UNIVERSITY OF CALIFORNIA SAN DIEGO

Remote Health Monitoring of Gas Turbines: Mathematical Modeling and Numerical Optimization Techniques

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Engineering Sciences (Mechancial Engineering)

by

Cody W. Allen

Committee in charge:

Professor Mauricio de Oliveira, Chair
Professor Robert Bitmead, Co-Chair
Professor Phillip Gill
Professor William McNeaney
Professor Kal Seshadri

2020

The dissertation of Cody W. Allen is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____
Co-Chair

_____
Chair

University of California San Diego

2020

DEDICATION

To Julia my wife, without whom I would have starved, among other problems.

EPIGRAPH

One should not pursue goals that are easily achieved. One must develop an instinct for what one can just barely achieve through one's greatest efforts.

— Albert Einstein

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I thank my advisor, Dr. Mauricio de Oliveira. Without his guidance and encouragement, this dissertation would not be a reality. I would also like to thank Dr. Chad Holcomb for his constant encouragement and flexibility in my schedule, as well as for paving the way for obtaining an employee sponsored PhD. In addition, I thank Marco Leon and Solar Turbines for their continued support throughout my process. I would also like to thank my committee, for the constructive feedback they provided during my Senate exam. My first true mathematical mentor also deserves thanks, Dr. Vadim Ponomarenko. He guided and pushed me in my mathematical development before arriving at UCSD, and without whom, I never would have made it.

I also thank my parents for always encouraging me to pursue my dreams and to keep working, as well as birthing me. The inventor of brewing coffee should be thanked as well, even though I do not know who should be honored. Finally, I will thank Julia one more time, for everything.

Chapter 2, in full, is a reprint of the material as it appears in ASME Turbo Expo 2016. Allen, Cody; de Oliveira, Maurcio, ASME 2016. The dissertation author was the primary investigator and author on this paper.

Chapter 3, in full, is a reprint of the material as it appears in ASME Turbo Expo 2017. Allen, Cody; de Oliveira, Maurcio, Holcomb, Chad; ASME 2017. The dissertation author was the primary investigator and author on this paper.

Chapter 4, in full, has been written based on patent application number 15852411. The dissertation author was the primary investigator and author on this patent application.

Chapter 5, in full, is a reprint of the material as it appears in ASME Turbo Expo 2020. Allen, Cody; de Oliveira, Maurcio; Balaji, Puru; Holcomb, Chad; ASME 2020. The dissertation author was the primary investigator and author on this paper.

Chapter 6, in full, is a reprint of the material as it appears in Journal of Gas Turbines and Power. Allen, Cody; de Oliveira, Maurcio, ASME 2018. The dissertation author was the primary

investigator and author on this paper.

Chapter 7, in full, is a reprint of the material as it was submitted to Journal of Optimization Theory and Application. Allen, Cody; de Oliveira, Maurcio, Springer 2020. The dissertation author was the primary investigator and author on this paper.

VITA

2016-2020	Ph. D. in Engineering Sciences (Mechancial Engineering), University of California San Diego
2012-2015	M. S. in Applied Mathematics, San Diego State University, San Diego
2004-2008	B. S. in Business Finance, Northern Arizona University, Flagstaff

PUBLICATIONS

C.W. Allen, M. de Oliveira. A Minimal Cardinality Solution to Fitting Sawtooth Piecewise Linear Functions. *Journal of Optimization Theory and Applications*; Submitted 2020

C.W. Allen, P. Balaji, M. de Oliveira, C.M. Holcomb. Axial Compressor Fouling and its Effect on Gas Turbine Fuel Consumption and Emissions. *ASME Turbo Conference*; Accepted to *ASME Turbo Conference* 2020, London (UK).

C.W. Allen, C. Holcomb, M. de Oliveira. Estimating Recoverable Performance Degradation Rates and Optimizing Maintenance Scheduling. *Journal of Engineering for Gas Turbines and Power*, Vol 141(1), 2018

C.W. Allen, M. de Oliveira, C.M. Holcomb. Gas Turbine Machinery Diagnostics: A Brief Review and a Sample Application. *ASME Turbo Conference* 2017, Charlotte NC (USA).

C.W. Allen, M. de Oliveira, C.M. Holcomb. Fault Detection Using Reduced Rank Linear Engine Models. *ASME Turbo Conference* 2016, Seoul NC (S.Korea).

C.W. Allen, M. de Oliveira. System and Method for Estimation of Gas Fuel Lower Heating Value using Energy Balances and Parametric Error Modeling. *Application Number: 15852411; US Patent App. 15/852,411*, 2019. Patent pending.

ABSTRACT OF THE DISSERTATION

Remote Health Monitoring of Gas Turbines: Mathematical Modeling and Numerical Optimization Techniques

by

Cody W. Allen

Doctor of Philosophy in Engineering Sciences (Mechancial Engineering)

University of California San Diego, 2020

Professor Mauricio de Oliveira, Chair
Professor Robert Bitmead, Co-Chair

This dissertation is a repository of techniques employed in the health management of industrial gas turbines. The first chapters cover methods of diagnostics using statistics and machine learning algorithms to identify deviations from normal operation. The middle chapters cover applications of prognostics in the sense of generating predictions of various sensor values of interest and comparing predictions with observed values through residual analysis. The later chapters develop applications of optimization techniques to the estimation of compressor degradation rates and to the creation of optimized maintenance schedules. The final chapter takes

a more scrutinizing look at the mathematics of estimating piecewise linear functions and proves a result about minimal cardinality. Taken in whole, this work covers a broad spectrum from applications to theory of remote health management of Gas Turbines, and generally, rotating machines.

Chapter 1

Introduction

There is growing demand in the turbomachinery industry to make more data driven decisions. The days of old, where accepting the idea stemming from the loudest voice in the room, are over.

In 2018 and 2019, *Data Science* was considered by many the "sexiest" job in the United States, and businesses and educational institutions were struggling to keep up with the emerging field. In 2020, much of this is still true; however, educational institutions such as UCSD have built data science centers and now offer master's degrees in the subject. Businesses now have teams of data scientists at the ready to massage data. For context, I am currently employed by Solar Turbines, and I am assumed to be a data scientist by many in the company although my exact title is Research Engineer. Whatever the case, I have had first hand experience at a large company developing and deploying machine learning models to solve real world problems. Throughout this text, I will insert concise opinions that did not appear in the original published texts with the goal of giving perspective gained over the past five years both from the commercial and academic lenses of view. It is my hope that this will provide the reader with interesting ideas to think about as well as stand as a time capsule of what role data science and machine learning play in

application to connected devices at the close of the decade.

Currently, a plethora of research is being done on gas turbine (GT) monitoring as evidenced by the large increase in total papers published in the past five years at the ASME Turbomachinery Exposition Conferences. This dissertation aims to add to that growing body of research. Primarily, the goal has been to discover methods for formulating, tracking and solving current problems in the industry. In the Introduction, a concise background discussion about the topics researched in the main body of this text is provided.

