y = \begin{bmatrix} y_{[1:m]} \\ y_{[m+1:p]} \end{bmatrix}.$$

It follows from (5.7) that

$$\delta(\mathbf{q}) y_{[1:m]} = U(\mathbf{q}) u \tag{5.9}$$

and

$$\delta(\mathbf{q}) y_{[m+1:p]} = L(\mathbf{q}) u. \tag{5.10}$$

Multiplying (5.9) on the left by $\operatorname{adj}(U(\mathbf{q}))$ yields

$$\delta(\mathbf{q}) \operatorname{adj}(U(\mathbf{q}))\, y_{[1:m]} = \det(U(\mathbf{q}))\, u. \tag{5.11}$$

Assuming $\det(U(\mathbf{q}))$ is not the zero polynomial and thus $U(\mathbf{q})$ is invertible, we substitute (5.11) into (5.10) to obtain

$$\delta(\mathbf{q}) \det(U(\mathbf{q}))\, y_{[m+1:p]} = \delta(\mathbf{q}) L(\mathbf{q}) \operatorname{adj}(U(\mathbf{q}))\, y_{[1:m]}.$$

Hence,

$$y_{[m+1:p]} = \Gamma(\mathbf{q}) y_{[1:m]},$$

where

$$\Gamma(\mathbf{q}) \triangleq \frac{\delta(\mathbf{q})}{\delta(\mathbf{q}) \det(U(\mathbf{q}))} L(\mathbf{q}) \operatorname{adj}(U(\mathbf{q})) \in \mathbb{R}^{(p-m)\times m}[\mathbf{q}] \tag{5.12}$$

is the MIMO PTF from $y_{[1:m]}$ to $y_{[m+1:p]}$. As explained in [58] for the case $m = 1$, the common factor $\delta(\mathbf{q})$ can be cancelled.

Note that

$$U(\mathbf{q}) = PQ(\mathbf{q}), \tag{5.13}$$

where

$$P \triangleq \begin{bmatrix} C_U & D_U \end{bmatrix} \in \mathbb{R}^{m\times(n+m)}$$

and

$$Q(\mathbf{q}) \triangleq \begin{bmatrix} \operatorname{adj}(\mathbf{q}I - A)\,B \\ \delta(\mathbf{q})I_m \end{bmatrix} \in \mathbb{R}^{(n+m)\times m}.$$

The following result uses (5.13) to provide a necessary condition for $\det U(\mathbf{q}) \neq 0$.

Proposition 5.3.1 If $\det U(\mathbf{q}) \neq 0$, then $\operatorname{rank} P = m$.

Proof 5.3.1

$$\begin{aligned} \operatorname{normal\,rank} U(\boldsymbol{q}) &= \operatorname{normal\,rank} PQ(\boldsymbol{q}) \\ &= \min\{\operatorname{rank} P, \operatorname{normal\,rank} Q(\boldsymbol{q})\} \\ &= \operatorname{rank} P = m. \end{aligned}$$

□

Note that

$$U(\mathbf{q}) = R(\mathbf{q})S, \tag{5.14}$$

where

$$R(\mathbf{q}) \triangleq \begin{bmatrix} C_U \mathrm{adj}\,(\mathbf{q}I - A) & \delta(\mathbf{q})I_m \end{bmatrix} \in \mathbb{R}^{m\times(n+m)}$$

and

$$S \triangleq \begin{bmatrix} B \\ D_U \end{bmatrix} \in \mathbb{R}^{(n+m)\times m}.$$

The following result uses (5.14) to provide a necessary condition for $\det U(\mathbf{q}) \neq 0$.

Proposition 5.3.2 If $\det U(\mathbf{q}) \neq 0$, then $\mathrm{rank}\, S = m$.

Proof 5.3.2

$$\begin{aligned} \mathrm{normal\,rank}\, U(\boldsymbol{q}) &= \mathrm{normal\,rank}\, R(\boldsymbol{q})S \\ &= \min\{\mathrm{normal\,rank}\, R(\boldsymbol{q}), \mathrm{rank}\, S\} \\ &= \mathrm{rank}\, S = m. \end{aligned} \qquad \square$$

The following result provides necessary and sufficient conditions for $\det U(\mathbf{q}) \neq 0$.

Proposition 5.3.3 $\det U(\mathbf{q}) \neq 0$ if and only if $G_U(\mathbf{q})$ has full normal rank.

Proof 5.3.3

$$\begin{aligned} \mathrm{normal\,rank}\, G_U(\boldsymbol{q}) &= \mathrm{normal\,rank}\, C_U \left(\boldsymbol{q}I - A\right)^{-1} B + D_U \\ &= \mathrm{normal\,rank}\, U(\boldsymbol{q}) = m. \end{aligned} \qquad \square$$

Propositions 5.3.1 and 5.3.2 provide necessary conditions for $\det U(\mathbf{q}) \neq 0$. However, the following example shows that these conditions are not sufficient for $G_U(\mathbf{q})$ to have full normal rank and thus for $\det U(q) \neq 0$.

Example 5.3.1 *Let*

$$A = -\frac{1}{4}\begin{bmatrix} 1 & -2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix}, \quad B = \begin{bmatrix} 2 & 2 \\ 0 & 1 \\ 1 & 0 \end{bmatrix},$$

$$C_U = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \quad D_U = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Note that A is asymptotically stable, B and C_U are full-rank, (A, B, C_U, D_U) is minimal, and $\operatorname{rank} P = \operatorname{rank} S = m$. However,

$$G_U(\boldsymbol{q}) = \frac{1}{\boldsymbol{q}^3 + \boldsymbol{q}^2 + \frac{9}{16}\boldsymbol{q} + \frac{5}{32}} \begin{bmatrix} -(\boldsymbol{q} + \frac{1}{4})(2\boldsymbol{q}+1) & -(\boldsymbol{q} + \frac{1}{2})(2\boldsymbol{q}+1) \\ \frac{1}{4}(\boldsymbol{q} + \frac{1}{4})(4\boldsymbol{q}-3) & \frac{1}{4}(\boldsymbol{q} + \frac{1}{2})(4\boldsymbol{q}-3) \end{bmatrix},$$

which does not have full normal rank.

5.4 MIMO PTF Order And Relative Degree

For $i = 1, \ldots, p$, we write

$$y_i(k) = C_i x(k) + D_i u(k),$$

and thus

$$\delta(\mathbf{q}) y_i = N_i(\mathbf{q}) u,$$

where $N_i(\mathbf{q})$ is the i^{th} row of $N(\mathbf{q})$, C_i is the i^{th} row of C, and D_i is the i^{th} row of D. For all $i \in \{1, \ldots, p\}$ and all $j \in \{1, \ldots, m\}$,

$$\eta_{i,j}(\mathbf{q}) \triangleq (N(\mathbf{q}))_{(i,j)} = C_i \text{adj}\,(\mathbf{q}I - A)\, B_j + D_{i,j}\delta(\mathbf{q}).$$

Proposition 5.4.1 Let $G_U(\mathbf{q})$ have full normal rank. Then

$$\deg\left(\det\left(U(\mathbf{q})\right)\right) \leq nm,$$

and, for all $i, j \in \{1, \ldots, m\}$,

$$\deg\left(\text{adj}\left(U(\mathbf{q})\right)_{(i,j)}\right) \leq n(m-1).$$

Proof 5.4.1 *For all $r \in \{1, \ldots, p\}$ the degree of $\eta_{r,j}(\boldsymbol{q})$ is n if $D_{r,j} \neq 0$ and $n-1$ otherwise (see Proposition 4.3.1). Hence, computing $\det U(\boldsymbol{q})$ using the cofactor expansion yields*

$$\begin{aligned}\deg\left(\det\left(U(\boldsymbol{q})\right)\right) &\leq m\left(\max_{i,j} \eta_{i,j}(\boldsymbol{q})\right)\\ &= mn.\end{aligned}$$

Furthermore,

$$\begin{aligned}\deg\left(\text{adj}\left(U(\boldsymbol{q})\right)_{(i,j)}\right) &\leq (m-1)\left(\max_{i,j} \eta_{i,j}(\boldsymbol{q})\right)\\ &= (m-1)n.\end{aligned}$$

□

Proposition 5.4.2 Let $G(\mathbf{q})$ have full normal rank. Then, for all $i \in \{1, \ldots, p-$

$m\}$ and all $j \in \{1, \ldots, m\}$,

$$\deg \left[L(\mathbf{q})\operatorname{adj}\left(U(\mathbf{q})\right)\right]_{(i,j)} \leq nm.$$

Proof 5.4.2 *For all $r \in \{1, \ldots, p\}$ the degree of $\eta_{r,j}(\boldsymbol{q})$ is n if $D_{r,j} \neq 0$ and $n-1$ otherwise, as shown in Proposition 4.3.1. Hence, for all $k \in \{1, \ldots, m\}$,*

$$\begin{aligned}\deg \left[L(\boldsymbol{q})\operatorname{adj}\left(U(\boldsymbol{q})\right)\right]_{(i,j)} &\leq \max_{i,j}\left(\deg \left[L(\boldsymbol{q})\right]_{(i,j)}\right) \\ &\quad + \max_{j,k}\left(\deg \left[\operatorname{adj}\left(U(\boldsymbol{q})\right)\right]_{(j,k)}\right) \\ &= n + n(m-1) \\ &= nm. \end{aligned}$$

□

Theorem 5.4.1 Let $G(\mathbf{q})$ have full normal rank and assume that the common factor $\delta(\mathbf{q})$ in (5.12) can be cancelled. Then, for all $i \in \{1, \ldots, p-m\}$ and for all $j \in \{1, \ldots, m\}$, the order of $\Gamma(\mathbf{q})_{(i,j)}$ is less than or equal to nm. Furthermore, the relative degree $d_{(i,j)}$ of $\Gamma(\mathbf{q})_{(i,j)}$ is given by

$$d_{(i,j)} \triangleq \deg \left[\det\left(U(\mathbf{q})\right)\right] - \deg \left[L(\mathbf{q})\operatorname{adj}\left(U(\mathbf{q})\right)\right]_{(i,j)}.$$

5.5 Three-Output, Two-Input Case

Let $p = 3$ and $m = 1$. From Theorem 4.4.1, the PTF from y_1 to y_3 is given by

$$y_3 = \frac{\eta_{3,1}(\mathbf{q})}{\eta_{1,1}(\mathbf{q})} y_1. \tag{5.15}$$

Next, let $m = 2$. Assuming $\delta(\mathbf{q})$ in (5.12) can be cancelled, the PTF from $Y_{[1:2]}$ to y_3 is given by

$$y_3 = \Gamma_{(1,1)}(\mathbf{q})y_1 + \Gamma_{(1,2)}y_2, \tag{5.16}$$

where

$$\Gamma_{(1,1)}(\mathbf{q}) = \frac{\eta_{3,1}(\mathbf{q})\eta_{2,2}(\mathbf{q}) - \eta_{3,2}(\mathbf{q})\eta_{2,1}(\mathbf{q})}{\eta_{1,1}(\mathbf{q})\eta_{2,2}(\mathbf{q}) - \eta_{2,1}(\mathbf{q})\eta_{1,2}(\mathbf{q})}$$

and

$$\Gamma_{(1,2)}(\mathbf{q}) = \frac{\eta_{3,2}(\mathbf{q})\eta_{1,1}(\mathbf{q}) - \eta_{3,1}(\mathbf{q})\eta_{1,2}(\mathbf{q})}{\eta_{1,1}(\mathbf{q})\eta_{2,2}(\mathbf{q}) - \eta_{2,1}(\mathbf{q})\eta_{1,2}(\mathbf{q})}.$$

It follows from (5.16) that, if two excitation signals are present, then the SISO PTF from y_1 to y_3 given by (5.15) is incorrect. In particular, (5.16) shows that both y_1 and y_2 contribute to y_3. Hence a MIMO PTF is needed to correctly characterize the output-output relationship.

5.6 Examples

Consider the mass-spring-damper structure in Figure 5.3, which has the equations of motion

$$M\ddot{q}(t) + C_{\mathrm{d}}\dot{q}(t) + Kq(t) = F(t), \tag{5.17}$$

where

$$q(t) = \begin{bmatrix} q_1(t) \\ q_2(t) \\ q_3(t) \end{bmatrix}, \quad M = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix},$$

$$C_\mathrm{d} = \begin{bmatrix} c_1 + c_2 & -c_2 & 0 \\ -c_2 & c_2 + c_3 & -c_3 \\ 0 & -c_3 & c_3 + c_4 \end{bmatrix},$$

$$K = \begin{bmatrix} k_1 + k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 + k_4 \end{bmatrix},$$

$$F(t) = \underline{B}u(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}.$$

We express (5.17) in state-space form (5.1), where $\bar{y}(t) = \begin{bmatrix} q_1(t) & q_2(t) & v_2(t) \end{bmatrix}^{\mathrm{T}}$,

$$\bar{x}(t) = \begin{bmatrix} q_1(t) \\ q_2(t) \\ q_3(t) \\ v_1(t) \\ v_2(t) \\ v_3(t) \end{bmatrix}, \quad \bar{A} \triangleq \begin{bmatrix} 0_{3\times 3} & I_3 \\ -M^{-1}K & -M^{-1}C_{\mathrm{d}} \end{bmatrix}, \quad \bar{B} \triangleq \begin{bmatrix} 0_{3\times 2} \\ M^{-1}\underline{B} \end{bmatrix},$$

and where $m_1 = \frac{1}{2}m_2 = \frac{1}{3}m_3 = 1$ kg, $k_1 = \frac{4}{5}k_2 = \frac{2}{3}k_3 = \frac{4}{7}k_4 = 4$ N/m, $c_1 = \frac{1}{2}c_2 = \frac{1}{3}c_3 = \frac{1}{4}c_3 = 0.1$ kg-m/s, and $h = 0.5$ s.

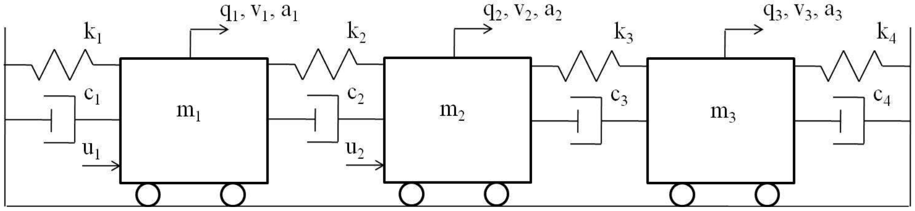


Figure 5.3: 3 DOF mass-spring-damper structure.

We use Markov parameter matrices to characterize the PTF estimate. To do this, we use the μ-Markov model structure

$$\begin{aligned} A_0 y_2(k) = & - A_\mu y_2(k-\mu) - \cdots - A_{\mu+n_{\mathrm{mod}}-1} y_2(k-\mu-n_{\mathrm{mod}}+1) \\ & + H_0 y_1(k) + \cdots + H_{\mu-1} y_1(k-\mu+1) \\ & + B_\mu y_1(k-\mu) + \cdots + B_{\mu+n_{\mathrm{mod}}-1} y_1(k-\mu-n_{\mathrm{mod}}+1) \end{aligned} \tag{5.18}$$

of order n_{mod}. The absence of terms involving $y_2(k-1), \ldots, y_2(k-\mu+1)$ is responsible for the explicit presence of the Markov parameter matrices $H_0, \ldots, H_{\mu-1}$.

The μ-Markov model structure has two principal advantages over the traditional

ARMAX structure. First, within the context of least squares identification with a white excitation signal, it is shown in [113] that the μ-Markov model provides consistent estimates of the Markov parameters in the presence of arbitrary output noise. Second, unlike parameter coefficients in an ARMAX model structure, the estimates of the Markov parameters are insensitive to the assumed model order n_{mod} as long as n_{mod} is larger than the true model order n. Consequently, only an upper bound on the true model order is needed.

For MIMO PTF identification, neither output signal is white, and thus we use quadratically constrained least squares (QCLS) with the μ-Markov model structure (5.18). As discussed in Appendix B and [50], if the noise autocorrelation matrices are known to within a scalar multiple, QCLS yields consistent parameter estimates in the presence of both input and output noise. If this assumption is not satisfied, instrumental variables methods [116] can be used to achieve consistent parameter estimates.

To quantify the difference between the estimated and actual Markov parameter matrices, we define

$$\varepsilon_{\mathcal{T}} \triangleq \frac{\|\mathcal{T} - \hat{\mathcal{T}}\|_2}{\|\mathcal{T}\|_2},$$

where

$$\mathcal{T} \triangleq \begin{bmatrix} \mathrm{vec}\,(H_0) \\ \vdots \\ \mathrm{vec}\,(H_3) \end{bmatrix}$$

and $\hat{\mathcal{T}}$ is the estimate of $\mathcal{T}$ obtained from QCLS.