1.1 Gas Turbine Monitoring and Fault Detection

1.1.1 Gas Turbine Condition Monitoring

GT performance prediction has been broadly classified into two approaches [SBW08], *model-based* and *model-free*. The model-based and model-free approaches each have their merit. Model-free approaches can lead to interesting conclusions, some inconsistent with the physical reality of a GT system. I first hand experience with output of model-free approaches, both acceptable and non-acceptable. Chapter 2 shall take a model-based approach, where expert knowledge of the GTs being modeled is an input during model creation.

In either case, monitoring and diagnostic activities help owners of GTs to identify deviations from normal operation, thus allowing operators to act before major malfunctions occur [VP12]. The newest paragon of architecture for such monitoring and analysis in the industrial turbine setting is an incorporation of an on-site and a remote based monitoring system [SR15]. The on-site logic continuously monitors control sensors and actuator positions. As part of the architecture, automated data acquisition and transmission systems on-site work to transfer the gathered data to the remote monitoring locations. From these remote monitoring stations, the sampled analog engine measurement data can be further analyzed by algorithms too intricate or memory intensive for on-site feasibility as well as by system experts or for post

processing [DSR+13][MKGC91][KNG92]. As is true with any modeling endeavor, the key to the performance of the technique is having a model that accurately reflects the nominal operating performance of the actual engine [SR15].

1.1.2 Diagnostics and Machine Learning for Condition Monitoring

Engine diagnostic monitoring and health management for GTs has been implemented, starting in the mid-1970s, in various forms. Algorithms and implementation have evolved significantly as both engines and computational capabilities have become more efficient, complex and powerful. Much of the early development of such methods was born out of the aerospace industry's engine's safety requirements[Vol14]; however, recent years have seen significant contributions from industrial applications. Advances in technology have enabled a proliferation of smaller, cheaper and more accurate sensors that are capable of generating a wealth of data for the analysis of the condition and overall health of an engine. This increase in engine data has lent itself to analysis by algorithms, ever-increasing in sophistication.

The definition of "Equipment Health Management" (EHM) has many subtle variations one can find when surveying the field of definitions; however, these can usually be distilled into three primary components [Vol14]:

1. monitoring
2. diagnostics
3. recommendations

The first component, *monitoring*, is the act of periodically observing the machine state while keeping a record of the observations. Monitoring can be done at the machine or remotely, depending on the machine setup. The current industry standard is to have the capability of remote monitoring, in which sensor values are captured at a given sampling frequency and then

bundled, sent to a remote source and stored[JRS+16]. Typically, monitoring is done with intent to determine if the currently observed machine state is at a nominal state or close to nominal state. This introduces the second component of engine health management, *diagnostics*. Diagnostics, according to Merriam-Webster [MW16], can be defined generally as "investigation or analysis of the cause or nature of a condition, situation or problem." Thus, by definition, diagnostics can only take place after a problem or condition has been observed.

Observation of a state that departs from the specified nominal state requires the observer to determine or classify what "departure" actually entails. This is a critical difference in the data science and machine learning vernacular. The distinction is that of *unsupervised* learning compared with *supervised* learning. In supervised learning, one has both the inputs and the outputs whereas, in unsupervised learning, one only has the inputs. These problems are fundamentally different. Here is an analogy: consider a man who weighs 225 pounds, is six feet tall and has a resting heart rate of 60 beats per minute (BpM) and a woman who weighs 115 pounds, is five and a half feet tall, and has a resting heart rate of 37 BpM. The question to answer is, are these people healthy? Let's further say you have a sample of 50 people, with measurements of height, weight and resting BpM. In the unsupervised case, there does not exist a per person label of healthy or unhealthy. Statistical methods can be employed to find deviations, but a deviation is not necessarily unhealthy. To definitively decide, labels are necessary. Once a label is available, the problem moves from the unsupervised class to the supervised class, and a host of algorithms can be employed to construct prediction functions. The point is that crossing the threshold from unclassified data to classified data is non-trivial and may be subjective. Generally, an expert must be employed to help classify enough data that supervised learning algorithms can become effective. The last decade saw the Internet of Things (IoT) flourish, and smart (i.e., wirelessly connected) devices have grown exponentially. Indeed, this is core technology for any remote monitoring, EHM strategy. However, data from analog sensors in time are unclassified/unlabeled. The problem of turning unclassified data into classified data will be of large interest in coming

years as applications of machine learning attempt to penetrate further into the IoT space.

The ability to classify unclassified data, at least in part, depends on the sophistication of the monitoring system, as well as the experience of the expert performing the investigation[VCZ+04]. Determining a label for the data is equivalent to providing a *result/recommendation* of (or based on) the diagnosis of the equipment's departure from a nominal operating state. That is, the diagnosis *is* the label of healthy or unhealthy. This final piece is when value is truly added from the standpoint of the equipment owner [GE15]. The equipment owner, armed with the result and/or recommendation, is then able to make educated decisions on what action to take next, as it relates to the afflicted equipment, in order to satisfy business requirements.

With the above definition of EHM in place, it is easy to see how various algorithms can be used to aid in this process. In the 2016 ASME Turbo Expo, a new, vogue term called the "digital twin" emerged in keynote addresses. This term refers to the application of algorithms which simulates the processes of the GT system to a high degree of accuracy. In fact, there are many applications of generalized modelling and simulation of GTs [W.R83][HA13][M.T09]. Delving further into the analysis of the GT as it relates to EHM, one finds more direct applications of algorithms to specific sections of the GT for degradation detection. Indeed, neither degradation and anomaly detection nor gas path analysis are new topics for algorithmic based analysis[HLBC16][R.K01]. There are numerous mathematical tools that can be employed with computers to aid in the analyzing of the monitored machinery, many of which can be found in Lee, Wu et all [LWZ+13]. In much of the literature on EHM, application of such algorithms are used directly to ascertain the state of the machine[LWZ+13][SBW08][MKGC91][IR16]. However, a critical step that is often overlooked or under emphasized in application of these methods is understanding the source of the measurements and how changes to this source can change the measurements and ultimately, the conclusions drawn. For example, many authors state that fuel flow is a critical parameter in gas path analysis [R.K01][HA13]. When looking at typical fuel flow time series data, it is easy to fall prey to the idea that the measurement has a great deal of

noise in it. However, this idea can be misleading. As will be seen later in chapter 3, different fuel valves have subtly different flow characteristics, which can lead an engineer to make incorrect conclusions about degradation when looking at data from a fleet of engines. Care must be taken when applying algorithms to fleets of data, and the investigator should be aware of physical differences in machines and what effects these differences produce in data. Indeed, machine learning practitioners are still bound by the laws of physics in the IoT space, an often foreign concept to fledgling practitioners.

1.2 Hybrid Prediction Models and Applications

Hybrid models are models that combine both physics and data driven methods. One instantiation of the idea is to constrain a model form to only take on values that are possible in reality. Piecewise models are excellent examples of this. In fact, many neural networks use this idea repeatedly, a reason they often perform well when given enough data to learn from.