5.6.1 Effect of model order with noise-free measurements

We investigate the effect of the μ-Markov model order $n_{\rm mod}$ on the accuracy of the estimated Markov parameter matrices of the PTF. We simulate (5.17) with $x_0 \neq 0$ to obtain the position q_1 of the first mass, the position q_2 of the second mass, and the velocity v_2 of the second mass, where u_1 and u_2 are realizations of white Gaussian processes with mean 0 and variance 1.

First, we consider the SISO PTF from q_1 to v_2 and use LS with the μ-Markov model structure (5.18) of relative degree 0 to estimate the first 4 (scalar) Markov parameters of the PTF from q_1 to v_2. For 10 realizations of 1000 samples of each u, we construct $\varepsilon_{\mathcal{T}}$ and compute the average estimation error $\bar{\varepsilon}_{\mathcal{T}}$ over all u. Plotting $\bar{\varepsilon}_{\mathcal{T}}$ as a function of the μ-Markov model order $n_{\rm mod}$ in Figure 5.4 shows that the Markov parameters are not correctly estimated for all $n_{\rm mod}$ from 1 to 15. This is expected because q_1 and v_2 are corrupted by contributions from both u_1 and u_2.

Next, we consider the two-input, one-output PTF from $\begin{bmatrix} q_1 & q_2 \end{bmatrix}^{\rm T}$ to v_2 and use LS with the μ-Markov model structure (5.18) of relative degree 0 to estimate the first 4 (2×2) Markov parameters of the PTF from $\begin{bmatrix} q_1 & q_2 \end{bmatrix}^{\rm T}$ to v_2. For 10 realizations of 1000 samples of each u, we construct $\varepsilon_{\mathcal{T}}$ and compute the average estimation error $\bar{\varepsilon}_{\mathcal{T}}$ over all u. Plotting $\bar{\varepsilon}_{\mathcal{T}}$ as a function of the μ-Markov model order $n_{\rm mod}$ in Figure 5.5 shows that the Markov parameters are correctly estimated for $n_{\rm mod} \geq 4$.

5.6.2 Consistency of the estimated MIMO PTF

We now investigate the effect of output noise on the accuracy of the identified PTF by adding noise to the output measurements. As shown in Figure 5.6, we assume that measurement y_i is corrupted by white, zero-mean Gaussian noise w_i and we assume that each measurement y_j is corrupted by white, zero-mean Gaussian noise v_{j-m}, where $i \in \{1, \ldots, m\}$ and $j \in \{m+1, \ldots, p\}$. Hence, the measured pseudo-inputs

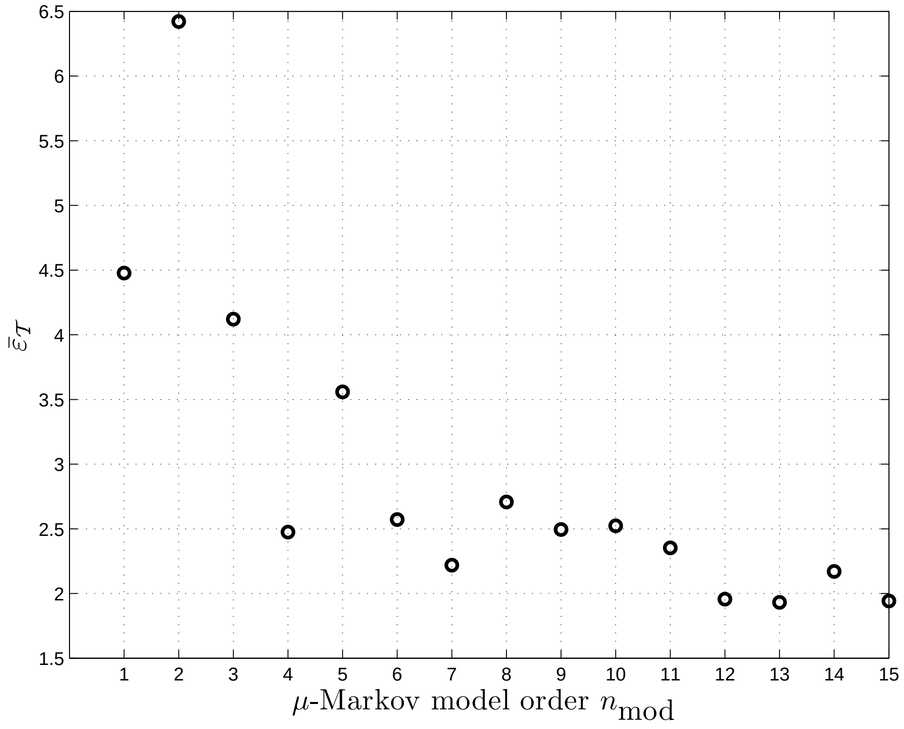


Figure 5.4: Error in the estimated Markov parameters of the SISO PTF from q_1 to v_2 as the μ-Markov model order increases.

$\xi_{[1:m]}$ are given by

$$\xi_{[1:m]} = y_{[1:m]} + v_{[1:m]},$$

and the measured pseudo-outputs $\xi_{[m+1:p]}$ are given by

$$\xi_{[m+1:p]} = y_{[m+1:p]} + w_{[1:p-m]}.$$

Since the pseudo inputs are not realizations of white random processes, LS does not yield consistent estimates of the Markov parameters. Hence, for comparison, we use both MIMO LS and MIMO QCLS with the μ-Markov model structure (5.18) with $n_{\text{mod}} = 4$ and relative degree 0 to estimate the first $\mu = 4$ Markov parameter matrices of the PTF from $\begin{bmatrix} q_1 & q_2 \end{bmatrix}^{\mathrm{T}}$ to v_2. To obtain consistent estimates, QCLS requires

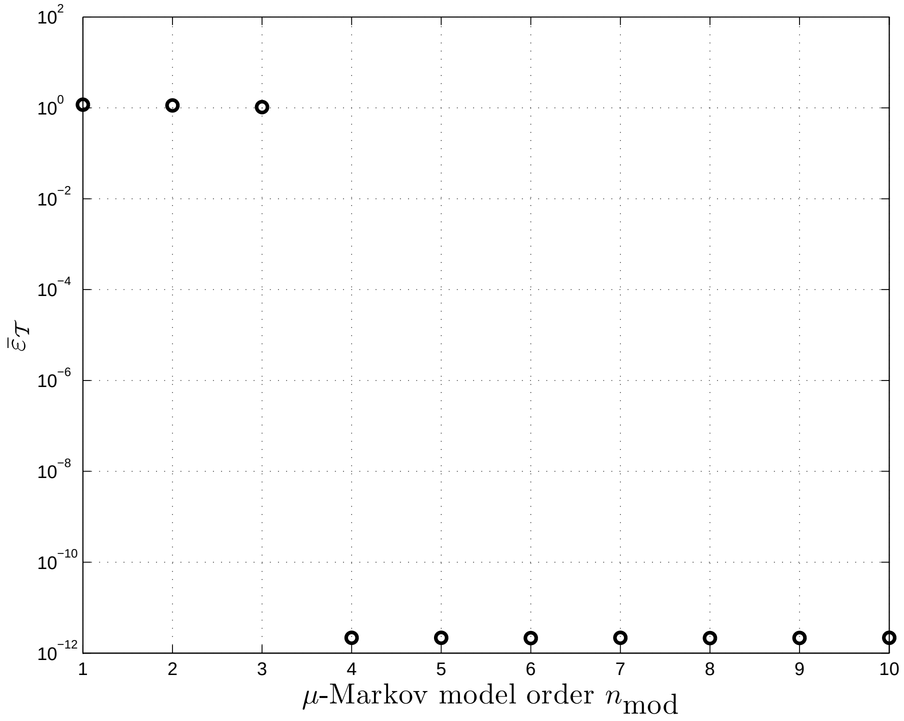


Figure 5.5: Error in the estimated Markov parameters of the MIMO PTF from $[q_1\ q_2]^{\mathrm{T}}$ to v_2 as the μ-Markov model order increases.

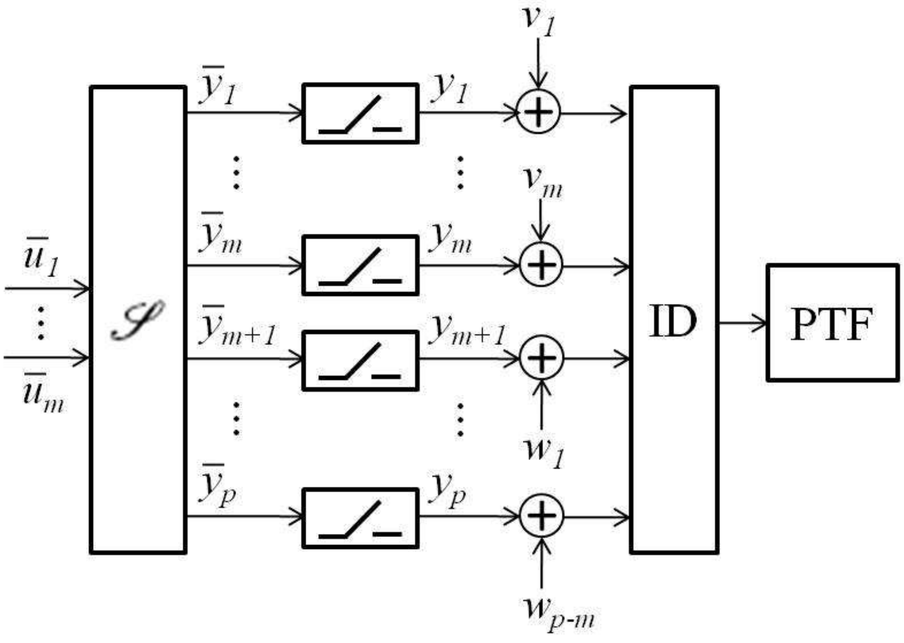


Figure 5.6: Effect of output noise on PTF identification.

knowledge of the noise autocorrelation to within a scalar multiple, as discussed in Appendix B and in [50].

We simulate (5.17) with $x_0 \neq 0$ to obtain the position q_1 of the first mass, the position q_2 of the second mass, and the velocity v_2 of the second mass, where u_1 and u_2 are realizations of white Gaussian processes with mean 0 and variance 1. We choose v_1, v_2, and w_1 so that the signal-to-noise ratio (SNR) of each measurement is 10. For LS and QCLS, we let $\mu = 4$ in (5.18) and estimate the Markov parameter matrices of the PTF from $\begin{bmatrix} q_1 & q_2 \end{bmatrix}^{\mathrm{T}}$ to v_2. We use the estimated and true Markov parameter matrices to compute $\varepsilon_{\mathcal{T}}$, which we average over 10 noise sequences and 10 input sequences u to obtain $\bar{\varepsilon}_{\mathcal{T}}$. The estimates provided by QCLS in Figure 5.7 appear consistent, while the estimates provided by LS do not. This result is expected since the pseudo-inputs q_1 and q_2 have colored spectra.

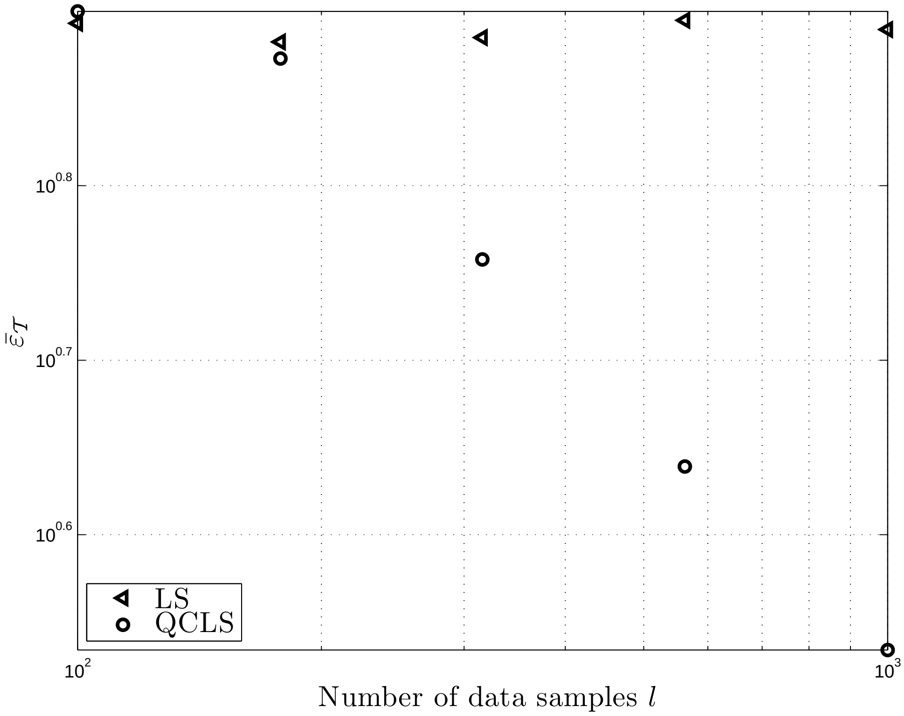


Figure 5.7: Consistent estimation of the PTF from $[q_1\ q_2]^{\mathrm{T}}$ to v_2, where white, uncorrelated noise is added to all sensor measurements

5.7 Conclusions

In this chapter, we defined a MIMO PTF to relate sets of output measurements and provided conditions on the number of excitations required to exactly estimate a MIMO PTF. We also provided necessary and sufficient conditions under which a MIMO PTF is defined, as well as lower bounds on the order and relative degree of the MIMO PTF. Using a simulated mass-spring-damper system, we demonstrated that a MIMO PTF can be estimated exactly if no noise is present in the system. By developing and applying a MIMO quadratically constrained least squares (QCLS) identification algorithm, we demonstrated that a MIMO PTF can be estimated consistently in the presence of output noise if the autocorrelation of the output noise is known to within a scalar multiple.

CHAPTER VI

Conclusions and Contributions

6.1 Conclusions

In this dissertation, we proposed several approaches for output-only fault detection.

First, we described the adjacent moving window peak detection (AMWPD) approach to detect an abrupt change in a noisy signal in real-time. We characterized the computational requirements of the AMWPD approach and provided techniques to reduce the computational complexity of the AMWPD approach. We also compared the AMWPD approach with existing techniques and showed that the AMWPD approach provides fewer false alarms and comparable detection speed.

Second, we applied a modified tabu search and probabilistic neural network (mTS + PNN) approach to classify tool wear in the shaving process. We defined novel features that we used in the mTS + PNN approach to yield improved classification accuracy. We also compared the mTS + PNN approach with an existing feature extraction and selection approach and showed that the mTS + PNN approach provides higher classification accuracy and requires less computational time.

Third, we characterized pseudo transfer functions (PTFs) that we use to detect faults in linear, time-invariant systems. For a system with one excitation and two outputs, we provided the order and relative degree of the single-input-single-output

(SISO) PTF and showed that the estimated PTF equals the true PTF if no noise is present in the data. We also applied quadratically-constrained least squares (QCLS) to estimate the PTF consistently for zero-mean output noise with known covariance. Finally, we showed that a change in the system dynamics can be reflected by a change in the estimated PTF.

Fourth, we extended the PTF approach to linear, time-invariant systems with multiple excitations and more than two outputs. We provided an upper bound on the order and relative degree of each entry of the multiple-input-multiple-output (MIMO) PTF and showed that the estimated MIMO PTF equals the true MIMO PTF if no noise is present in the data. Finally, we developed a MIMO QCLS technique to estimate the MIMO PTF consistently for zero-mean output noise with known covariance.

6.2 Contributions

This dissertation has four main contributions. First, we developed, validated, and implemented the AMWPD approach, which yields few false alarms and fast detection time, to detect an abrupt change in a noisy signal in real-time. Second, we defined key features that we used to classify the tool wear in a shaving process. We also applied the mTS + PNN feature selection and clustering approach to achieve fast and accurate tool wear classification. Third, we defined a SISO PTF, which requires minimal assumptions about the excitation and output noise, to detect a fault in a linear system that has a single excitation and two outputs. Fourth, we defined a MIMO PTF to detect a fault in a linear system that has non-unit-rank excitation and more than two outputs.

6.3 Proposed Future Work

Future work can be conducted in each area addressed by this dissertation. A short summary of some possible directions for future research is provided below.

The AMWPD approach should be tested on other applications with data from other types of output measurements. Also, the sensitivity of the AMWPD approach to the spectrum of the measurement noise must be investigated to quantify the robustness of the AMWPD approach to non-white, non-Gaussian noise. Finally, extensions of the AMWPD approach to real-time detection of other types of anomalies (i.e., an increase in variance or change in spectrum) could provide more comprehensive fault detection capability.