Further applications fall into the realm of mathematical modeling, where the investigator comes up with some form of a mathematical model to fit experimental data. Often the model will have coefficients that are found by fitting the model to the data. In our view, this is another form of a hybrid model. With this idea of a hybrid model, then hybrid models and so-called *first order* physics models can be quite similar but remain different. First order physics models are models that are derived from known physical laws. However, often there are still parameters that must be estimated based on a given physical setup. An excellent example of this is the well known Plank equation for Blackbody radiation. As the story goes, Plank had some data and one afternoon after a lunch party, sat down with pencil and paper and developed an equation that modeled the data. Years later, a physics based derivation showed that the model form was well founded in physical law [Pic08].

1.2.1 Hybrid Models for Predicting Gas Lower Heating Values

In the GT gas path (the pathway that air and fuel are mixed and combusted) fuel plays an important role in calculating the total usable energy the GT can produce. In particular, there are two properties that can be used to accurately tell how much energy is available and how much quantity of fuel is necessary to achieve that output, the *lower heating value* (LHV) and the *specific gravity* (SG) of the gas. The LHV is a measure of the calorific chemical energy available in a fixed amount of fuel, and the SG is a measure of the density of the fuel.

Heating values can be determined in a number of ways, with complexity and accuracy depending on the given method. The heating value of a fuel can be determined experimentally by employing an adiabatic bomb calorimeter, which measures the enthalpy change between reactants and products[SA05]. Prediction of heating values of lignocellulosics and carbonaceous materials from *proximate analysis* is another standard procedure[CMRMR01]. In addition to these methods, there exist correlation based methods that use tables and elemental analysis of fuels[SC02]. In Chapter 4 we employ a hybrid model to estimate the most likely LHV value, given some operational data from the GT. The model is a composite of a first principles model and an error function. The error function is a data driven, second order polynomial model that is fit to the difference of the observations and predicted values from the first principles approach. The final prediction is the sum of the error function and first principles model. As will be shown, this hybrid approach is superior to the first principles model by itself, due mainly to the fact that it offers many more degrees of freedom and can be fit to observations.

1.2.2 Hybrid Models for Estimating Upper Bounds on Fuel Flow and CO_2 Emissions in the Presence of Compressor Fouling

Axial compressor degradation in the industrial GT is well documented in the literature. Up to 70%-85% of GT usable power loss can be attributed to compressor fouling [MHFW89],

where compressor *fouling* is defined as the adherence of particles to airfoils and annulus surfaces. The fouling process is well documented; see [KB12] and [LS86] for example.

The main effects of axial compressor fouling reported in the literature include a reduction in air flowrate, reduction in pressure ratio and loss of efficiency [LS86, MPSV10, JPSM18]. Kurz and Brun note that for a given shaft speed of a fouled compressor, each subsequent stage will see a lower Mach number (due mainly to higher temperature). In a fouled compressor, the net effect of this is that stages will work at lower than design efficiencies, and in whole, overall efficiency will be reduced. C. B. Meyer-Homji et. al. [MHBS13] note that as compressor efficiency drops, the axial compressor consumes even more power for a given flow and pressure ratio, effectively lowering the GT's output while increasing the fuel flow.

The primary approach to modeling compressor fouling used extensively in the literature is the adaptive modeling technique. The procedure involves scaling nominal compressor map components to match observed engine performance data. For compressor fouling, this includes using multiplicative factors applied to the corrected mass flow of air, pressure ratio and the compressor efficiency [SMP89, ARDM12, HLD$^+$18]. The scaled values are then used to calculate changes in performance metrics, such as lost power and increased heat rate [KB12].

To regain lost performance due to fouling, operators must decide how and when to wash the compressor or ramp up the fuel demand. Compressor washing will be discussed further in Section 1.3.1. Ramping up power demand (and consequently, fuel demand) only works if the engine was not previously at full load, or if the operator is willing to risk over-firing the engine for a given period of time. In either case, the effect is an incremental increase in fuel flow and correspondingly additional emissions. The literature often characterizes these increases indirectly as part of an overall fuel calculation for a given fouled state. In many studies, the values range from 0%-3% [ARDM12, HLD$^+$18, ELITG15]

Chapter 5 details compressor fouling and simulation and asks what the cost of fouling is in terms of additional fuel and additional CO_2. The model developed is hybrid; it is derived from

first principles alone, however, has a multiplicative coefficient for use in fitting data.

1.3 Optimization Techniques Applied to Gas Turbines

1.3.1 Optimized Maintenance Scheduling

GTs have become an integral part of the energy sector both in the US and abroad. Indeed, according to the US Energy Information Administration, projections of the energy consumption of Natural Gas have risen considerably. The Administration notes, "Natural gas use increases more than other fuel sources in terms of quantity of energy consumed, led by demand from the industrial and electric power sectors" [EIA17]. It is clear from this statement that US natural gas fired power plant output is expected to grow in coming years, independent of US energy policy. With this expected growth in mind, pursuing the operation and maintenance of GTs and in particular, industrial GTs, is more important now than ever.

Many power plant components deteriorate over time, and the component may have fully or almost fully recoverable deterioration with some basic maintenance. For example, compressors experience many forms of deterioration such as increased tip clearances, changes in airfoil geometry and changes in airfoil surface quality [R.K01]. Brun and Kurz point out that the first two degradation mechanisms generally result in non-recoverable degradation but that the third, airfoil surface quality changes, can at least be partially recovered with washing of the compressor blades. Another set of examples are found by considering air or oil filters and associated losses as a result of fouling. In industrial GTs, air inlet filter blockage can build up causing a pressure drop at the inlet of the compressor, which in turn can drive performance loss [SCBS09]. Performance loss can manifest indirectly as well. Consider degraded lube oil filters due to clogging from impurities. This leads to pressure loss in the lube oil system, which leads to decreased oil flow rates across bearings, resulting in longer oil dwell times and increased oil exit temperatures [MHT83]. Many modern control systems have bearing oil exit temperature threshold values

that if breached, will cause the engine to shutdown. Often, reducing the load on the engine will lead to decreased drain temperatures that allow the engine to continue running but at reduced load, an indirect form of recoverable performance degradation. Finally, similar scenarios occur in fin tubed air heat exchangers, where debris buildup can lead to decreased airflow through the heat exchanger and reduced heat rejection [WHTT17], causing lube oil temperatures to rise throughout the engine, with the same effect as oil filter degradation.

These examples illustrate various mechanisms of component degradation that is often fully or at least partially recoverable by performing a maintenance session. Further, after selection of an appropriate performance metric, many of these scenarios have linear or quasi-linear rates of degradation which can be modeled using a discontinuous *sawtooth* function. The sawtooth name comes from the constant decline with sporadic jumps up due to maintenance being performed. This function will be defined in Chapter 6 and revisited in Chapter 7. The bottom plot in Figure 1.3 shows a typical sawtooth signal.

The working example in this dissertation will focus on recoverable fouling of axial compressors due to particle deposition (as discussed in Section 1.2.2), and assumes that the performance degradation is fully recoverable upon performing a maintenance session. So called *hot section* degradation is not considered. We note that the time scales of nonrecoverable degradation are usually much longer than those of recoverable degradation. Indeed, I have seen field data in which time scales can be up to a factor of five or greater. Therefore, in the optimization procedure, the relative effect of nonrecoverable degradation can be assumed constant (i.e., in a given period where optimization shall be applied which is limited by the length of time in which accurate forecasts are available). Thus, for a given optimization period, by making the assumption that at the beginning of the period, the GT has no non-recoverable degradation, we are merely biasing the degradation by a constant offset, which has no effect on the optimization procedure or calculated result in regards to optimal number of maintenance sessions. Lastly, the method developed in Chapter 6 can be applied to any of the aforementioned degradation

mechanisms as long as a performance related instrumentation is available and a performance metric can be defined.

Compressor degradation is well documented in the literature, and in more recent years, the emergence of economic optimization utilizing compressor washing has been observed. Frequent on-line washing has been proposed as a cost effective method for keeping compressor blades clean, and is a nice alternative to off-line washing. However, many experimental evaluations have shown that such a cleaning regime does not return the same level of performance as off-line washing [BGFK15]. This has primarily been attributed to the fact that on-line washes are only effective in the first few stages [Boy12]. The degradation rate has been shown to depend heavily on the amount of contaminants entering the compressor through inlet air filters, and to a lesser extent, on the load at which the turbine is operating. The rate of degradation is not entirely unique, but is most often in reference to the calculated isentropic (or polytropic) efficiency of the compressor [BG07]. The effect of performance degradation can equivalently be seen by observing increases in heat rate for a fixed power output, ambient conditions, and fuel properties. Yet another way to observe the degradation rate is by forming the ratio of measured compressor discharge pressure to the theoretical discharge pressure given by a high fidelity engine simulation model or a well trained machine learning model, set with the same initial and boundary conditions. The general form of the degradation rate can be seen in Fig. 1.1, where Aretakis et al. give an illustration of the different forms of degradation. We observe that the degradation rate with and without on-line washing may be approximated over a wash interval as locally linear. In Fig. 1.2, Boyce et al. show isentropic efficiency degradation calculations for a set of compressors running side by side, in which one received no washing over a six month period of time (lower data) and the other received regular washing (upper data). Figures 1.1 and 1.2 show that degradation rates can be approximated with linear functions, and if load and inlet filtration do not vary significantly, the degradation rate can be fixed in time. We will use this fixed degradation rate as a basis for the optimization of maintenance session scheduling in the method developed in Chapter 6.

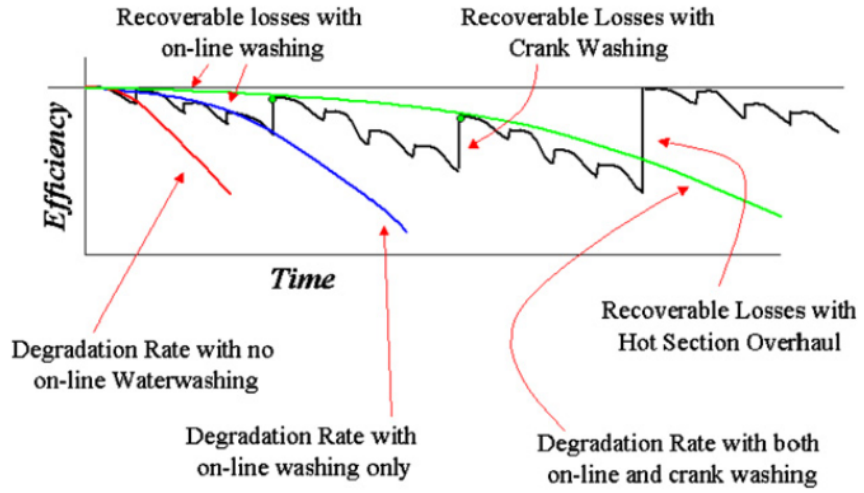


Figure 1.1: Image Credit: Aretakis et al. [ARDM12]. Depicts different degradation rates of compressor isentropic efficiency with various cleaning methods applied.

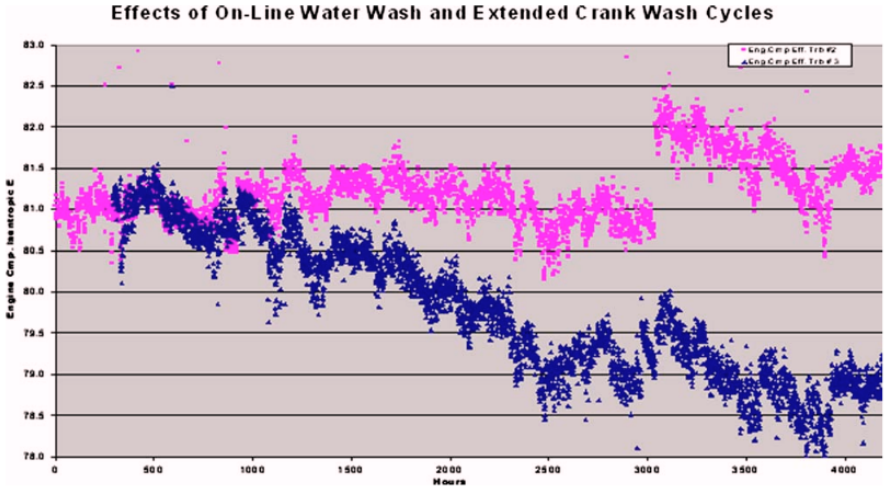


Figure 1.2: Image Credit: Boyce [BG07]. Depicts degradation of compressor isentropic efficiency for two engines (side by side at site) over 6-month period where no compressor washing occurs (bottom curve) and where regular on-line washing occurred (top curve). The rate of degradation when no washing occurs can be approximated by a line, giving a linear rate of degradation.

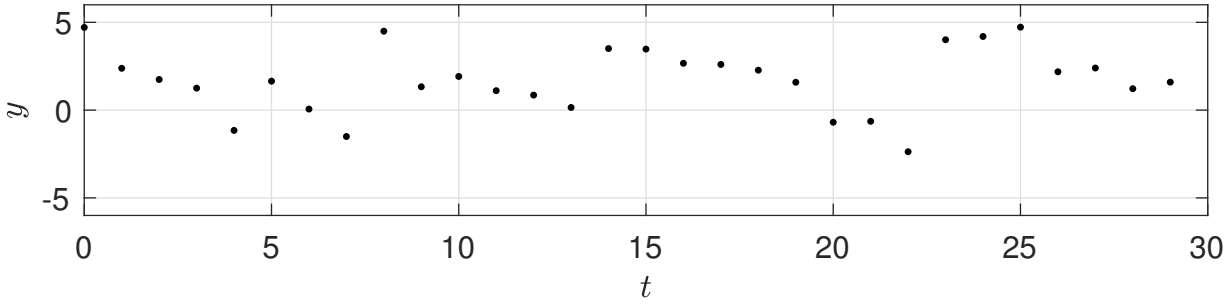
1.3.2 The Mathematics Behind Minimal Cardinality Piecewise Linear Models with Applications to Degradation Rate Modeling

Numerous signals in applied science and systems analysis take the form of a so called *sawtooth* signal. This is a signal that is piecewise-linear (PWL), where each segment has the same negative slope, and which yields a "sawtooth" shape when plotted in time. At a given discontinuity, the difference between one point of the function or another can take on any positive real value. The first and second plots in Figure 1.3 shows a typical example of noisy data generated by a system which produces a sawtooth signal in time. The data was generated by adding noise to a underlying sawtooth function with a given set of discontinuities. In the next section, we give various examples of physical system that can produce such sawtooth signals.