Due to time and cost constraints, the mTS + PNN approach was only validated on data from a single tool. Hence, data from additional tools should be used to confirm these results. Also, future research should investigate how many tools must be used for training to ensure sufficient capability to classify data from a tool that has not been used to train the mTS + PNN approach. Furthermore, the mTS + PNN approach could be tested on other types of manufacturing applications with other types of output measurements. Finally, future work should investigate the effect of the activation function in the pattern layer on the accuracy of the classification result, perhaps comparing the proposed activation function (3.18) with a hyperbolic tangent function or a radial basis function. Similarly, computational time and accuracy should be compared if the activation function in the output layer is nonlinear.

The sensitivity and robustness of the PTF fault detection approach must be compared with existing methods for fault detection, including statistical process control techniques, fault detection using standard system identification approaches, and feature extraction, selection, and classification approaches. Identification of PTFs for nonlinear systems, especially systems with Hammerstein or Wiener nonlinearities, as shown in Figure 6.1, is also a subject of future work. Furthermore, additional experi-

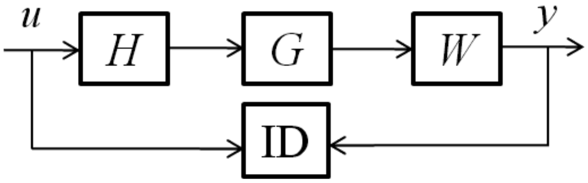


Figure 6.1: Identification of a system with linear dynamics G and static Hammerstein H and Wiener W nonlinearities.

mental validation of the PTF approach on aerospace, civil, and mechanical structures would motivate implementation and commercialization.

Consistent identification of PTFs is another area for future study. Although this dissertation considers both standard and quadratically-constrained approaches to batch least squares identification of time-series models, other identification approaches should also be considered, especially instrumental-variables [116]. Although we applied recursive least squares to identify a PTF online in [56], additional investigation into the consistency of online PTF identification techniques is required.

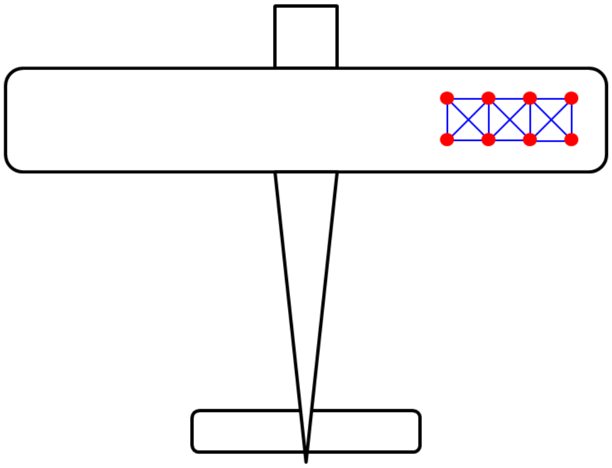


Figure 6.2: Illustration of a how a network of sensors can be used for fault localization.

Finally, fault isolation (damage localization) using the PTF approach is an area for future study, especially in structural health monitoring applications. As shown in Figure 6.2, a MIMO PTF can relate a large number of output measurements.

Assume a crack occurs in the wing in Figure 6.2. If the entry of the MIMO PTF relating output j to output k changes more than the other entries of the MIMO PTF, then the crack may have occurred between outputs j and k. However, simulated and experimental data must be used to examine this hypothesis.

APPENDICES

APPENDIX A

Proofs of SISO PTF Identities

Key Lemmas

We define

$$\alpha \triangleq \left[\begin{array}{cccc} \alpha_0 & \alpha_1 & \cdots & \alpha_{n-1} \end{array} \right]^{\mathrm{T}},$$

$$A_{\mathrm{c}} \triangleq SAS^{-1} = \left[\begin{array}{cc} 0_{(n-1)\times 1} & I_{n-1} \\ \multicolumn{2}{c}{-\alpha^{\mathrm{T}}} \end{array} \right],$$

$$B_{\mathrm{c}} \triangleq SB = \left[\begin{array}{cccc} 0 & \cdots & 0 & 1 \end{array} \right]^{\mathrm{T}}.$$

For $i = 1, 2$, we define

$$C_{\mathrm{c},i} \triangleq C_i S^{-1} = \left[\begin{array}{cccc} \beta_{i,0} & \beta_{i,1} & \cdots & \beta_{i,n-1} \end{array} \right] - \beta_{i,n}\alpha^{\mathrm{T}}.$$

For $j = 0, \ldots, n$, we define

$$\chi_j \triangleq \begin{cases} e_{j+1}^{\mathrm{T}}, & 0 \le j \le n-1, \\ -\alpha, & j = n, \end{cases}$$

where

$$e_k \triangleq \begin{bmatrix} 0_{(k-1)\times 1} \\ 1 \\ 0_{(n-k)\times 1} \end{bmatrix} \in \mathbb{R}^n.$$

Finally, for $i, j = 0, \ldots, n$, we define

$$f_{i,j} \triangleq \chi_i A_{\mathrm{c}}^j.$$

Lemma A.0.1 For all $i, j = 0, \ldots, n$,

$$f_{i,j} = f_{j,i}.$$

Proof A.0.1 *Note*

$$A_{\mathrm{c}}^j = \begin{cases} \begin{bmatrix} e_{j+1}^{\mathrm{T}} \\ \vdots \\ e_n^{\mathrm{T}} \\ -\alpha^{\mathrm{T}} \\ \vdots \\ -\alpha^{\mathrm{T}} A_{\mathrm{c}}^{j-1} \end{bmatrix}, & 0 \le j \le n-1, \\ \begin{bmatrix} -\alpha^{\mathrm{T}} \\ \vdots \\ -\alpha^{\mathrm{T}} A_{\mathrm{c}}^{j-1} \end{bmatrix}, & j = n. \end{cases}$$

If $i = j$, the result follows immediately.

If $0 \leq i \leq n-1$ and $j = n$,

$$f_{i,n} = e_{i+1}^{\mathrm{T}} \begin{bmatrix} -\alpha^{\mathrm{T}} \\ \vdots \\ -\alpha^{\mathrm{T}} A_{\mathrm{c}}^{j-1} \end{bmatrix} = -\alpha^{\mathrm{T}} A_{\mathrm{c}}^{i} = f_{n,i}.$$

If $0 \leq i \leq n-j-1$ and $0 \leq j \leq n-1$,

$$f_{i,j} = e_{i+1}^{\mathrm{T}} \begin{bmatrix} e_{j+1}^{\mathrm{T}} \\ \vdots \\ e_{n}^{\mathrm{T}} \\ -\alpha^{\mathrm{T}} \\ \vdots \\ -\alpha^{\mathrm{T}} A_{\mathrm{c}}^{j-1} \end{bmatrix} = e_{i+j+1}^{\mathrm{T}} = e_{j+1}^{\mathrm{T}} \begin{bmatrix} e_{i+1}^{\mathrm{T}} \\ \vdots \\ e_{n}^{\mathrm{T}} \\ -\alpha^{\mathrm{T}} \\ \vdots \\ -\alpha^{\mathrm{T}} A_{\mathrm{c}}^{i-1} \end{bmatrix} = f_{j,i}.$$

If $n-j \leq i \leq n-1$ and $0 \leq j \leq n-1$,

$$f_{i,j} = e_{i+1}^{\mathrm{T}} \begin{bmatrix} e_{j+1}^{\mathrm{T}} \\ \vdots \\ e_{n}^{\mathrm{T}} \\ -\alpha^{\mathrm{T}} \\ \vdots \\ -\alpha^{\mathrm{T}} A_{\mathrm{c}}^{j-1} \end{bmatrix} = -\alpha^{\mathrm{T}} A_{\mathrm{c}}^{i+j-n} = e_{j+1}^{\mathrm{T}} \begin{bmatrix} e_{i+1}^{\mathrm{T}} \\ \vdots \\ e_{n}^{\mathrm{T}} \\ -\alpha^{\mathrm{T}} \\ \vdots \\ -\alpha^{\mathrm{T}} A_{\mathrm{c}}^{i-1} \end{bmatrix} = f_{j,i}. \qquad \square$$

Proposition A.0.1

$$\sum_{i=0}^{n} \beta_{2,i} C_1 A^i = \sum_{i=0}^{n} \beta_{1,i} C_2 A^i. \tag{A.1}$$

Proof A.0.2 *Because (A, B) is controllable, from (4.10) there exists a nonsingular S that transforms A, B, C_1, and C_2 into controllable canonical form. Hence from*

Lemma A.0.1 we have

$$\begin{aligned}
\sum_{i=0}^{n} \beta_{2,i} C_1 A^i &= \sum_{i=0}^{n} \beta_{2,i} C_{\mathrm{c},1} A_{\mathrm{c}}^i S \\
&= \sum_{i=0}^{n} \beta_{2,i} \left[\sum_{j=0}^{n-1} \left(\beta_{1,j} e_{j+1}^{\mathrm{T}} \right) - \beta_{1,n} \alpha^{\mathrm{T}} \right] A_{\mathrm{c}}^i S \\
&= \sum_{i=0}^{n} \sum_{j=0}^{n} \beta_{2,i} \beta_{1,j} f_{j,i} S \\
&= \sum_{i=0}^{n} \sum_{j=0}^{n} \beta_{1,i} \beta_{2,j} f_{j,i} S \\
&= \sum_{i=0}^{n} \beta_{1,i} \left[\sum_{j=0}^{n-1} \left(\beta_{2,j} e_{j+1}^{\mathrm{T}} \right) - \beta_{2,n} \alpha^{\mathrm{T}} \right] A_{\mathrm{c}}^i S \\
&= \sum_{i=0}^{n} \beta_{1,i} C_{\mathrm{c},2} A_{\mathrm{c}}^i S \\
&= \sum_{i=0}^{n} \beta_{1,i} C_2 A^i \qquad \square
\end{aligned}$$

Proposition A.0.2 For all $k = 0, \ldots, n$,

$$\sum_{i=k}^{n} \beta_{2,i} H_{1,i-k} = \sum_{i=k}^{n} \beta_{1,i} H_{2,i-k}. \tag{A.2}$$

Proof A.0.3 *We consider the cases $k = n$, $k = n-1$, and $k = 0$. The remaining cases are proved in a similar fashion.*

For $k = n$,

$$\sum_{i=n}^{n} \beta_{2,i} H_{1,i-n} = \beta_{2,n} D_1 = \beta_{2,n} \beta_{1,n} = \beta_{1,n} D_2 = \sum_{i=n}^{n} \beta_{1,i} H_{2,i-n}.$$

For $k = n-1$,

$$\begin{aligned}
\sum_{i=n-1}^{n} \beta_{2,i} H_{1,i-n+1} &= \beta_{2,n-1} D_1 + \beta_{2,n} C_1 B \\
&= \beta_{2,n-1}\beta_{1,n} + \beta_{2,n} C_{\mathrm{c},1} B_{\mathrm{c}} \\
&= \beta_{2,n-1}\beta_{1,n} + \beta_{2,n}\beta_{1,n-1} - \beta_{2,n}\beta_{1,n}\alpha_{n-1} \\
&= \beta_{1,n-1}\beta_{2,n} + \beta_{1,n} C_{\mathrm{c},2} B_{\mathrm{c}} \\
&= \beta_{1,n-1} D_2 + \beta_{1,n} C_2 B \\
&= \sum_{i=n-1}^{n} \beta_{1,i} H_{2,i-n+1}.
\end{aligned}$$

For $k = 0$, we define

$$\xi_{i,j} \triangleq \begin{cases} 0, & i = 0, 0 \le j \le n-1, \\ 1, & i = 0, j = n, \\ e_{j+1}^{\mathrm{T}} A_{\mathrm{c}}^{i-1} e_n, & 1 \le i \le n, 0 \le j \le n-1, \\ -\alpha^{\mathrm{T}} A_{\mathrm{c}}^{i-1} e_n, & 1 \le i \le n, j = n, \end{cases}$$

and note that, using a proof similar to the proof of Lemma A.0.1, it can be shown that $\xi_{i,j} = \xi_{j,i}$. Hence,

$$\begin{aligned}
\sum_{i=0}^{n} \beta_{2,i} H_{1,i} &= \beta_{2,0}\beta_{1,n} + \beta_{2,1} C_{\mathrm{c},1} B_{\mathrm{c}} + \cdots + \beta_{2,n} C_{\mathrm{c},1} A_{\mathrm{c}}^{n-1} B_{\mathrm{c}} \\
&= \beta_{2,0}\beta_{1,n} + \sum_{i=1}^{n} \beta_{2,i} \left[\sum_{j=0}^{n-1} \beta_{1,j} e_{j+1}^{\mathrm{T}} - \beta_{1,n}\alpha^{\mathrm{T}} \right] A_{\mathrm{c}}^{i-1} B_{\mathrm{c}} \\
&= \sum_{i=0}^{n} \sum_{j=0}^{n} \beta_{2,i}\beta_{1,j}\xi_{i,j} \\
&= \sum_{i=0}^{n} \sum_{j=0}^{n} \beta_{1,i}\beta_{2,j}\xi_{i,j} \\
&= \beta_{1,0}\beta_{2,n} + \sum_{i=1}^{n} \beta_{1,i} \left[\sum_{j=0}^{n-1} \beta_{2,j} e_{j+1}^{\mathrm{T}} - \beta_{2,n}\alpha^{\mathrm{T}} \right] A_{\mathrm{c}}^{i-1} B_{\mathrm{c}} \\
&= \beta_{1,0}\beta_{2,n} + \beta_{1,1} C_{\mathrm{c},2} B_{\mathrm{c}} + \cdots + \beta_{1,n} C_{\mathrm{c},2} A_{\mathrm{c}}^{n-1} B_{\mathrm{c}} \\
&= \sum_{i=0}^{n} \beta_{1,i} H_{2,i}.
\end{aligned}$$

□

Proof of Proposition 4.4.1

Assume $l > 2n$. Then

$$N_2\Gamma_1 = \begin{bmatrix} \sum_{i=0}^{n} \beta_{2,i} C_1 A^i \\ \vdots \\ \sum_{i=0}^{n} \beta_{2,i} C_1 A^{l-n-1+i} \end{bmatrix}, \tag{A.3}$$

$$N_1\Gamma_2 = \begin{bmatrix} \sum_{i=0}^{n} \beta_{1,i} C_2 A^i \\ \vdots \\ \sum_{i=0}^{n} \beta_{1,i} C_2 A^{l-n-1+i} \end{bmatrix}. \tag{A.4}$$

From (A.1), it follows that the first component of (A.3) and the first component of (A.4) are identical. Multiplying (A.1) on the right by A^{q-1} implies that, for $q \in \{2, 3, \ldots, l-n\}$, the q^{th} component of (A.3) and the q^{th} component of (A.4) are also identical. □

Proof of Proposition 4.4.2

Assume $l > 2n$. Then

$$N_2\mathcal{H}_1 = \begin{bmatrix} \sigma_{2,1}(1,1) & \ldots & \sigma_{2,1}(1,n+1) & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \sigma_{2,1}(l-n,1) & \ldots & \sigma_{2,1}(1,1) & \ldots & \sigma_{2,1}(1,n+1) \end{bmatrix}$$

and

$$N_1\mathcal{H}_2 = \begin{bmatrix} \sigma_{1,2}(1,1) & \dots & \sigma_{1,2}(1,n+1) & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \sigma_{1,2}(l-n,1) & \dots & \sigma_{1,2}(1,1) & \dots & \sigma_{1,2}(1,n+1) \end{bmatrix}$$

are Toeplitz, where

$$\sigma_{j,k}(r,s) \triangleq \sum_{i=s-1}^{n} \beta_{j,i} H_{k,i+r-s}.$$

For $q \in \{0,1,\dots,l-n-2\}$, multiplying (A.1) on the right by A^qB implies that, for $r \in \{2,3,\dots,l-n\}$,

$$\sigma_{2,1}(r,1) = \sigma_{1,2}(r,1).$$

Furthermore, setting $k = s-1$ in (A.2) implies that, for $s \in \{1,2,\dots,n+1\}$,

$$\sigma_{2,1}(1,s) = \sigma_{1,2}(1,s). \qquad \square$$

Interesting Matrix Equality

We define

$$N \triangleq \begin{bmatrix} 0_{1\times(n-1)} & 0 \\ I_{n-1} & 0_{(n-1)\times 1} \end{bmatrix} \in \mathbb{R}^{n\times n}, \quad e_j \triangleq \begin{bmatrix} 0_{(j-1)\times 1} \\ 1 \\ 0_{(n-j)\times 1} \end{bmatrix} \in \mathbb{R}^{n}, \quad R_j \triangleq \begin{bmatrix} 0 & \cdots & 0 & 1 \\ \vdots & \iddots & \iddots & 0 \\ 0 & \iddots & \iddots & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{j\times j},$$

$$U_k \triangleq R_k N^k = \begin{bmatrix} R_{n-k} & 0_{(n-k)\times k} \\ 0_{k\times(n-k)} & 0_{k\times k} \end{bmatrix} \in \mathbb{R}^{n\times n}, \quad L_k \triangleq N^k R_k = \begin{bmatrix} 0_{k\times k} & 0_{k\times(n-k)} \\ 0_{(n-k)\times k} & R_{n-k} \end{bmatrix} \in \mathbb{R}^{n\times n}.$$