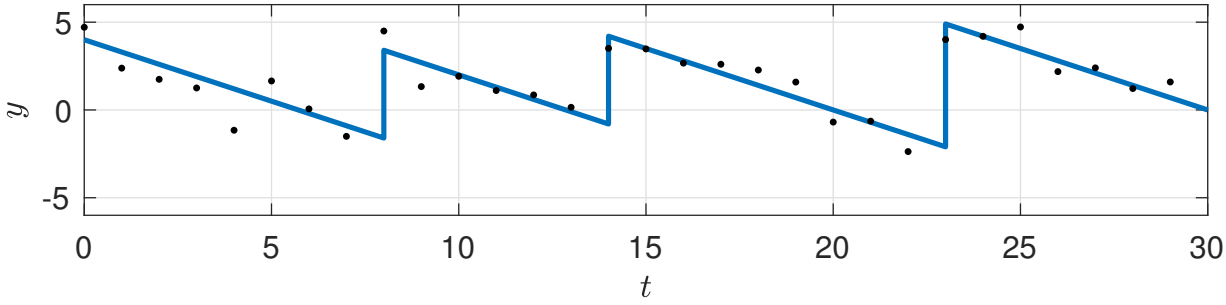
1.3.3 Physical Systems Producing Sawtooth Signals

In monitoring a gas turbine power plant's output, output power can degrade from nominal levels due to compressor fouling. Accretion of particle deposits on compressor blades disrupt the aerodynamic properties of the airfoils and lead to less power output [AHdO19]. This reduces the plant's ability to generate enough power to meet electrical load requirements and increases running costs due to lost efficiency [TMBK14]. Tracking of gas turbine efficiency produces shallow quadratic curves in time with ostensibly random positive jumps. These jumps occur when maintenance is performed and results in the return of efficiency. In industry, these curves are approximated by PWL functions with constant slope which yield sawtooth signals in time. In many cases, a daily degradation rate is sought, and when maintenance records are not available, maintenance interventions must be estimated from historical data.

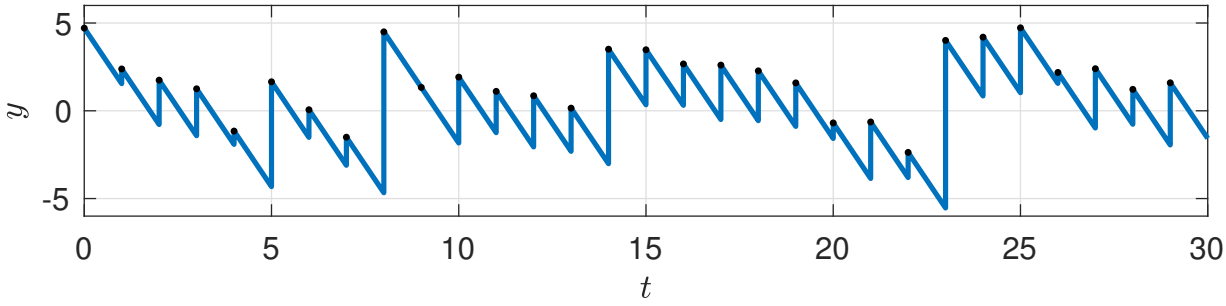
In CO_2 capture technology, there exist performance parameters that indicate capturing effectiveness. Amine plants treat the flue gas streams coming from combined heat and power plants using solvent based CO_2 capture technology. One way to detect solvent degradation is



(a) Observed sawtooth noisy samples



(b) Noisy samples and the true underlying function that generated the observations



(c) An error free fittingh solution.

Figure 1.3: Sawtooth function and samples.

achieved by monitoring the solvent viscosity, which over time exhibits increases as the solvent becomes less potent [FFdC$^+$17]. The increase in solvent viscosity follows linear growth until the solvent is replaced and a jump occurs in the monitored signal as it returns to baseline. Again, a rate of degradation is often sought from historical data.

Diverse examples of signals that can be approximated by sawtooth signals can be found in the literature and include applications such as monitoring marine vessels efficiency[Log11] or photovoltaic cell efficiency [JKVN16]. Sawtooth signals appear in various engineering planning activites, such as pavement maintenance planning [TS94]. Another application is to signals of constant magnitudes that can non-continuously jump to new magnitudes, such as the signals found in communication signal processing.

1.3.4 Existing Piecewise-Linear Function Approximation Methods

The general method of using PWL functions to approximate more complex functions is a well studied problem. Recent contributions have used mixed-integer linear programs with PWL functions to approximate multidimensional functions such as the Environmental Protection Agency's complex emissions model [MF10]. The application of convex programming has been investigated for special cases of PWL fitting. A general unconstrained version is described by solving the least squares problem [BV04]. Additionally, progress has been made by applying a least squares partition algorithm which is based on the so-called *Max-affine* function. This solution of the general PWL problem results in application of an iterative PWL fitting algorithm. The algorithm developed is heuristic but shown to be reliable when many iterations are performed with varied initial points [SPB08]. Magnani and Boyd show that as the number of sub-domains increase, then the *Root Mean Squared Error* (RMSE) tends to 0. This method has been applied to finding minimal PWL representations for quadratic cost functions by applying a constraint on the number of possible segments of the PWL function [AMM13]. Toriello et. al. found that a number of related works concentrate on the modeling of the given piecewise linear function and

of subsequent incorporation of the function into a mixed-integer program [TV12]. Further work has developed bounds on deviations of the PWL function from the nonlinear multivariate function [RK15]. In each of the applications of PWL fitting methods above, noiseless data is assumed. That is, the measured data is assumed to perfectly represent the true underlying function, and the question is how to find an optimal representation of the underlying function by a PWL function.

In the case of noisy data, the problem of fitting PWL functions to data generated by a nonlinear function increases in difficulty. There is considerably less information in the literature of the application of PWL function fitting with noisy data. Ingle et. al. develop a slope estimation algorithm for PWL data that is corrupted by Gaussian noise. The method employs the use of stochastic hidden Markov model to find breakpoints and Bayesian maximum a posteriori approach to estimate a discrete set of slopes, which are then utilized in a dynamic programming algorithm for posterior density maximization[IBSV15]. Another recently proposed method is that of fitting *jump models* to allow for temporally ordered data with noise. This method alternates between two steps, estimating parameters of multiple models and estimating the temporal sequence of model activation until convergence [BBPB18]. One theoretical shortcoming of fitting PWL functions to noisy data is the lack of guarantees of total model complexity; that is, a bound on the number of piecewise components emitted by a fitting method and a predefined tuning parameter. This complexity problem will be of primary concern in Chapter 7, where a proof will be given for the developed solution, showing that the solution is of minimal cardinality and hence model complexity.

We close the Introduction with a thought from Albert Einstein, my hero. “Try not to become a man of success, but try rather to become a man of value”. In performing research for this dissertation, I have attempted to create a valuable work for colleagues and future researchers. Without further ado, let us proceed to the research.

Chapter 2

Fault Detection Using Reduced Rank Linear Engine Models

2.1 Introduction

In this chapter, we will develop a rank reduced model that yields much more flexibility in input and output parameters. Such a model is superior to high-fidelity physics based models in applications such as near real-time monitoring. In addition, high-fidelity models cannot run when certain required input measurements such as ambient temperature or fuel energy content are not available, rendering the high-fidelity model useless. The motivation comes from the discussion in Sections 1.1.1 and 1.1.2.

We will first create a training set of data using a high-fidelity, physics based turbine model iterated over the expected operating regime with expected nominal values for the required input variables. Then we will implement a PCA analysis yielding the principle components and singular values that are proportional to the variances of the principle components [WAL03] [DHS01]. From this, we produce a reduced rank model using Total Least Squares that maps static behavior

of a turbine operating in closed loop. We note that this procedure is not quite principle component regression. In principle component regression, one fits the principle components to the response, or dependent variable [JWHT13]. In this chapter, we make no assertion of a dependent variable and instead minimize the orthogonal distances from the subset of selected measurements to a plane. In doing so, we do not introduce bias into the model. Selection of measurement variables are driven by sensors typically found in the field. We then formulate a minimal variance predictor that operates on measurement data and that is capable of incorporating both uncertainties and domain knowledge of measurement noise and accuracy, to enable refinement of the model. This allows the user to introduce bias into specific model inputs based on existing knowledge of sensors or specific operating conditions. Next, a comparison of predictions of the operational parameters to the actual measurement data is done. A calculation of how close the field engine is operating to the reference model is introduced by implementing the Mahalanobis distance. The Mahalanobis distance is a χ^2 statistic, and enables the user to quantitatively assess how close an engine under investigation is operating to the best model prediction. We perform an analysis of the residues to infer the health of the engine and show that the errors observed in modeling are within 3% variance and zero mean and approximately follow a Gaussian distribution. Finally, a discussion around how Binomial or non-Gaussian residual distributions indicate degradation and can be used to select candidates for further investigation of faults.

2.2 Chapter Notation

n	Number of variables
k	Time [sample number]
N	Number of sample data points
$\boldsymbol{\varphi}$	Individual observation vector
$\boldsymbol{z}$	Vector of reduced variables

Φ	Matrix of observations after standardization
TLS	Total Least Squares
SVD	Singular Value Decomposition
$(A, \boldsymbol{b})$	Model parameters
$\boldsymbol{y}$	Individual measurement vector
$\boldsymbol{z}$	Reduced parameter vector
$\hat{Z}$	Reduced rank prediction model
ρ^2	Mahalanobis distance statistic
C	User Defined input matrix
$\sum_{\eta}$	User defined measurement error
$\boldsymbol{\zeta}$	Model errors

2.3 Total Least Squares Modeling Algorithm

We seek to build a model of n variables, or parameters. The variables can be combinations of input and output data generated from a reference model of high fidelity or actual measurement machine data from an engine in the field. For example, common parameters into any model of a gas turbine include ambient temperature and pressure, compressor discharge pressure, downstream combustor temperature, and fuel properties. To begin with, we will take input from a high fidelity performance model. Let $n, k \in \mathbb{N}$ where n is the number of variables used in the model and k is a given time instance, then we have

$$\boldsymbol{\varphi}(k)^T = \begin{pmatrix} \varphi_1(k) & \varphi_2(k) & \cdots & \varphi_n(k) \end{pmatrix} \in \mathbb{R}^{n \times 1} \tag{2.1}$$

With N data points, an aggregate data matrix Φ is constructed as

$$\Phi = \begin{bmatrix} \boldsymbol{\varphi}(1) & \boldsymbol{\varphi}(2) & \cdots & \boldsymbol{\varphi}(N) \end{bmatrix} \in \mathbb{R}^{n \times N} \tag{2.2}$$

To mitigate numerical errors, we normalize or standardize the set of data per variable so that Φ is a set of normalized or standardized vectors of data.

2.3.1 Creating TLS Models From Data

We now turn our attention to TLS modeling of the supplied data. The modeling can be represented equivalently in two forms, one explicit involving additional variables, $\boldsymbol{z}$, and one implicit in which linear relationships are sought involving only the original variables, $\boldsymbol{\varphi}$. These models take the form

$$\boldsymbol{\varphi} = A\boldsymbol{z} + \boldsymbol{b} + \boldsymbol{\zeta} \tag{2.3}$$

$$B^T \boldsymbol{\varphi} = \boldsymbol{c} + \boldsymbol{\varepsilon} \tag{2.4}$$

where $\boldsymbol{\zeta}$ and $\boldsymbol{\varepsilon}$ represent zero mean perturbations or noise terms.

The rationale behind such model is that the data supplied are not fully independent and hence, there exists a reduced coordinate system such that the data can be represented. That is, the measurements relating to the gas turbine performance are limited in their degrees of freedom. Note that models (2.3) and (2.4) do not assume an *a priori* division of the data into inputs and output, but rather seek only to find relationships between the observed variables. Such an idea is closely related to the one in *Principal Component Analysis*. Indeed, the determination of the above model parameters to be described in the next section can be thought of as a specialized version of principal component analysis. In this chapter we will focus primarily on the application of this technique to gas turbines and the algorithms for implementation of these models.

Explicit Algorithm

With $\boldsymbol{\varphi}$ assembled in (2.2), the $(A, \boldsymbol{b})$ parameters define the model and $\boldsymbol{\zeta}$ describes the modeling error. We estimate the parameters $(A, \boldsymbol{b})$ from data as follows. First, define

$$\hat{\boldsymbol{b}} := \frac{1}{N} \sum_{k=1}^{N} \boldsymbol{\varphi}(k), \qquad \hat{\boldsymbol{b}} \in \mathbb{R}^{n \times 1}, \tag{2.5}$$

as the extended expectation (empirical mean). In order to estimate A we subtract $\hat{\boldsymbol{b}}$ from the data by defining $\bar{\Phi}$ as

$$\bar{\Phi} = \Phi - \hat{\boldsymbol{b}} = \begin{bmatrix} \boldsymbol{\varphi}(1) - \hat{\boldsymbol{b}} & \boldsymbol{\varphi}(2) - \hat{\boldsymbol{b}} & \cdots & \boldsymbol{\varphi}(N) - \hat{\boldsymbol{b}} \end{bmatrix} \in \mathbb{R}^{n \times N} \tag{2.6}$$