Furthermore, let $a \in \mathbb{C}^n$ denote the n^{th} column of A and, for $j = 1, \ldots, n$, define

$$Q_j \triangleq U_{n-j} + \sum_{i=1}^{n-1} \left(e_j^{\mathrm{T}} A^{i-1} a \right) L_i.$$

Lemma A.0.2 For every positive integer k,

$$A^k b = N^k b + \sum_{i=0}^{k-1} (e_n^{\mathrm{T}} N^i b) A^{k-i-1} a. \tag{A.5}$$

Proof A.0.4 We prove (A.5) by induction. For $k = 1$,

$$\begin{aligned} Ab &= (N + a e_n^{\mathrm{T}}) b \\ &= Nb + (e_n^{\mathrm{T}} b) a \\ &= Nb + (e_n^{\mathrm{T}} N^0 b) A^0 a, \end{aligned}$$

where $N^0 = A^0 = I_n$. From (A.5), we have

$$\begin{aligned} A^{k+1} b &= A(A^k b) \\ &= A \left[N^k b + \sum_{i=0}^{k-1} (e_n^{\mathrm{T}} N^i b) A^{k-i-1} a \right] \\ &= A N^k b + A \sum_{i=0}^{k-1} (e_n^{\mathrm{T}} N^i b) A^{k-i-1} a \\ &= \left(N + a e_n^{\mathrm{T}} \right) N^k b + \sum_{i=0}^{k-1} (e_n^{\mathrm{T}} N^i b) A^{k-i} a \\ &= N^{k+1} b + (e_n^{\mathrm{T}} N^k b) A^0 a + \sum_{i=0}^{k-1} (e_n^{\mathrm{T}} N^i b) A^{k-i} a \\ &= N^{k+1} b + \sum_{i=0}^{k} (e_n^{\mathrm{T}} N^i b) A^{k-i} a. \end{aligned}$$

□

Proposition A.0.3 Let $b, c \in \mathbb{C}^n$ and let

$$A = \begin{bmatrix} 0 & 0 & \cdots & 0 & a_1 \\ 1 & 0 & \cdots & 0 & a_2 \\ 0 & 1 & \cdots & 0 & a_3 \\ \vdots & \vdots & \ddots & & \vdots \\ 0 & 0 & \cdots & 1 & a_n \end{bmatrix} \in \mathbb{C}^{n \times n}.$$

Then

$$\begin{bmatrix} b & Ab & \cdots & A^{n-1}b \end{bmatrix} c = \begin{bmatrix} c & Ac & \cdots & A^{n-1}c \end{bmatrix} b.$$

Proof A.0.5 *From Lemma A.0.2, we have*

$$
\begin{aligned}
&\begin{bmatrix} b & Ab & \cdots & A^{n-1}b \end{bmatrix} c \\
&= \begin{bmatrix} b & Nb + (e_n^{\mathrm{T}}b)a & \cdots & N^{n-1}b + \sum_{i=0}^{n-2}(e_n^{\mathrm{T}}N^i b)A^{n-i-2}a \end{bmatrix} c \\
&= \begin{bmatrix} b & Nb & \cdots & N^{n-1}b \end{bmatrix} c + \begin{bmatrix} 0_{n\times 1} & (e_n^{\mathrm{T}}b)a & \cdots & \sum_{i=0}^{n-2}(e_n^{\mathrm{T}}N^i b)A^{n-i-2}a \end{bmatrix} c \\
&= \begin{bmatrix} b_1 & 0 & \cdots & 0 \\ b_2 & b_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ b_n & \cdots & b_2 & b_1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} + \begin{bmatrix} 0_{n\times 1} & (e_n^{\mathrm{T}}b)a & \cdots & (e_n^{\mathrm{T}}N^{n-2}b)a \end{bmatrix} c \\
&\qquad + \begin{bmatrix} 0_{n\times 2} & (e_n^{\mathrm{T}}b)Aa & \cdots & (e_n^{\mathrm{T}}N^{n-3}b)Aa \end{bmatrix} c + \cdots + \begin{bmatrix} 0_{n\times (n-1)} & (e_n^{\mathrm{T}}b)A^{n-2}a \end{bmatrix} c \\
&= \begin{bmatrix} b_1 & 0 & \cdots & 0 \\ b_2 & b_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ b_n & \cdots & b_2 & b_1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} + \begin{bmatrix} 0_{n\times 1} & b_n a & \cdots & b_2 a \end{bmatrix} c \\
&\qquad + \begin{bmatrix} 0_{n\times 2} & b_n Aa & \cdots & b_3 Aa \end{bmatrix} c + \cdots + \begin{bmatrix} 0_{n\times (n-1)} & b_n A^{n-2}a \end{bmatrix} c \\
&= \begin{bmatrix} b_1c_1 \\ b_2c_1 + b_1c_2 \\ \vdots \\ b_nc_1 + \cdots + b_1c_n \end{bmatrix} + (b_nc_2 + \cdots + b_2c_n)a + (b_nc_3 + \cdots + b_3c_n)Aa + \cdots + b_nc_nA^{n-2}a \\
&= \begin{bmatrix} b^{\mathrm{T}}U_{n-1}c \\ b^{\mathrm{T}}U_{n-2}c \\ \vdots \\ b^{\mathrm{T}}U_0c \end{bmatrix} + \sum_{i=1}^{n-1}(b^{\mathrm{T}}L_ic)A^{i-1}a \quad = \begin{bmatrix} b^{\mathrm{T}}U_{n-1}c + b^{\mathrm{T}}\left(\sum_{i=1}^{n-1}\left(e_1^{\mathrm{T}}A^{i-1}a\right)L_i\right)c \\ b^{\mathrm{T}}U_{n-2}c + b^{\mathrm{T}}\left(\sum_{i=1}^{n-1}\left(e_2^{\mathrm{T}}A^{i-1}a\right)L_i\right)c \\ \vdots \\ b^{\mathrm{T}}U_0c + b^{\mathrm{T}}\left(\sum_{i=1}^{n-1}\left(e_n^{\mathrm{T}}A^{i-1}a\right)L_i\right)c \end{bmatrix} \\
&= \begin{bmatrix} b^{\mathrm{T}}Q_1c \\ \vdots \\ b^{\mathrm{T}}Q_nc \end{bmatrix} \quad = \begin{bmatrix} c & Ac & \cdots & A^{n-1}c \end{bmatrix} b,
\end{aligned}
$$

where the last equality follows from the fact that $Q_1, \ldots, Q_n$ *are symmetric.*

APPENDIX B

MIMO QCLS with a μ-Markov Model Structure

Problem Formulation

We consider identification of a MIMO system, where the noise-free input is $u_0(k) \in \mathbb{R}^m$ and the noise-free output is $y_0(k) \in \mathbb{R}^p$. Hence,

$$A(\mathbf{q})y_0 = B(\mathbf{q})u_0 \tag{B.1}$$

with transfer function

$$G(\mathbf{q}) \triangleq A^{-1}(\mathbf{q})B(\mathbf{q}),$$

where $A \in \mathbb{R}^{p\times p}[\mathbf{q}]$ has full normal rank, $B \in \mathbb{R}^{p\times m}[\mathbf{q}]$, $k = 0, \ldots, l$ is a positive integer, A and B are left coprime, and every entry of $G(\mathbf{q})$ is a proper function of $\mathbf{q}$.

We write (B.1) as a μ-Markov time-series model

$$\begin{aligned} 0 = {} & \alpha_0 y_0(k) + \alpha_\mu y_0(k-\mu) + \cdots + \alpha_{k'} y_0(k-k') \\ & - H_0 u_0(k) - \cdots - H_{\mu-1} u_0(k-\mu+1) \\ & - B_\mu u_0(k-\mu) - \cdots - B_{k'} u_0(k-k'), \end{aligned} \tag{B.2}$$

where

$$k' \triangleq \mu + n - 1,$$

and for $i = 0, \mu, \mu+1, \ldots, k'$, $\alpha_i \in \mathbb{R}$.

Applying the vec operator to (B.2), we obtain

$$\begin{aligned} 0 = {} & \alpha_0 y_0(k) + \alpha_\mu y_0(k-\mu) + \cdots + \alpha_{k'} y_0(k-k') \\ & - \left(u_0^{\mathrm{T}}(k) \otimes I_p\right) \operatorname{vec}(H_0) - \cdots - \left(u_0^{\mathrm{T}}(k-\mu+1) \otimes I_p\right) \operatorname{vec}(H_{\mu-1}) \\ & - \left(u_0^{\mathrm{T}}(k-\mu) \otimes I_p\right) \operatorname{vec}(B_\mu) - \cdots - \left(u_0^{\mathrm{T}}(k-k') \otimes I_p\right) \operatorname{vec}(B_{k'}). \end{aligned} \tag{B.3}$$

Errors-In-Variables Formulation

We write (B.3) as

$$\phi_0^{\mathrm{T}}(k)\theta = 0,$$

where

$$\phi_0(k) \triangleq \begin{bmatrix} y_0^{\mathrm{T}}(k) \\ y_0^{\mathrm{T}}(k-\mu) \\ \vdots \\ y_0^{\mathrm{T}}(k-k') \\ -u_0(k)\otimes I_p \\ \vdots \\ -u_0(k-k')\otimes I_p \end{bmatrix} \in \mathbb{R}^{[pm(n+\mu)+n+1]\times p}$$

and

$$\theta \triangleq \begin{bmatrix} \alpha_0 \\ \alpha_\mu \\ \vdots \\ \alpha_{k'} \\ \mathrm{vec}(H_0) \\ \vdots \\ \mathrm{vec}(H_{\mu-1}) \\ \mathrm{vec}(B_\mu) \\ \vdots \\ \mathrm{vec}(B_{k'}) \end{bmatrix} \in \mathbb{R}^{pm(n+\mu)+n+1}.$$

Assuming $l \geq k''$, we define

$$\Phi_0 \triangleq \begin{bmatrix} \phi_0^{\mathrm{T}}(k') \\ \vdots \\ \phi_0^{\mathrm{T}}(l) \end{bmatrix} \in \mathbb{R}^{p(l-\mu-n+2)\times[pm(n+\mu)+n+1]}$$

so that

$$\Phi_0\theta = 0,$$

where

$$k'' \triangleq (\mu + n)(m+1) + \frac{n+1}{p} - 2.$$

Next, we assume that $y_0(k)$ is corrupted by output noise $w(k)$ and we assume that $u_0(k)$ is corrupted by input noise $v(k)$ so that the measured output $y(k)$ is given by

$$y(k) = y_0(k) + w(k),$$

and the measured input $u(k)$ is given by

$$u(k) = u_0(k) + v(k).$$

Hence,

$$\begin{aligned} y(k) &= G(\mathbf{q})u_0(k) + w(k) \\ &= G(\mathbf{q})u(k) - G(\mathbf{q})v(k) + w(k). \end{aligned} \tag{B.4}$$

We write (B.4) in regression form

$$A(\mathbf{q})y(k) - B(\mathbf{q})u(k) = A(\mathbf{q})w(k) - B(\mathbf{q})v(k),$$

or

$$\phi^{\mathrm{T}}(k)\theta = \psi^{\mathrm{T}}(k)\theta,$$

where

$$\phi(k) = \phi_0(k) + \psi(k),$$

and

$$\psi(k) \triangleq \begin{bmatrix} w^{\mathrm{T}}(k) \\ w^{\mathrm{T}}(k-\mu) \\ \vdots \\ w^{\mathrm{T}}(k-k') \\ -v(k) \otimes I_p \\ \vdots \\ -v(k-k') \otimes I_p \end{bmatrix} \in \mathbb{R}^{[pm(n+\mu)+n+1]\times p}.$$

The regression matrix Φ is given by

$$\Phi = \Phi_0 + \Psi,$$

where

$$\Psi \triangleq \begin{bmatrix} \psi^{\mathrm{T}}(\mu+n-1) \\ \vdots \\ \psi^{\mathrm{T}}(l) \end{bmatrix} \in \mathbb{R}^{p(l-\mu-n+2)\times[pm(n+\mu)+n+1]}.$$

Definition B.0.1 For all $l \geq k''$, the input sequence $\{u_0(k)\}_{k=k''}^{l}$ is *persistently exciting* for $G(\mathbf{q})$ if $\operatorname{rank}\Phi_0 = pm(n+\mu)+n$.

Definition B.0.2 For all $l \geq k''$, the input sequence $\{u_0(k)\}_{k=k''}^{l}$ and the noise sequences $\{w(k)\}_{k=k''}^{l}$ and $\{v(k)\}_{k=k''}^{l}$ are *jointly persistently exciting* for $G(\mathbf{q})$ if $\operatorname{rank}\Phi = pm(n+\mu)+n+1$

For $\kappa \geq k'$, let $\sigma_{pm(n+\mu)+n,\kappa}$ be the second-smallest singular value of $\begin{bmatrix} \phi_0^{\mathrm{T}}(\kappa) \\ \vdots \\ \phi_0^{\mathrm{T}}(k'') \end{bmatrix}$.

Definition B.0.3 For all $l \geq k''$, the sequence $\{u_0(k)\}_{k=k''}^{l}$ is *infinitely persistently exciting* for $G(\mathbf{q})$ if there exists $\varepsilon > 0$ such that, for all $\kappa \geq k'$, $\sigma_{pm(n+\mu)+n,\kappa} > \varepsilon$.

QCLS for MIMO Systems with Known Noise Covariance

The QCLS problem is given by

$$\min_{\hat{\theta}\in\mathcal{D}(N)} \mathcal{J}(\hat{\theta}), \tag{B.5}$$

where $N \in \mathbb{R}^{[pm(n+\mu)+n+1]\times[pm(n+\mu)+n+1]}$ is symmetric and

$$\mathcal{D}(N) \triangleq \left\{\hat{\theta} \in \mathbb{R}^{pm(n+\mu)+n+1} : \hat{\theta}^{\mathrm{T}} N \hat{\theta} = 1\right\}.$$

If $N = N_{LS}$, solutions of the QCLS problem (B.5) are solutions of the standard least squares problem with $\alpha_0 = \pm 1$, where

$$N_{LS} \triangleq \begin{bmatrix} 1 & 0_{1\times[pm(n+\mu)+n]} \\ 0_{[pm(n+\mu)+n]\times 1} & 0_{[pm(n+\mu)+n]\times[pm(n+\mu)+n]} \end{bmatrix}.$$

Note that the $\hat{\theta}$ which solves the QCLS problem (B.5) corresponds to the generalized eigenvector associated with the smallest positive generalized eigenvalue of (M, N), where

$$M \triangleq \frac{1}{l}\Phi^{\mathrm{T}}\Phi.$$

For the remainder of this chapter, we assume that $\{w(k)\}_{k=0}^{\infty}$ and $\{v(k)\}_{k=0}^{\infty}$ are stationary, have finite second moments, and are jointly ergodic random processes in the sense that, for all $\rho, \sigma \in \{0, 1, 2\}$ and for all i,

$$\mathbb{E}[w^\rho(i)v^\sigma(i)] = \lim_{l\to\infty} \frac{1}{l} \sum_{k=0}^{l} w^\rho(k)v^\sigma(k) \,\mathrm{wp1}.$$

For $k' \le k \le l$, we define

$$R \triangleq \mathbb{E}[\psi(k)\psi^{\mathrm{T}}(k)] = \begin{bmatrix} R_{ww} & -R_{wv} \\ -R_{vw} & R_{vv} \end{bmatrix} \tag{B.6}$$

and note that R is positive semi-definite and, since $w(k)$ and $v(k)$ are stationary, R is independent of k.

We define

$$M_0 \triangleq \frac{1}{l}\Phi_0^{\mathrm{T}}\Phi_0,$$

$$\mathscr{M}_0 \triangleq \lim_{l\to\infty} M_0,$$

and

$$\mathscr{M} \triangleq \lim_{l\to\infty} M,$$

where the limits are defined when they exist.

Proposition B.0.4 Assume that $\{u_0(k)\}_{k=0}^{l}$ and $\{y_0(k)\}_{k=0}^{l}$ are bounded and satisfy (B.1). Also assume that $w(k)$ and $v(k)$ have zero mean. For all $l \ge k''$, assume that $\{u_0(k)\}_{k=k''}^{l}$ is persistently exciting for $G(\mathbf{q})$ and assume that $\{u_0(k)\}_{k=k''}^{l}$, $\{w(k)\}_{k=k''}^{l}$, and $\{v(k)\}_{k=k''}^{l}$ are jointly persistently exciting for $G(\mathbf{q})$. Finally, assume that $M_0 + R > 0$. Then, for all $\eta > 0$, QCLS with $N = \eta\bar{R}$ provides an unbiased estimate $\hat{\theta}$ of θ.

Proof B.0.6 *The proof is analogous to the proof of Proposition 9.3 in [50].* □

Proposition B.0.5 Assume that $\mathscr{M}_0$ exists. Assume that $\{u_0(k)\}_{k=0}^{l}$ and $\{y_0(k)\}_{k=0}^{l}$ are bounded and satisfy (B.1). Also assume that $w(k)$ and $v(k)$ have zero mean. For all $l \geq k''$, assume that $\{u_0(k)\}_{k=k''}^{l}$ is infinitely persistently exciting for $G(\mathbf{q})$ and assume that $\{u_0(k)\}_{k=k''}^{l}$, $\{w(k)\}_{k=k''}^{l}$, and $\{v(k)\}_{k=k''}^{l}$ are jointly persistently exciting for $G(\mathbf{q})$. Finally, assume that $\mathscr{M} > 0$. Then, for all $\eta > 0$, QCLS with $N = \eta\bar{R}$ provides a consistent estimate $\hat{\theta}$ of θ.

Proof B.0.7 *The proof is analogous to the proof of Theorem 10.11 in [50].* □

Example: Noise with Known Covariance

Consider the two-input, one-output, asymptotically stable, non-minimum phase system with minimal realization (A,B,C,D) given by

$$A = \begin{bmatrix} 0 & 1 \\ -0.7 & 0.1 \end{bmatrix}, \quad B = I_2,$$
$$C = \begin{bmatrix} 0.3 & 1 \end{bmatrix}, \quad D = 0_{1\times 2}. \tag{B.7}$$

To quantify the difference between the estimated and actual Markov parameter matrices, we define

$$\varepsilon_{\mathcal{T}} \triangleq \frac{\|\mathcal{T} - \hat{\mathcal{T}}\|_2}{\|\mathcal{T}\|_2},$$

where

$$\mathcal{T} \triangleq \begin{bmatrix} \operatorname{vec}(H_0) \\ \vdots \\ \operatorname{vec}(H_3) \end{bmatrix}$$

and $\hat{\mathcal{T}}$ is an estimate of $\mathcal{T}$.

We investigate the effect of noise on the accuracy of the identified MIMO TF. We simulate (B.4) with

$$G(\mathbf{q}) = C\left(qI - A\right)^{-1}B + D,$$

where A, B, C, and D are given by (B.7), u_1 and u_2 are realizations of white Gaussian processes with mean 0 and variance 1, and we choose v_1, v_2, and w so that the signal-to-noise ratio (SNR) is 3. For comparison, we estimate the Markov parameter matrices using standard least squares (LS) and QCLS with known noise covariance

$$R = \begin{bmatrix} 0.385I_3 & 0_{3\times 12} \\ 0_{12\times 3} & 0.111I_{12} \end{bmatrix}.$$

For LS and QCLS, we let $\mu = 4$ and estimate the Markov parameter matrices of the TF. We use the estimated and true Markov parameter matrices to compute $\varepsilon_{\mathcal{T}}$, which we average over 10 noise sequences and 5 input sequences to obtain $\bar{\varepsilon}_{\mathcal{T}}$. Figure B.1 shows that QCLS provides consistent estimates of the TF, while LS does not. For the result shown in Figure B.1, we note that $x_0 \neq 0$ and $n_{\text{mod}} = 4$.

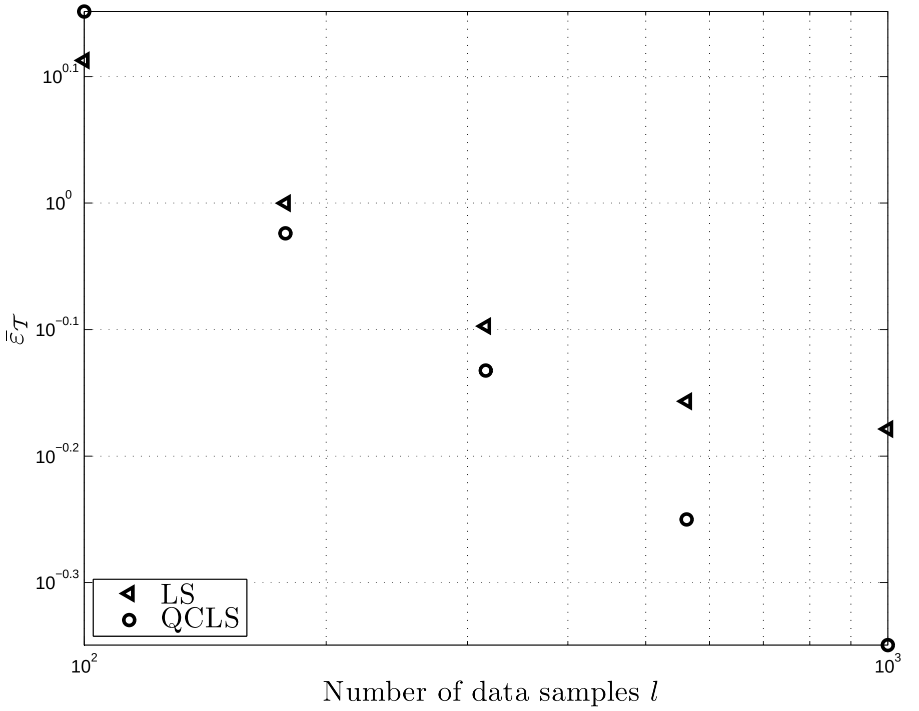


Figure B.1: Error in the estimated TF using MIMO μ-Markov QCLS, where zero-mean, white, uncorrelated noise is added to u_1, u_2, and y.

APPENDIX C

Conditions for Consistent Estimates

Problem Formulation

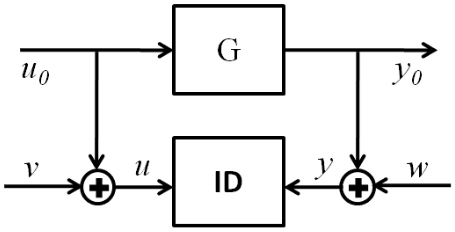


Figure C.1: Errors-in-variables approach to system identification.

We consider the identification problem shown in Figure C.1, where the true input u_0 and true output y_0 are corrupted by input noise v and output noise w, respectively, so that the measured input $u = u_0 + v$ and the measured output $y = y_0 + w$. We assume that G has order n and is linear, time-invariant, stable, and causal. We also assume that the number of data samples $l \gg n$ and only consider batch (not recursive) least squares (LS) and quadratically-constrained least squares (QCLS).

Consistent Estimates Using LS

First, we consider estimation of the coefficients $a_0, \ldots, a_n, b_0, \ldots, b_n$ of the autoregressive moving average (ARMA) time-series model

$$a_0 y_0(k) + \cdots + a_n y_0(k-n) = b_0 u_0(k) + \cdots + b_n u_0(k-n). \tag{C.1}$$

For consistent and unbiased estimates of the coefficients, the estimated model order $\hat{n} = n$ and $v(k) \equiv 0$. Furthermore, one of the following conditions must be satisfied

- *White equation error:* The data satisfies

$$a_0 y(k) + \cdots + a_n y(k-n) = b_0 u(k) + \cdots + b_n u(k-n) + w(k). \tag{C.2}$$

- *Finite impulse response:* The data satisfies

$$a_0 y(k) = b_0 u(k) + \cdots + b_n u(k-n) + a_0 w(k). \tag{C.3}$$

Second, we consider estimation of the first μ Markov parameters $H_0, \ldots, H_{\mu-1}$ of the μ-Markov time-series model [117]

$$\begin{aligned} 0 = {} & a_0 y_0(k) + a_\mu y_0(k-\mu) + \cdots + a_{k'} y_0(k-k') \\ & - H_0 u_0(k) - \cdots - H_{\mu-1} u_0(k-\mu+1) \\ & - b_\mu u_0(k-\mu) - \cdots - b_{k'} u_0(k-k'), \end{aligned} \tag{C.4}$$

where

$$k' \triangleq \mu + n - 1.$$

For consistent and unbiased estimates of the Markov parameters, the following conditions must be satisfied

- $\hat{n} \geq n$.
- $v(k) \equiv 0$.
- u_0 is a realization of a white random process.
- w is a realization of a stationary random process.

Consistent Estimates Using QCLS

First, we consider estimation of the coefficients of the ARMA time series model (C.1). For consistent and unbiased estimates of the coefficients, the following conditions must be satisfied

- $\hat{n} = n$.
- w and v are stationary, have zero mean and finite second moments, and arise from jointly ergodic random processes.
- u_0 is infinitely persistently exciting (see Definition B.0.3).
- u_0, v, and w are jointly persistently exciting exciting (see Definition B.0.2).
- The orders of the coloring filters of v and w are less than $\hat{n} + 1$.
- The noise covariance R, defined in (B.6), is known to within a scalar multiple.

Second, we consider estimation of the first μ Markov parameters $H_0, \ldots, H_{\mu-1}$ of the μ-Markov time-series model (C.4). For consistent and unbiased estimates of the coefficients, the following conditions must be satisfied

- $\hat{n} \geq n$ (the case $\hat{n} > n$ is under investigation).

- w and v are stationary, have zero mean and finite second moments, and arise from jointly ergodic random processes.

- u_0 is infinitely persistently exciting (see Definition B.0.3).

- u_0, v, and w are jointly persistently exciting exciting (see Definition B.0.2).

- The orders of the coloring filters of v and w are less than $\hat{n}+1$.

- The noise covariance R, defined in (B.6), is known to within a scalar multiple.

Conditions under which Solution to Discrete-Time Lyapunov Equation is Toeplitz

Proposition C.0.6 If R is symmetric and if $\lambda_i(A) \neq 1/\lambda_j(A^{\mathrm{T}})$ for all i, j, then the solution Q to the discrete-time Lyapunov equation

$$Q = AQA^{\mathrm{T}} + R \tag{C.5}$$

exists and is unique and symmetric.

Proof C.0.8 Since $\lambda_i(A) \neq 1/\lambda_j(A^{\mathrm{T}})$ for all i, j, the solution Q to (C.5) exists and is unique [118]. We show that Q is symmetric by contradiction. Hence, we assume $R \neq R^{\mathrm{T}}$ and $Q = Q^{\mathrm{T}}$. From (C.5), we have

$$\begin{aligned} R^{\mathrm{T}} &= Q^{\mathrm{T}} - AQ^{\mathrm{T}}A^{\mathrm{T}}, \\ &= Q - AQA^{\mathrm{T}}, \\ &= R, \end{aligned}$$

which is a contradiction. □

Proposition C.0.7 If A and B are written in controllable-canonical form, that is,

$$A \triangleq \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ a_1 & a_2 & a_3 & \cdots & a_n \end{bmatrix}, \quad B \triangleq \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix},$$

where $\{a_i\}_{i=1}^n \in \mathbb{R}$, and if $\lambda_i(A) \neq 1/\lambda_j(A)$ for all i, j, then the solution Q to the discrete-time Lyapunov equation

$$Q = AQA^{\mathrm{T}} + BB^{\mathrm{T}} \tag{C.6}$$

exists and is unique, symmetric, and Toeplitz.

Proof C.0.9 From Proposition C.0.6, it follows that the solution Q to (C.6) exists and is unique and symmetric. From (C.6), we have

$$\begin{bmatrix} q_{1,1} & \cdots & q_{1,n} \\ \vdots & \vdots & \vdots \\ q_{n-1,1} & \cdots & q_{n-1,n} \\ q_{n,1} & \cdots & q_{n,n}-1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ a_1 & a_2 & a_3 & \cdots & a_n \end{bmatrix} \begin{bmatrix} q_{1,2} & \cdots & q_{1,n} & \sum_{i=1}^n a_i q_{1,i} \\ \vdots & \vdots & \vdots & \vdots \\ q_{n-1,2} & \cdots & q_{n-1,n} & \sum_{i=1}^n a_i q_{n-1,i} \\ q_{n,2} & \cdots & q_{n,n} & \sum_{i=1}^n a_i q_{n,i} \end{bmatrix}$$
$$= \begin{bmatrix} q_{2,2} & \cdots & q_{2,n} & \sum_{i=1}^n a_i q_{2,i} \\ \vdots & \vdots & \vdots & \vdots \\ q_{n,2} & \cdots & q_{n,n} & \sum_{i=1}^n a_i q_{n,i} \\ \sum_{i=1}^n a_i q_{i,2} & \cdots & \sum_{i=1}^n a_i q_{i,n} & \sum_{i=1}^n \sum_{j=1}^n a_i a_j q_{i,j} \end{bmatrix}. \tag{C.7}$$

From (C.7), we observe that, for $i = 1, \ldots, n$,

$$q_{i,i} = \begin{cases} q_{i+1,i+1}, & i < n, \\ 1 + a_1 \left(a_1 q_{1,1} + \cdots + a_n q_{1,n}\right) + \cdots + a_n \left(a_1 q_{n-1} + \cdots + a_n q_{n,n}\right), & i = n. \end{cases}$$

Furthermore, for $1 \leq i < j \leq n$,

$$q_{i,j} = \begin{cases} q_{i+1,j+1}, & i < n, \\ a_1 q_{i+1,1} + \cdots + a_n q_{i+1,n}, & i = n. \end{cases}$$

Hence, Q is Toeplitz. □

APPENDIX D

Max Rank of Regressor Matrix Due to Nonzero Initial Conditions

Problem Formulation

Consider the time-series model

$$y(k) = -A_1 y(k-1) - \cdots - A_n y(k-n) + B_0 u(k) + \cdots + B_n u(k-n), \tag{D.1}$$

where $k = n, n+1, \ldots, l-1$, $l > n$, $y(k) \in \mathbb{R}^p$, $u(k) \in \mathbb{R}^m$, $A_1, \ldots, A_n \in \mathbb{R}^{p \times p}$, and $B_0, \ldots, B_n \in \mathbb{R}^{p \times m}$. We decompose $y(k)$ into its free and forced components, that is,

$$y(k) = y_{\text{free}}(k) + y_{\text{forced}}(k),$$

where for $k \geq n$, $y_{\text{free}}(k)$ and $y_{\text{forced}}(k)$ satisfy

$$y_{\text{free}}(k) = -A_1 y_{\text{free}}(k-1) - \cdots - A_n y_{\text{free}}(k-n)$$

and

$$y_{\text{forced}}(k) = -A_1 y_{\text{forced}}(k-1) - \cdots - A_n y_{\text{forced}}(k-n) + B_0 u(k) + \cdots + B_n u(k-n).$$