Now compute the singular value decomposition (SVD) of $\bar{\Phi}$

$$\bar{\Phi} = U \Sigma V^T, \qquad UU^T = I, \qquad VV^T = I, \tag{2.7}$$

with $\Sigma \geq 0$ diagonal. In order to distinguish between signals and noise partition U and Σ as follows:

$$U = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \tag{2.8}$$

where the first block of U, i.e. U_1, and the matrix Σ_1 have r columns where $r < n$ and are chosen as the columns associated with the first r singular values such that $\sigma_1, \sigma_2, ..., \sigma_r \gg \sigma_{r+1} \sigma_n \geq 0$. It is based on this partitioning that we estimate

$$\hat{A} := U_1, \qquad \hat{A} \in \mathbb{R}^{n \times r} \tag{2.9}$$

The remaining $m = n - r$ columns of U_2 and the associated matrix Σ_2 characterize the model

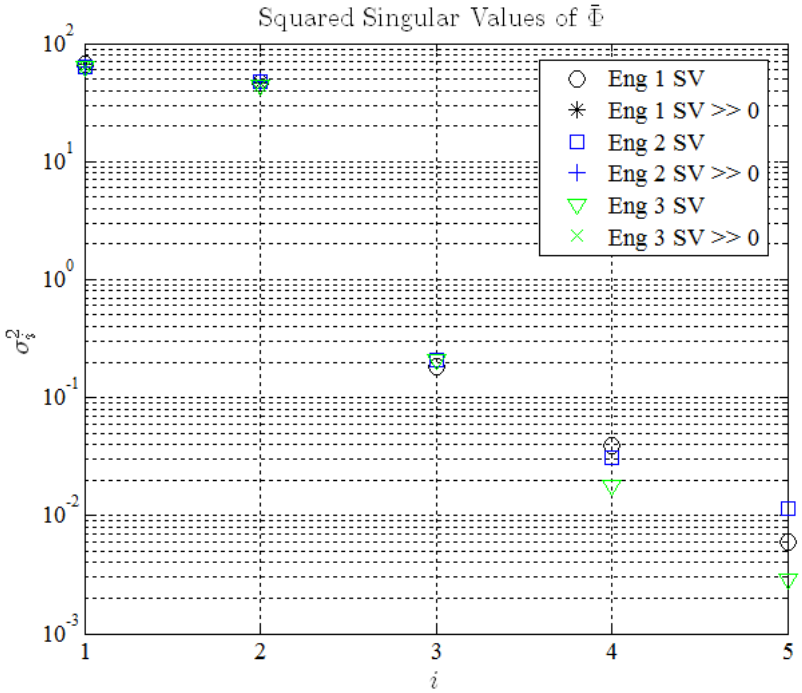


Figure 2.1: Singular Values of Covariance Matrix plotted on logarithmic scale. Three different turbine engine-types are shown. In each case, we have a rank two matrix.

errors. Indeed, it is possible to show that

$$\hat{\Sigma}_\varphi = \hat{A}\hat{\Sigma}_z\hat{A}^T + \hat{\Sigma}_\zeta, \qquad \hat{\Sigma}_z = \frac{1}{N-1}\Sigma_1^2, \qquad \hat{\Sigma}_\zeta = \frac{1}{N-1}U_2\Sigma_2^2U_2^T.$$

where $\hat{\Sigma}_x$ denotes the sample covariance matrix of the signal x, and that because $\sigma_r \gg \sigma_{r+1}$ we have that

$$\rho_\zeta^2 = \text{trace}(\hat{\Sigma}_\zeta) = \frac{1}{N-1}\sum_{k=r+1}^{n}\sigma_k^2 \tag{2.10}$$

is small.

Indeed, Figure 2.1 shows examples of the singular values obtained in the modeling of three different turbine engine-types. Note the logarithmic scale on the y-axis. Therefore, of the five selected operating parameters, only two are necessary to describe most of the variation of

the system, as indicated by the two non-zero singular values. However, this is specific to our selection of operating parameters and the operating ranges (for example, power output range) that data were collected over. The rank of the model can change based on the selection of operating parameters and operating ranges to be input into the model. Additionally, we would like to point out that in this model, the separation of singular values is easily identified, but that may not always be true. Indeed, we were delighted to see a linear model fit our industrial gas turbine data so well. More often, especially in analyzing aircraft engines, one linear model may not be appropriate for the entire operating range. It is common practice for aircraft engines to use a family of linear models and controllers for take-off, ascent, cruise, etc [KT04].

In the following sections we no longer distinguish between the model parameters $(A, \boldsymbol{b})$ and their estimates $(\hat{A}, \hat{\boldsymbol{b}})$ as computed in this section.

2.3.2 Calculating Predictions Based on Measurements

Once the model is fit to the reference data set, we then seek to make predictions by applying the trained model to either validation data or measurements from equipment in operation. With the model parameters $(A, \boldsymbol{b})$ calculated using some reference model data set, assume that a set of measurement data, $\boldsymbol{y}(k)$, is available comprising the entire set or a subset of the variables in the model, $\boldsymbol{\varphi}$. Let

$$\boldsymbol{y}(k) = C\boldsymbol{\varphi}(k) + \boldsymbol{\eta}(k), \qquad \boldsymbol{y}(k) \in \mathbb{R}^{p\times 1} \tag{2.11}$$

be a p-vector of measurements at time instance k, with $\boldsymbol{\eta}$ representing noise (or uncertainty) on the measurement. For example the matrix C is the identity matrix when $\boldsymbol{y}(k)$ is a measurement of the entire set of variables $\boldsymbol{\varphi}(k)$. In the same vein, it can be adapted to represent incomplete or redundant measurements.

Similar to the model data matrix Φ in (2.2), define a set of N_y measurements collected in

Table 2.1: Model and Measurement Covariances

Model Covariances	Measurement Covariances
$\Sigma_z = \frac{1}{N-1}\Sigma_1^2$	$\Sigma_y = C\Sigma_\varphi C^T + \Sigma_\eta$
$\Sigma_\zeta = \frac{1}{N-1}(U_2\Sigma_2^2 U_2^T)$	$\Sigma_\eta =$ user defined
$\Sigma_\varphi = \frac{1}{N-1}\left(A\Sigma_z A^T + \Sigma_\zeta\right)$	

the matrix Y as

$$Y = \begin{bmatrix} \boldsymbol{y}(1) & \boldsymbol{y}(2) & \cdots & \boldsymbol{y}(N_y) \end{bmatrix} \in \mathbb{R}^{p\times N_y} \tag{2.12}$$

To examine how the measurement data compares to the reference model we construct predictions for the reduced order coordinate $\hat{\boldsymbol{z}}$ and the full order data vector $\hat{\boldsymbol{\varphi}}$.

In the present section we consider the linear predictors written in the form

$$\hat{\boldsymbol{\varphi}}(k) = K_\varphi \boldsymbol{y}(k) + L_\varphi, \tag{2.13}$$

$$\hat{\boldsymbol{z}}(k) = K_z \boldsymbol{y}(k) + L_z \tag{2.14}$$

where (K_φ, L_φ) and (K_z, L_z) are parameters to be calculated to produce optimal prediction. Once $\hat{\boldsymbol{\varphi}}$ and $\hat{\boldsymbol{z}}$ are available, estimates of the errors $\boldsymbol{\varepsilon}(k)$ and $\boldsymbol{\zeta}(k)$ may also be calculated.