We define

$$Y_{\text{free}} \triangleq \left[\begin{array}{cccc} y_{\text{free}}(n) & y_{\text{free}}(n+1) & \cdots & y_{\text{free}}(l-1) \end{array}\right] \in \mathbb{R}^{p\times(l-n+1)}, \tag{D.2}$$

$$\Phi_{\text{free}} \triangleq \left[\begin{array}{c} \Phi_{y_{\text{free}}} \\ 0 \end{array}\right] \in \mathbb{R}^{[pn+m(n+1)]\times(l-n)}, \tag{D.3}$$

$$\Phi_{y_{\text{free}}} \triangleq \left[\begin{array}{cccc} y_{\text{free}}(n-1) & y_{\text{free}}(n) & \cdots & y_{\text{free}}(l-2) \\ y_{\text{free}}(n-2) & y_{\text{free}}(n-1) & \cdots & y_{\text{free}}(l-3) \\ \vdots & \vdots & \ddots & \vdots \\ y_{\text{free}}(0) & y_{\text{free}}(1) & \cdots & y_{\text{free}}(l-n-1) \end{array}\right] \in \mathbb{R}^{pn\times(l-n)},$$

and

$$\theta \triangleq \left[\begin{array}{cccccc} -A_1 & \cdots & -A_n & B_0 & \cdots & B_n \end{array}\right] \in \mathbb{R}^{p\times[pn+m(n+1)]}$$

so that

$$\theta\Phi_{\text{free}} = Y_{\text{free}}. \tag{D.4}$$

We also define

$$Y_{\text{forced}} \triangleq \left[\begin{array}{cccc} y_{\text{forced}}(n) & y_{\text{forced}}(n+1) & \cdots & y_{\text{forced}}(l-1) \end{array}\right]^{p\times(l-n+1)}, \tag{D.5}$$

$$\Phi_{\text{forced}} \triangleq \begin{bmatrix} \Phi_{y_{\text{forced}}} \\ \Phi_u \end{bmatrix} \in \mathbb{R}^{[pn+m(n+1)]\times(l-n)}, \tag{D.6}$$

$$\Phi_{y_{\text{forced}}} \triangleq \begin{bmatrix} y_{\text{forced}}(n-1) & y_{\text{forced}}(n) & \cdots & y_{\text{forced}}(l-2) \\ y_{\text{forced}}(n-2) & y_{\text{forced}}(n-1) & \cdots & y_{\text{forced}}(l-3) \\ \vdots & \vdots & \ddots & \vdots \\ y_{\text{forced}}(0) & y_{\text{forced}}(1) & \cdots & y_{\text{forced}}(l-n-1) \end{bmatrix} \in \mathbb{R}^{pn\times(l-n)},$$

and

$$\Phi_u \triangleq \begin{bmatrix} u(n) & u(n+1) & \cdots & u(l-1) \\ u(n-1) & u(n) & \cdots & u(l-2) \\ \vdots & \vdots & \ddots & \vdots \\ u(0) & u(1) & \cdots & u(l-n-1) \end{bmatrix} \in \mathbb{R}^{m(n+1)\times(l-n)}$$

so that

$$\theta\Phi_{\text{forced}} = Y_{\text{forced}}. \tag{D.7}$$

Adding (D.4) and (D.7) we obtain

$$\theta\Phi = Y, \tag{D.8}$$

where

$$\Phi \triangleq \begin{bmatrix} \Phi_y \\ \Phi_u \end{bmatrix} \triangleq \begin{bmatrix} \Phi_{y_{\text{free}}} + \Phi_{y_{\text{forced}}} \\ \Phi_u \end{bmatrix} \triangleq \Phi_{\text{free}} + \Phi_{\text{forced}} \in \mathbb{R}^{[pn+m(n+1)]\times(l-n)}, \tag{D.9}$$

$$\Phi_y \triangleq \begin{bmatrix} y(n-1) & y(n) & \cdots & y(l-2) \\ y(n-2) & y(n-1) & \cdots & y(l-3) \\ \vdots & \vdots & \ddots & \vdots \\ y(0) & y(1) & \cdots & y(l-n-1) \end{bmatrix} \in \mathbb{R}^{pn\times(l-n)},$$

and

$$Y \triangleq Y_{\text{free}} + Y_{\text{forced}} = \begin{bmatrix} y(n) & y(n+1) & \cdots & y(l-1) \end{bmatrix}. \tag{D.10}$$

State-Space Modeling

Consider a minimal state-space realization of the time series model (D.1) given by

$$x(k+1) = Ax(k) + Bu(k), \quad x(0) = x_0, \tag{D.11}$$

$$y(k) = Cx(k) + Du(k), \tag{D.12}$$

where $k = 0, 1, \ldots, l-1$, $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}, C \in \mathbb{R}^{p\times n}$, and $D \in \mathbb{R}^{p\times m}$. We note that $x(k)$ can be decomposed as

$$x(k) = x_{\text{free}}(k) + x_{\text{forced}}(k),$$

where $x_{\text{free}}(k)$ and $x_{\text{forced}}(k)$ satisfy

$$x_{\text{free}}(k+1) = Ax_{\text{free}}(k), \quad x_{\text{free}}(0) = x_0,$$

$$y_{\text{free}}(k) = Cx_{\text{free}}(k),$$

and

$$x_{\text{forced}}(k+1) = Ax_{\text{forced}}(k) + Bu(k), \quad x_{\text{forced}}(0) = 0,$$

$$y_{\text{forced}}(k) = Cx_{\text{forced}}(k) + Du(k).$$

Furthermore

$$y_{\text{free}}(k) = CA^k x_0,$$

and thus

$$\Phi_{y_{\text{free}}} = \begin{bmatrix} CA^{n-1}x_0 & CA^n x_0 & \cdots & CA^{l-2}x_0 \\ CA^{n-2}x_0 & CA^{n-1}x_0 & \cdots & CA^{l-3}x_0 \\ \vdots & \vdots & \ddots & \vdots \\ Cx_0 & CAx_0 & \cdots & CA^{l-n-1}x_0 \end{bmatrix}. \tag{D.13}$$

We factor $\Phi_{y_{\text{free}}}$ as

$$\Phi_{y_{\text{free}}} = \bar{I}_{n,p}\mathcal{O}_n(A,C)\mathcal{K}_{l-n}(A,x_0), \tag{D.14}$$

where

$$\mathcal{O}_i(A,C) \triangleq \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{i-1} \end{bmatrix},$$

$$\mathcal{K}_j(A,x_0) \triangleq \begin{bmatrix} x_0 & Ax_0 & \cdots & A^{j-1}x_0 \end{bmatrix},$$

and

$$\bar{I}_{q,r} \triangleq \begin{bmatrix} 0 & \cdots & 0 & I_r \\ \vdots & \iddots & \iddots & 0 \\ 0 & \iddots & \iddots & \vdots \\ I_r & 0 & \iddots & 0 \end{bmatrix} \in \mathbb{R}^{qr\times qr}. \tag{D.15}$$

We factor $\Phi_{y_{\rm forced}}$ as

$$\Phi_{y_{\rm forced}} = \bar{I}_{n,p}\bar{H}\bar{U}, \tag{D.16}$$

where

$$\bar{H} \triangleq \begin{bmatrix} \mathcal{H}_0 & \mathcal{H}_1 & \cdots & \mathcal{H}_{l-n-1} \\ \mathcal{H}_1 & \mathcal{H}_2 & \cdots & \mathcal{H}_{l-n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{H}_{n-1} & \mathcal{H}_n & \cdots & \mathcal{H}_{l-2} \end{bmatrix} \in \mathbb{R}^{pn\times m(l-1)(l-n)},$$

$$\bar{U} \triangleq \begin{bmatrix} U & 0_{m(l-1)\times 1} & \cdots & 0_{m(l-1)\times 1} \\ 0_{m(l-1)\times 1} & U & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0_{m(l-1)\times 1} & \cdots & 0_{m(l-1)\times 1} & U \end{bmatrix} \in \mathbb{R}^{m(l-1)(l-n)\times(l-n)},$$

$$U \triangleq \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(l-2) \end{bmatrix} \in \mathbb{R}^{p\times m(l-1)},$$

and for $i = 0, 1, \ldots, l-2$,

$$\begin{aligned} \mathcal{H}_i &\triangleq \begin{bmatrix} CA^{i-1}B & CA^{i-2}B & \cdots & CB & D & 0_{p\times m(l-i-2)} \end{bmatrix} \\ &= \begin{bmatrix} C\mathcal{K}_i(A,B)\bar{I}_{n,p} & D & 0_{p\times m(l-i-2)} \end{bmatrix} \in \mathbb{R}^{p\times m(l-1)}. \end{aligned} \tag{D.17}$$

Conditioning of Φ

Proposition D.0.8

$$\operatorname{rank}(\Phi_{\text{free}}) = \operatorname{rank}(\Phi_{y_{\text{free}}}) = \operatorname{rank}(\mathcal{K}_{l-n}(A, x_0)).$$

Proof D.0.10 Since $\bar{I}_{n,p}$ is nonsingular, $\operatorname{rank}(\mathcal{O}_n(A,C)) = n$, and $\mathcal{K}_{l-n}(A,x_0)$ has n rows, from (D.3) and (D.14) it follows from Sylvester's inequality that

$$\begin{aligned} \operatorname{rank}(\mathcal{K}_{l-n}(A,x_0)) &\geq \operatorname{rank}(\Phi_{\text{free}}) \\ &= \operatorname{rank}(\Phi_{y_{\text{free}}}) \\ &= \operatorname{rank}(\bar{I}_{n,p}\mathcal{O}_n(A,C)\mathcal{K}_{l-n}(A,x_0)) \\ &= \operatorname{rank}(\mathcal{O}_n(A,C)\mathcal{K}_{l-n}(A,x_0)) \\ &\geq \operatorname{rank}(\mathcal{O}_n(A,C)) + \operatorname{rank}(\mathcal{K}_{l-n}(A,x_0)) - n \\ &= \operatorname{rank}(\mathcal{K}_{l-n}(A,x_0)). \end{aligned}$$ □

Let $\sigma_1(\Phi) \geq \sigma_2(\Phi) \geq \cdots$ denote the singular values of Φ, and let $\rho(A)$ denote the spectral radius of A. We now examine the singular values $\sigma_i(\Phi)$ as the contribution of y_{free} to Φ increases.

Proposition D.0.9 Consider (D.1) with input $\{u(k)\}_{k=0}^{l-1}$ and a sequence $\{x_{0i}\}_{i=1}^{\infty}$ of initial conditions, where $\|x_{0i}\| \to \infty$ as $i \to \infty$ and, for all i, $r = \operatorname{rank}(\mathcal{K}_{l-n}(A, x_{0i}))$. Then

$$\frac{\sigma_{r+1}(\Phi_i)}{\sigma_r(\Phi_i)} \to 0 \quad \text{as} \quad i \to \infty,$$

where

$$\Phi_i \triangleq \alpha_i \Phi_{\text{free},i} + \Phi_{\text{forced}},$$

$$\Phi_{\text{free},i} \triangleq \begin{bmatrix} \bar{I}_{n,p}\mathcal{O}_n(A,C)\mathcal{K}_{l-n}(A,\hat{x}_{0i}) \\ 0 \end{bmatrix},$$

$$\alpha_i \triangleq \|x_{0i}\|,$$

and

$$\hat{x}_{0i} \triangleq \frac{x_{0i}}{\|x_{0i}\|}.$$

Proof D.0.11 Using Proposition D.0.8, it follows that $\operatorname{rank}(\Phi_{\text{free},i}) = r$. Since

$\sigma_i(\cdot)$ is continuous and $\alpha_i \to \infty$ as $i \to \infty$, we have

$$\begin{aligned}
\lim_{i\to\infty} \frac{\sigma_{r+1}(\Phi_i)}{\sigma_r(\Phi_i)} &= \lim_{i\to\infty} \frac{\sigma_{r+1}(\alpha_i \Phi_{\text{free},i} + \Phi_{\text{forced}})}{\sigma_r(\alpha_i \Phi_{\text{free},i} + \Phi_{\text{forced}})} \\
&= \lim_{i\to\infty} \frac{\sigma_{r+1}(\Phi_{\text{free},i} + \frac{1}{\alpha_i}\Phi_{\text{forced}})}{\sigma_r(\Phi_{\text{free},i} + \frac{1}{\alpha_i}\Phi_{\text{forced}})} \\
&= \frac{\sigma_{r+1}(\Phi_{\text{free},i})}{\sigma_r(\Phi_{\text{free},i})} \\
&= 0. \qquad \square
\end{aligned}$$

Proposition D.0.9 shows that the condition number of Φ increases as the contribution of the free response to Φ increases relative to the contribution of the input $u(k)$.

Proposition D.0.10 Let $M \in \mathbb{R}^{n\times n}$ be nonsingular and assume MAM^{-1} is a Jordan form matrix. Then (A, x_0) is controllable if and only if A is cyclic and, for every eigenvalue λ of A, the component of Mx_0 corresponding to the lowest-right entry of the Jordan block associated with λ is nonzero.

Proof D.0.12 See [119], pp. 209-214. □

Proposition D.0.11 Let $U \neq 0$ and let $l > 2n$. Then

$$\operatorname{rank}(\bar{H}) + (l-n)(m - ml + 1) \leq \operatorname{rank}(\Phi_{y_{\text{forced}}}) \leq \operatorname{rank}(\bar{H}).$$

Proof D.0.13 Since $\bar{I}_{n,p}$ is nonsingular, it follows from (D.16) and Sylvester's inequality that

$$\begin{aligned}\operatorname{rank}(\bar{H}) &\geq \operatorname{rank}(\Phi_{y_{\rm forced}})\\ &= \operatorname{rank}(\bar{I}_{n,p}\bar{H}\bar{U})\\ &= \operatorname{rank}(\bar{H}\bar{U})\\ &\geq \operatorname{rank}(\bar{H}) + \operatorname{rank}(\bar{U}) - m(l-1)(l-n)\\ &= \operatorname{rank}(\bar{H}) + (l-n) - m(l-1)(l-n). \qquad \Box\end{aligned}$$

Stochastic Analysis

Lemma D.0.3 Let $\beta \in \mathbb{R}$, assume that A is asymptotically stable and (A, B) is controllable, and let Q_β satisfy the discrete-time Lyapunov equation

$$Q_\beta = \beta^2 A^{\rm T} Q_\beta A + BB^{\rm T}. \tag{D.18}$$

Then $\operatorname{tr} Q_\beta \to \infty$ as $\beta \uparrow \frac{1}{\rho(A)}$.

Proof D.0.14 Write Q_β as

$$Q_\beta = \sum_{i=0}^{\infty} \beta^{2i} A^i BB^{\rm T} A^{i{\rm T}}.$$

Let $\lambda \in \mathbb{C}$ be an eigenvalue of A such that $|\lambda| = \rho(A)$, and let $v \in \mathbb{C}^n$ be an eigenvector of $A^{\rm T}$ associated with λ such that $v^*v = 1$. Since (A, B) is controllable, it follows that rank $\left(\begin{bmatrix} \lambda I - A & B \end{bmatrix}\right) = n$. Therefore,

$$\begin{aligned} 0 \neq & \begin{bmatrix} \lambda I - A^{\mathrm{T}} \\ B^{\mathrm{T}} \end{bmatrix} v \\ = & \begin{bmatrix} \lambda v - \lambda v \\ B^{\mathrm{T}} v \end{bmatrix} \\ = & \begin{bmatrix} 0 \\ B^{\mathrm{T}} v \end{bmatrix}. \end{aligned}$$

Hence $B^{\mathrm{T}} v \neq 0$. Therefore

$$\operatorname{tr} Q_\beta \geq v^* Q_\beta v = v^* B B^{\mathrm{T}} v \sum_{i=0}^{\infty} |\beta\lambda|^{2i} \to \infty \quad \text{as} \quad \beta \uparrow \frac{1}{\rho(A)}. \qquad \square$$

Numerical Examples

Example D.0.1 To illustrate Proposition D.0.9 for $n = r = 2$, let

$$A = \begin{bmatrix} -0.1 & 0.2 \\ 0 & 0.5 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \tag{D.19}$$

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad D = 0. \tag{D.20}$$

We choose $l = 100$ and initial condition

$$x_{0i} = \alpha_i \begin{bmatrix} 0.96 \\ -0.27 \end{bmatrix}. \tag{D.21}$$

For each α_i, we choose 10 Gaussian white noise input sequences $u(k)$ with zero mean and standard deviation $\sigma_u = 1$. Then we simulate (D.11)–(D.12) for each $u(k)$ and each α_i using the minimal realization (D.19)–(D.20) with the initial condition in (D.21). Next, we construct Φ and compute $\frac{\sigma_3(\Phi)}{\sigma_2(\Phi)}$ for each $u(k)$. Finally, we average $\frac{\sigma_3(\Phi)}{\sigma_2(\Phi)}$ for each α_i. Figure D.1 shows that $\frac{\sigma_3(\Phi)}{\sigma_2(\Phi)}$ decreases as α increases, which demonstrates Proposition D.0.9 for $n = r = 2$. To illustrate Proposition D.0.9 for $n = 2, r = 1$, we choose $l = 100$ and initial condition

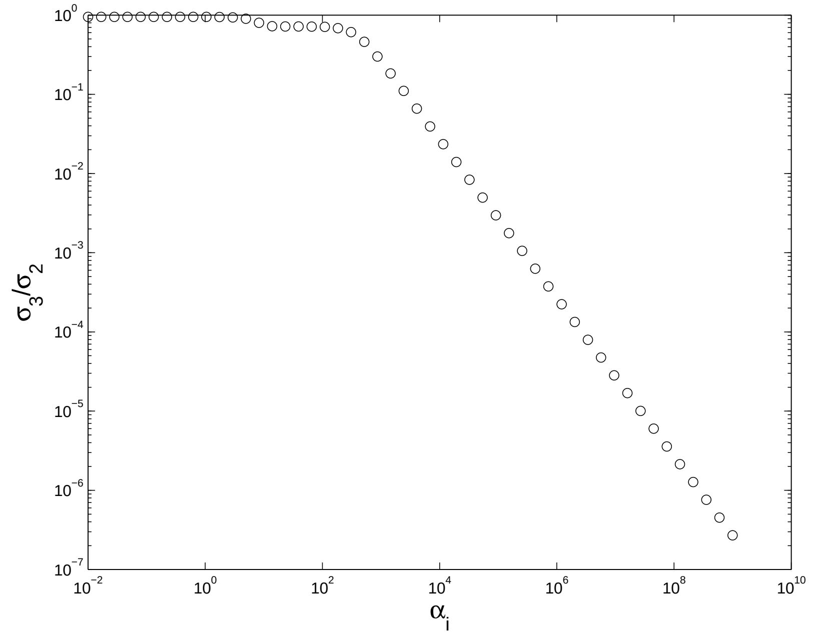


Figure D.1: Change in $\frac{\sigma_3(\Phi)}{\sigma_2(\Phi)}$ with α_i, averaged over 10 realizations of white noise $u(k)$ with $l = 100$.

$$x_{0i} = \alpha_i \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \tag{D.22}$$

For each α_i, we choose 10 Gaussian white noise input sequences $u(k)$ with zero mean and standard deviation $\sigma_u = 1$. Then we simulate (D.11)–(D.12) for each $u(k)$ and each α_i using the minimal realization (D.19)–(D.20) with the initial condition in

(D.22). Next, we construct Φ and compute $\frac{\sigma_2(\Phi)}{\sigma_1(\Phi)}$ for each $u(k)$. Finally, we average $\frac{\sigma_2(\Phi)}{\sigma_1(\Phi)}$ for each α_i. Figure D.2 shows that $\frac{\sigma_2(\Phi)}{\sigma_1(\Phi)}$ decreases as α_i increases, which demonstrates Proposition D.0.10 for $n = 2, r = 1$.

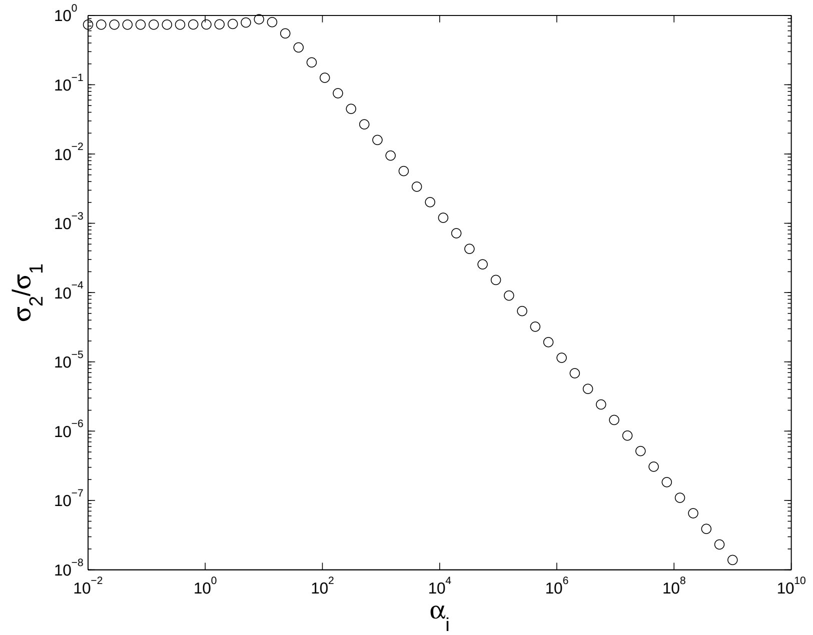


Figure D.2: Change in $\frac{\sigma_2(\Phi)}{\sigma_1(\Phi)}$ with α_i, averaged over 10 realizations of white noise $u(k)$ with $l = 100$.

Example D.0.2 To illustrate Lemma D.0.3 for $n = 2$, we choose $\beta \in [0, 2)$ and A and B as in (D.19) and compute $\operatorname{tr} Q_\beta$. To examine the behavior of the individual states $x(k)$ as β approaches 2, we choose initial condition $x_0 = \begin{bmatrix} 0 & 0 \end{bmatrix}^{\mathrm{T}}$. Next, we simulate

$$x(k+1) = \beta A x(k) + B u(k), \quad x(0) = x_0 \tag{D.23}$$

using a set of zero-mean, unity-variance Gaussian white noise input sequences $\{u(k)\}_{k=0}^{999}$. Then we define an "effective" initial condition $x^*(0) = x(1000)$ associated with each white noise input sequence. For each β and each $x_0 = x^*(0)$, we choose a zero-mean, unity-variance Gaussian white noise input sequence $\{u(k)\}_{k=0}^{l-1}$ and simulate (D.23). Then we compute an estimate of the size of the state (the trace of the steady-state covariance of $x(k)$) for each β, given by

$$\begin{aligned} \sigma_x^2 &\triangleq \frac{1}{l}\sum_{i=0}^{l-1} \left(x(i)-\bar{x}\right)^{\mathrm{T}} \left(x(i)-\bar{x}\right), \\ &\approx \operatorname{tr}\mathbb{E}[xx^{\mathrm{T}}], \end{aligned} \tag{D.24}$$

where

$$\bar{x} \triangleq \frac{1}{l}\sum_{i=0}^{l-1} x(i). \tag{D.25}$$

Repeating this process, we compute an average of σ_x^2, which we plot along with $\operatorname{tr} Q_\beta$ in Figure D.3. Figure D.3 shows that σ_x^2 converges to $\operatorname{tr} Q_\beta$ as l increases. Figure D.3 also shows that both σ_x^2 and $\operatorname{tr} Q_\beta$ approach ∞ as $\beta \uparrow 2$.

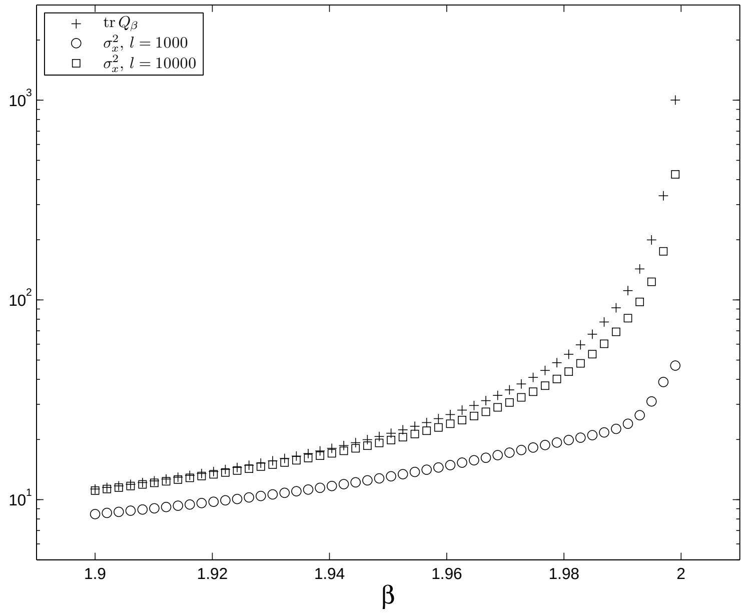


Figure D.3: Plot of σ_x^2 and Q_β versus β for $n = 2$, where σ_x^2 approaches ∞ as β approaches 2 and σ_x^2 converges to Q_β as l increases.

APPENDIX E

Zeros of the Discrete-Time Transfer Function from Excitation to Acceleration in a Mass-Spring-Damper System

Problem Formulation

Consider the mass-spring-damper system shown in Figure E.1. The equations of motion for this system are given by

$$\dot{\bar{x}}(t) = \bar{A}x(t) + \bar{B}u(t), \tag{E.1}$$

$$\bar{y}(t) = C\bar{x}(t) + D\bar{u}(t), \tag{E.2}$$

where $\bar{x}(t) \triangleq \begin{bmatrix} q(t) & \dot{q}(t) \end{bmatrix}^{\mathrm{T}}$ is the state of the system at time t, $\bar{u}(t)$ is the excitation at time t, and $\bar{y}(t)$ is the sensor measurement at time t. Assuming $\bar{u}(t)$ remains constant between samples, we can discretize (E.1) to obtain

$$x(k+1) = Ax(k) + Bu(k), \tag{E.3}$$

where $h > 0$ is the sampling interval, $x(k) \triangleq \bar{x}(kh)$, $u(k) \triangleq \bar{u}(kh)$, $A \triangleq e^{\bar{A}h}$, and

$$B \triangleq \int_0^h e^{\bar{A}\tau} d\tau \bar{B}.$$

Denoting $y(k) \triangleq \bar{y}(kh)$, (E.2) and (E.3) imply

$$y(k) = Cx(k) + Du(k). \tag{E.4}$$

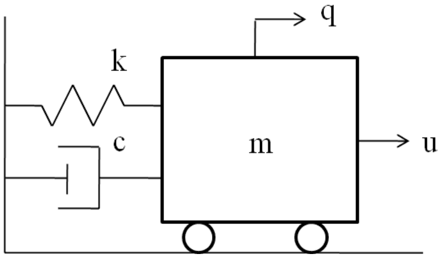


Figure E.1: Illustration of mass-spring-damper system.

Zeros of the Discrete-Time Transfer Function from Excitation to Acceleration: Double Integrator

Let

$$\bar{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} 0 \\ 1/m \end{bmatrix},$$

$C = \begin{bmatrix} 0 & 0 \end{bmatrix}$, and $D = 1/m$, where $m > 0$. Hence,

$$A = \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} \frac{h^2}{2m} \\ \frac{h}{m} \end{bmatrix}.$$

Then the zeros of the discrete-time transfer function from u to y are given by the roots of $C \operatorname{adj}(zI - A) B + D \det(zI - A)$. Hence, both zeros occur at $z = 1$.

Zeros of the Discrete-Time Transfer Function from Excitation to Acceleration: Damped Rigid Body

Let

$$\bar{A} = \begin{bmatrix} 0 & 1 \\ 0 & c \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} 0 \\ 1/m \end{bmatrix},$$

$C = \begin{bmatrix} 0 & c \end{bmatrix}$, and $D = 1/m$. Hence,

$$A = \begin{bmatrix} 1 & \frac{e^{ch}-1}{c} \\ 0 & e^{ch} \end{bmatrix}, \quad B = \begin{bmatrix} \frac{e^{ch}-ch-1}{mc^2} \\ \frac{e^{ch}-1}{mc} \end{bmatrix}.$$

Then the zeros of the discrete-time transfer function from u to y are given by the roots of $C \operatorname{adj}(zI - A) B + D \det(zI - A)$. Hence, both zeros occur at $z = 1$.

Zeros of the Discrete-Time Transfer Function from Excitation to Acceleration: Undamped Oscillator

Let

$$\bar{A} = \begin{bmatrix} 0 & 1 \\ -k & 0 \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} 0 \\ 1/m \end{bmatrix},$$

$C = \begin{bmatrix} -k & 0 \end{bmatrix}$, and $D = 1/m$, where $k > 0$. Hence,

$$A = \begin{bmatrix} \cos(\sqrt{k}h) & \frac{1}{k}\sin(\sqrt{k}h) \\ -sqrtk\sin(\sqrt{k}h) & \cos(\sqrt{k}h) \end{bmatrix}, \quad B = \begin{bmatrix} \frac{1-\cos(\sqrt{k}h)}{km} \\ \frac{\sin(\sqrt{k}h)}{\sqrt{k}m} \end{bmatrix}.$$

Then the zeros of the discrete-time transfer function from u to y are given by the roots of $C\,\mathrm{adj}\,(zI - A)\,B + D\det(zI - A)$. Hence, the zeros occur at

$$z = 1, \cos(\sqrt{k}h).$$

We note that both zeros do not occur at $z = 1$.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] R. Isermann, "Process fault detection based on modeling and estimation methods - a survey," *Automatica*, vol. 20, no. 4, pp. 387–404, 1984.

[2] J. J. Gertler, *Fault Detection and Diagnosis in Engineering Systems.* New York, NY: Marcel Dekker Inc., 1998.

[3] D. Balageas, C.-P. Fritzen, and A. Güemes, *Structural Health Monitoring.* London, UK: ISTE, 2006.

[4] B. Peeters and G. De Roeck, "Stochastic system identification for operational modal analysis: A review," *Journal of Dynamic Systems, Measurement, and Control*, vol. 123, no. 4, pp. 659–667, 2001.

[5] S. J. Qin, "Statistical process monitoring: basics and beyond," *Journal of Chemometrics*, vol. 17, pp. 480–502, 2003.

[6] K. F. Martin, "A review by discussion of condition monitoring and fault diagnosis in machine tools," *International Journal of Machine Tools and Manufacture*, vol. 34, no. 4, pp. 527–551, 1994.

[7] R. Kothamasu, S. H. Huang, and W. H. VerDuin, "System health monitoring and prognostics - a review of current paradigms and practices," *International Journal of Advanced Manufacturing Technology*, vol. 28, pp. 1012–1024, 2006.

[8] R. Yan and R. X. Gao, “Approximate entropy as a diagnostic tool for machine health monitoring,” *Mechanical Systems and Signal Processing*, vol. 21, no. 2, pp. 824–839, 2007.

[9] G.-L. Qian, S.-N. Gu, and J.-S. Jiang, “The dynamic behaviour and crack detection of a beam with a crack,” *Journal of Sound and Vibration*, vol. 138, no. 2, pp. 233–243, 1990.

[10] J.-T. Kim and N. Stubbs, “Crack detection in beam-type structures using frequency data,” *Journal of Sound and Vibration*, vol. 259, no. 1, pp. 145–160, 2003.

[11] O. Basir and X. Yuan, “Engine fault diagnosis based on multi-sensor information fusion using Dempster-Shafer evidence theory,” *Information Fusion*, vol. 8, no. 4, pp. 379–386, 2007.

[12] R. J. Patton and J. Chen, “Observer-based fault detection and isolation: robustness and applications,” *Control Engineering Practice*, vol. 5, no. 5, pp. 671–682, 1997.

[13] R. Isermann, “Model-based fault detection and diagnosis - status and applications,” *Annual Reviews in Control*, vol. 29, pp. 71–85, 2005.

[14] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, “A review of process fault detection and diagnosis part i: Quantitative model-based methods,” *Computers and Chemical Engineering*, vol. 27, pp. 293–311, 2003.

[15] J. S. Oakland, *Statistical Process Control*. Butterworth-Heinemann, 2003.

[16] A. D. Pouliezos and G. S. Stavrakakis, *Real-Time Fault Monitoring of Industrial Processes*. Kluwer Academic Publishers, 1994.

[17] M. Basseville and I. V. Nikiforov, *Detection of Abrupt Changes - Theory and Application*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1993.

[18] J. M. Lucas and M. S. Saccucci, "Exponentially weighted moving average control schemes: Properties and enhancements," *Technometrics*, vol. 32, no. 1, pp. 1–12, 1990.

[19] D. C. Montgomery, *Introduction to Statistical Quality Control*. John Wiley and Sons, 2001.

[20] A. K. S. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical Systems and Signal Processing*, vol. 20, no. 7, pp. 1483–1510, 2006.

[21] L. Wang and R. X. Gao, *Condition Monitoring and Control for Intelligent Manufacturing*. London, UK: Springer, 2006.

[22] I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh, *Feature Extraction: Foundations and Applications*. Springer-Verlag, 2006.

[23] J. J. Zakrajsek and D. G. Lewicki, "Detecting gear tooth fatigue cracks in advance of complete fracture," in *Technology Showcase on Integrated Monitoring, Diagnostics, and Failure Prevention*, (Mobile, AL), pp. 1–12, April 1996.

[24] R. X. Du, M. A. Elbestawi, and S. Li, "Tool condition monitoring in turning using fuzzy set theory," *International Journal of Machine Tools and Manufacture*, vol. 32, no. 6, pp. 781–796, 1992.

[25] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, no. 7–8, pp. 1157–1182, 2003.

[26] A. Bell and T. Sejnowski, "The 'independent components' of natural scenes are edge filters," *Vision Research*, vol. 37, no. 23, pp. 3327–3338, 1997.

[27] P. Pudil, J. Novovicova, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognition Letters*, vol. 15, no. 11, pp. 1119–1125, 1994.

[28] K. Fukunaga, *Introduction to Statistical Pattern Recognition.* Academic Press, 1991.

[29] C. Bishop, *Pattern Recognition and Machine Learning.* Springer, 2006.

[30] R. Isermann, "Supervision, fault-detection and fault-diagnosis methods - an introduction," *Control Engineering Practice*, vol. 5, no. 5, pp. 639–652, 1997.

[31] M. D. Spiridonakos, A. G. Poulimenos, and S. D. Fassois, "Output-only identification and dynamic analysis of time-varying mechanical structures under random excitation: a comparative assessment of parametric methods," *J. Sound & Vib.*, vol. 329, pp. 768–785, 2010.

[32] S. M. Pandit and S.-M. Wu, *Time Series and System Analysis with Applications.* Malabar, FL: R. E. Krieger Publishing Company, 2^{nd} ed., 1990.

[33] M. Basseville, A. Benveniste, M. Goursat, L. Hermans, L. Mevel, and H. Van der Auweraer, "Output-only subspace-based structural identification: from theory to industrial testing practice," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 123, pp. 668–676, 2001.

[34] L. Mevel, A. Benveniste, M. Basseville, M. Goursat, B. Peeters, H. Van der Auweraer, and A. Vecchio, "Input/output versus output only data processing for structural identification–application to in-flight data analysis," *J. Sound & Vib.*, vol. 295, pp. 531–552, 2006.

[35] A. Deraemaeker, E. Reynders, G. De Roeck, and J. Kullaa, "Vibration-based structural health monitoring using output-only measurements under changing environment," *Mech. Sys. & Sig. Proc.*, vol. 22, pp. 34–56, 2008.

[36] B. Peeters and G. De Roeck, “Reference-based stochastic subspace identification for output-only modal analysis,” *Mech. Sys. & Sig. Proc.*, vol. 13, pp. 855–878, 1999.

[37] F. Poncelet, G. Kerschen, J.-C. Golinval, and D. Verhelst, “Output-only modal analysis using blind source separation techniques,” *Mech. Sys. & Sig. Proc.*, vol. 21, pp. 2335–2358, 2007.

[38] E. Parloo, S. Vanlanduit, P. Guillaume, and P. Verboven, “Increased reliability of reference-based damage identification techniques by using output-only data,” *J. Sound & Vib.*, vol. 270, pp. 813–832, 2004.

[39] C. Rainieri and G. Fabbrocino, “Automated output-only dynamic identification of civil engineering structures,” *Mech. Sys. & Sig. Proc.*, vol. 24, no. 3, pp. 678–695, 2009.

[40] A. Agneni, L. B. Crema, and G. Coppotelli, “Output-only analysis of structures with closely spaced poles,” *Mech. Sys. & Sig. Proc.*, vol. 24, no. 5, pp. 1240–1249, 2010.

[41] C. Devriendt and P. Guillaume, “The use of transmissibility measurements in output-only modal analysis,” *Mechanical Systems and Signal Processing*, vol. 21, no. 7, pp. 2689–2696, 2007.

[42] C. Devriendt, G. De Sitter, and P. Guillaume, “An operational modal analysis approach based on parametrically identified multivariable transmissibilities,” *Mech. Sys. & Sig. Proc.*, vol. 24, no. 5, pp. 1250–1259, 2010.

[43] P. Mohanty and D. J. Rixen, “Operational modal analysis in the presence of harmonic excitation,” *Journal of Sound and Vibration*, vol. 270, pp. 93–109, 2002.

[44] M. Basseville, “Detecting changes in signals and systems - a survey,” *Automatica*, vol. 24, no. 3, pp. 309–326, 1988.

[45] L. Zhang, L. B. Jack, and A. K. Nandi, “Fault detection using genetic programming,” *Mechanical Systems and Signal Processing*, vol. 19, no. 2, pp. 271–289, 2005.

[46] R. Rengaswamy and V. Venkatasubramanian, “A fast training neural network and its updation for incipient fault detection and diagnosis,” *Computers and Chemical Engineering*, vol. 24, no. 2–7, pp. 431–437, 2000.

[47] R. J. Patton, P. M. Frank, and R. N. Clark, *Issues of Fault Diagnosis for Dynamic Systems*. London, UK: Springer-Verlag, 2000.

[48] V. S. Cherkassky and F. Mulier, *Learning from Data: Concepts, Theory, and Methods*. Hoboken, NJ: Wiley-IEEE, 2007.

[49] T. Söderström and P. Stoica, *System Identification*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[50] H. J. Palanthandalam-Madapusi, T. H. van Pelt, and D. S. Bernstein, “Parameter consistency and quadratically constrained errors-in-variables least-squares identification,” *Int. J. Contr.*, vol. 83, no. 4, pp. 862–877, 2010.

[51] J.-N. Juang, *Applied System Identification*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[52] L. Ljung, *System Identification: Theory for the User*. Upper Saddle River, NJ: Prentice-Hall, 2$^{\text{nd}}$ ed., 1999.

[53] A. J. Brzezinski, L. Li, X. Qiao, and J. Ni, “A new method for grinder dressing fault mitigation using real time peak detection,” *International Journal of Advanced Manufacturing Technology*, vol. 45, pp. 470–480, 2009.

[54] A. J. Brzezinski, Y. Wang, D. K. Choi, X. Qiao, and J. Ni, "Feature-based tool condition monitoring in a gear shaving application," in *ASME International Conference on Manufacturing Science and Engineering*, (Evanston, IL), pp. 65–74, October 2008.

[55] Y. Wang, A. J. Brzezinski, X. Qiao, and J. Ni, "Shaving tool wear classification using heuristic feature selection," *submitted to ASME Journal of Manufacturing Science and Engineering*.

[56] A. M. D'Amato, A. J. Brzezinski, M. S. Holzel, J. Ni, and D. S. Bernstein, "Sensor-only noncausal blind identification of pseudo transfer functions," in *Proc. SYSID*, (Saint-Malo, France), pp. 1698–1703, July 2009.

[57] A. J. Brzezinski, S. L. Kukreja, J. Ni, and D. S. Bernstein, "Sensor-only fault detection using pseudo transfer function identification," in *Proc. Amer. Contr. Conf.*, (Baltimore, MD), pp. 5433–5438, June 2010.

[58] A. J. Brzezinski, S. L. Kukreja, J. Ni, and D. S. Bernstein, "Pseudo transfer functions in sensor-only identification," *Automatica*, submitted.

[59] A. J. Brzezinski, S. L. Kukreja, D. S. Bernstein, and J. Ni, "Identification of sensor-only mimo pseudo transfer functions," in *submitted to Conference on Decision and Control*, (Orlando, FL), December 2011.

[60] E. A. Weaver and G. Richardson, "Threshold setting and the cycling of a decision threshold," *System Dynamics Review*, vol. 22, pp. 1–26, 2006.

[61] H. Yang, J. Mathew, and L. Ma, "Fault diagnosis of rolling element bearings using basis pursuit," *Mechanical Systems and Signal Processing*, vol. 19, pp. 341–356, 2005.

[62] S. Gu, J. Ni, and J. Yuan, “Non-stationary signal analysis and transient machining process condition monitoring,” *International Journal of Machine Tools and Manufacture*, vol. 42, pp. 41–51, 2002.

[63] H. R. Martin and F. Honarvar, “Application of statistical moments to bearning failure detection,” *Applied Acoustics*, vol. 44, pp. 66–77, 1995.

[64] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*. New Jersey: Prentice-Hall, 1999.

[65] S. Tomazic and S. Znidar, “A fast recursive stft algorithm,” in *Electrotechnical Conference*, (Bari, Italy), pp. 1025–1028, May 1996.

[66] L. Li, L. Qu, and X. Liao, “Haar wavelet for machine fault diagnosis,” *Mechanical Systems and Signal Processing*, vol. 21, pp. 1773–1786, 2007.

[67] M. J. Arnold, M. Roessgen, and B. Boashash, “Filtering real signals through frequency modulation and peak detection in the time-frequency plane,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, (Adelaide, SA, Australia), pp. 345–348, April 1994.

[68] H. Choi and W. J. Williams, “Improved time-frequency representation of multicomponent signals using exponential kernels,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, pp. 862–871, 1989.

[69] C. J. Li and J. D. Limmer, “Model-based condition index for tracking gear wear and fatigue damage,” *Wear*, vol. 241, no. 1, pp. 26–32, 2000.

[70] J. Gertler and D. Singer, “A new structural framework for parity equation based failure detection and isolation,” *Automatica*, vol. 26, no. 2, pp. 381–388, 1990.

[71] S. Alag, A. Agogino, and M. Morjaria, “A methodology for intelligent sensor measurement, validation, fusion, and fault detection for equipment monitoring

and diagnosis," *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing*, vol. 15, pp. 307–320, 2001.

[72] J. Jiang and Q. Zhao, "Comparison of parameter and state estimation based FDI algorithms," in *IFAC Conference On System Identification*, (Kitakyushu, Japan), pp. 655–660, 1997.

[73] J. D. Gibson and J. L. Melsa, *Introduction to Nonparametric Detection with Applications.* IEEE Press, 1996.

[74] R. Isermann, "Fault diagnosis of machines via parameter estimation and knowledge processing," *Automatica*, vol. 29, no. 4, pp. 815–835, 1993.

[75] L. Jiang, D. Djurdjanovic, and J. Ni, "A new method for sensor degradation detection, isolation, and compensation in linear systems," in *ASME International Mechanical Engineering Congress and Exposition*, (Seattle, WA), pp. 1089–1101, November 2007.

[76] W. A. Shewhart, *Economic Control of Quality of Manufactured Product.* D. Van Nostrand Company, 1931.

[77] W. E. Deming, "On probability as a basis for action," *The American Statistician*, vol. 29, no. 4, pp. 146–152, 1975.

[78] S. Malkin, *Grinding Technology: Theory and Applications of Machining with Abrasives.* Ellis Horwood Ltd., 1989.

[79] H. K. Tonshoff, T. Friemuth, and J. C. Becker, "Process monitoring in grinding," *CIRP Annals - Manufacturing Technology*, vol. 51, no. 2, pp. 551–571, 2002.

[80] R. Tetl and D. D'Addona, "Grinding wheel management through neuro-fuzzy

forecasting of dressing cycle time," *CIRP Annals - Manufacturing Technology*, vol. 52, pp. 407–410, 2003.

[81] J. F. Gomes de Oliveira and D. A. Dornfeld, "Application of ae contact sensing in reliable grinding monitoring," *CIRP Annals - Manufacturing Technology*, vol. 50, pp. 217–220, 2001.

[82] T. W. Liao, C. Ting, J. Qu, and P. J. Blau, "A wavelet-based methodology for grinding wheel condition monitoring," *International Journal of Machine Tools and Manufacture*, vol. 47, pp. 580–592, 2007.

[83] J.-S. Kwak and M.-K. Ha, "Detection of dressing time using the grinding force signal based on the discrete wavelet decomposition," *International Journal of Advanced Manufacturing Technology*, vol. 23, no. 1-2, pp. 87–92, 2004.

[84] I. Inasaki and K. Okamura, "Monitoring of dressing and grinding processes with acoustic emission signals," *CIRP Annals - Manufacturing Technology*, vol. 34, pp. 277–280, 1985.

[85] J. F. MacGregor and T. Kourti, "Statistical process control of multivariate processes," *Control Engineering Practice*, vol. 3, pp. 403–414, 1995.

[86] U. S. Cutting Tool Institute, *Metal Cutting Tool Handbook*. New York: Industrial Press, Inc., 7th ed., 1989.

[87] F. Klocke and T. Schröder, "Gear shaving: Simulation and technological studies," in *ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, (Chicago, IL), pp. 257–264, September 2003.

[88] C. Hung, J. Liu, S. Chang, and H. Lin, "Simulation of gear shaving with considerations of cutter assembly errors and machine setting parameters," *Interna-*

tional Journal of Advanced Manufacturing Technology, vol. 35, no. 3–4, pp. 400–407, 2007.

[89] M. Lv and X. Yang, "Design and manufacture of a shaving cutter with unequal depth gashes," *Journal of Materials Processing Technology*, vol. 129, no. 1–3, pp. 193–195, 2002.

[90] M. A. Elbestawi, T. A. Papazafifiou, and R. X. Du, "In-process monitoring of tool wear in milling using cutting force signature," *International Journal of Machine Tools and Manufacure*, vol. 31, no. 1, pp. 55–73, 1991.

[91] J. J. Zakrajsek and D. G. Lewicki, "Detecting gear tooth fatigue cracks in advance of complete fracture," *TriboTest*, vol. 4, no. 4, pp. 407–422, 1998.

[92] D. Yu, Y. Yang, and J. Cheng, "Application of time-frequency entropy method based on Hilbert-Huang transform to gear fault diagnosis," *Measurement*, vol. 40, pp. 823–830, 2007.

[93] Y. Wang, L. Li, J. Ni, and S. Huang, "Feature selection using tabu search with long-term memories and probabilistic neural networks," *Pattern Recognition Letters*, vol. 30, no. 7, pp. 661–670, 2009.

[94] W. Siedlecki and J. Sklansky, "A note on genetic algorithms for large-scale feature selection," *Pattern Recognition Letters*, vol. 10, no. 11, pp. 335–347, 1989.

[95] J. Rafiee, F. Arvani, A. Harifi, and M. H. Sadeghi, "Intelligent condition monitoring of a gearbox using artificial neural network," *Mechanical Systems and Signal Processing*, vol. 21, no. 4, pp. 1746–1754, 2007.

[96] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1–3, pp. 37–52, 1987.

[97] D. S. Bernstein, *Matrix Mathematics.* Princeton University Press, 2^{nd} ed., 2009.

[98] D. E. Dimla Sr. and P. M. Lister, “On-line metal cutting tool condition monitoring ii: tool-state classification using multi-layer perceptron neural networks,” *International Journal of Machine Tools and Manufacture*, vol. 40, no. 5, pp. 769–781, 2000.

[99] A. K. Jain, R. P. W. Duin, and J. Mao, “Statistical pattern recognition: A review,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4–37, 2000.

[100] M. A. Tahir, A. Bouridane, and F. Kurugollu, “Simultaneous feature selection and feature weighting using hybrid tabu search/K-nearest neighbor classifier,” *Pattern Recognition Letters*, vol. 28, no. 4, pp. 438–446, 2007.

[101] D. F. Specht, “Probabilistic neural networks,” *Neural Networks*, vol. 3, no. 1, pp. 109–118, 1990.

[102] P. D. Wasserman, *Advanced Methods in Neural Computing.* New York, NY: Van Nostrand Reinhold, 1993.

[103] K. Levenberg, “A method for the solution of certain non-linear problems in least squares,” *The Quarterly of Applied Mathematics*, vol. 2, pp. 164–168, 1944.

[104] G. Xu, H. Liu, L. Tong, and T. Kailath, “A least-squares approach to blind channel identification,” *IEEE Trans. Sig. Proc.*, vol. 43, no. 12, pp. 2982–2993, 1995.

[105] D. E. Adams, *Health Monitoring of Structural Materials and Components: Methods with Applications.* Chicester, UK: John Wiley & Sons, Ltd., 2007.

[106] W. E. Boyce and R. C. DiPrima, *Elementary Differential Equations and Boundary Value Problems.* Hoboken, NJ: Wiley Interscience, 1965.

[107] R. Kalman, B. L. Ho, and K. Narendra, "Controllability of linear dynamical systems," *Contributions to Differential Equations*, vol. 1, pp. 189–213, 1963.

[108] R. H. Middleton and G. C. Goodwin, *Digital Control and Estimation.* Prentice Hall, 1990.

[109] J. Argüero and G. C. Goodwin, "Identifiability of errors in variables dynamical systems," *Automatica*, vol. 44, pp. 371–382, 2008.

[110] K. J. Åström, P. Hagander, and J. Sternby, "Zeros of sampled systems," *Automatica*, vol. 20, no. 1, pp. 31–38, 1984.

[111] S. R. Weller, W. Moran, B. Ninness, and A. D. Pollington, "Sampling zeros and the Euler-Frobenius polynomials," *IEEE Trans. Automatic Contr.*, vol. 46, no. 2, pp. 340–343, 2001.

[112] I. Markovsky and S. Rao, "Palindromic polynomials, time-reversible systems, and conserved quantities," in *Proc. Mediterranean Conf. on Control. and Automation*, (Ajaccio, France), pp. 125–130, June 2008.

[113] M. Kamrunnahar, B. Huang, and D. G. Fisher, "Estimation of Markov parameters and time-delay/interactor matrix," *Chemical Engineering Science*, vol. 55, pp. 3353–3363, 2000.

[114] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control.* Prentice Hall, 1996.

[115] B. D. O. Anderson and J. B. Moore, *Optimal Filtering.* Englewood Cliffs, NJ: Prentice-Hall, 1979.

[116] T. Söderström and P. Stoica, "Instrumental variable methods for system identification," *Circuits Systems Signal Processing*, vol. 21, no. 1, pp. 1–9, 2002.

[117] T. H. van Pelt and D. S. Bernstein, "Least squares identification using μ-Markov parameterizations," in *Proc. Conf. Dec. Contr.*, (Tampa, FL), pp. 618–619, December 1998.

[118] P. Lancaster and M. Tismenetsky, *The Theory of Matrices: with applications.* San Diego, CA: Academic Press, 2^{nd} ed., 1985.

[119] C.-T. Chen, *Linear System Theory and Design.* New York, NY: Oxford University Press, 2^{nd} ed., 1984.