It is possible to show that the values of the gains in (2.13) and (2.14) that produce predictors are optimal in the least-squares sense[KSH00]. The gains are computed from the covariance matrices listed in Table 6.1 by the formulas

$$K_\varphi = \left(\Sigma_\varphi C^T\right)\Sigma_y^{-1} \qquad\qquad \varphi = \boldsymbol{b} - K_\varphi C\boldsymbol{b} \tag{2.15}$$

With (2.15), the prediction from (2.13) may be written in matrix form as

$$\hat{\Phi} = K_{\varphi} Y + L_{\varphi} \tag{2.16}$$

Similarly,

$$K_z = \left(\Sigma_z A^T C^T\right) \Sigma_y^{-1} \qquad\qquad L_z = -K_z C b \tag{2.17}$$

With (2.17), the reduced order prediction from (2.14) may be written in matrix form as

$$\hat{Z} = K_z Y + L_z \tag{2.18}$$

To evaluate the model fit, the modeling errors $\boldsymbol{\zeta}$ need to be examined. The modeling errors and descriptive statistics require calculation of the covariances of the measurements and predictions. We will investigate this later in the chapter.

2.4 Fitting Models to Reference Data Generated from High Fidelity Engine Models

In this section we will show and discuss our fits of our explicit models to our high-fidelity engine models to produce our reference models. In vogue language, this is the training phase of the machine learning process.

2.4.1 The Effectiveness of Our Models

It is well known that gas turbine high-fidelity models are nonlinear [Mat96]. We were surprised to find that linear models so accurately captured the operating behavior of our engines.

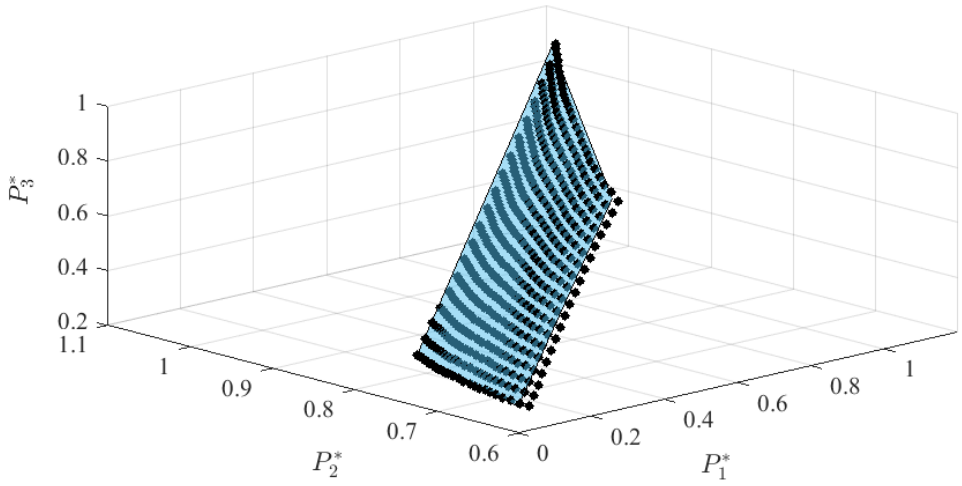


Figure 2.2: High Fidelity Turbine Model represented by black points. Reduced Rank Model Fit displayed as blue plane.

What we are seeing is that we lose the full degrees of freedom in operating parameters due to implementation of closed-loop control. Therefore, we were able to accurately model field based operating measurements subject to closed-loop control using linear models. In Figure 2.2, we see our model fit overlaid with the high-fidelity turbine model in three operating dimensions. As can be seen, while nonlinearity does exist, it is minimal across the selected operating parameters.

In Figure 2.3, we show three different turbine models, plotted in the same operating parameter space. The data come from the high-fidelity engine models and illustrate the variation in operating regimes in parameter space per turbine model. Note that the selected engine-types vary in power output from 5 MW to 14 MW.

Figure 2.4 shows the same three turbine engine-types plotted after applying a linear transformation and normalization to the raw measurements. Again, nonlinearities can be seen; however, again we see that this is minimal and thus, approximations with linear models may be applied. This shows that with the correctly defined linear transformation and normalization, we can represent three different turbine engine-types by the same underlying model with minimal loss in operating data resolution. We note that this was unexpected; here we see that three different

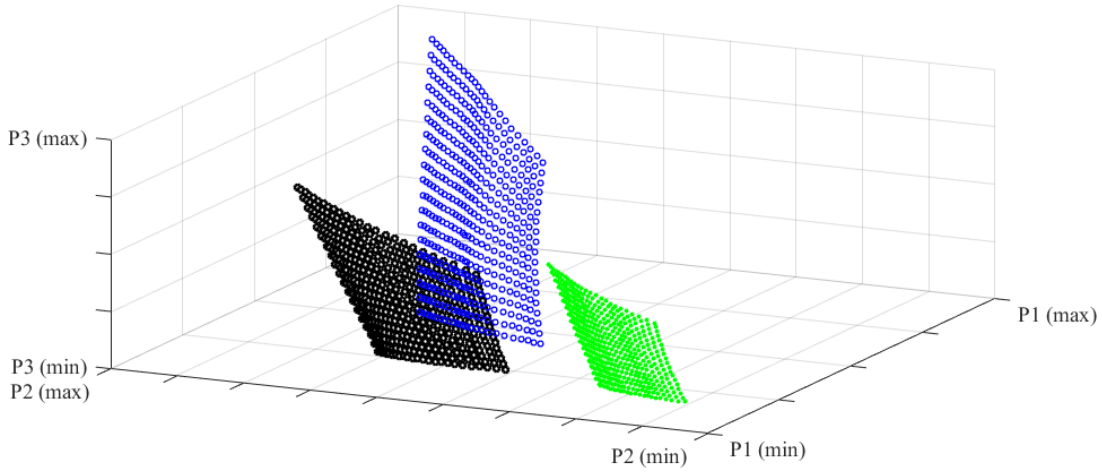


Figure 2.3: High-fidelity mathematical models for three different turbine engine-types in three operating parameters.

Table 2.2: Prediction Residuals from Model Fit

Parameter	Turbine 1	Turbine 2	Turbine 3
ζ_1	0.0011	0.0014	0.0011
ζ_2	0.0038	0.0031	0.0035
ζ_3	0.0029	0.0037	0.0028
ζ_4	0.0027	0.0031	0.0024
ζ_5	0.0029	0.0026	0.0024

engine-types are effectively the same engine when properly standardized and normalized.

In Table 2.2 we present a comparison of the individual turbine model fits. We fit models to each engine-type individually, using the same operating parameters and underlying model developed above. For a first check, we then make predictions by using our training data as the input to the model to produce predictions. From this, we produce residues by taking the 2-norm of each individual parameter and its prediction. We expect the residues to be very small due to the narrow band of operation our selected engines run in. Note that industrial turbines generally have more limited ranges of operations compared with aircraft engines. Note that in an industrial environment, a process like this can be mimicked in order to keep the number of underlying mathematical models small compared to the number of turbine configurations in the field. This

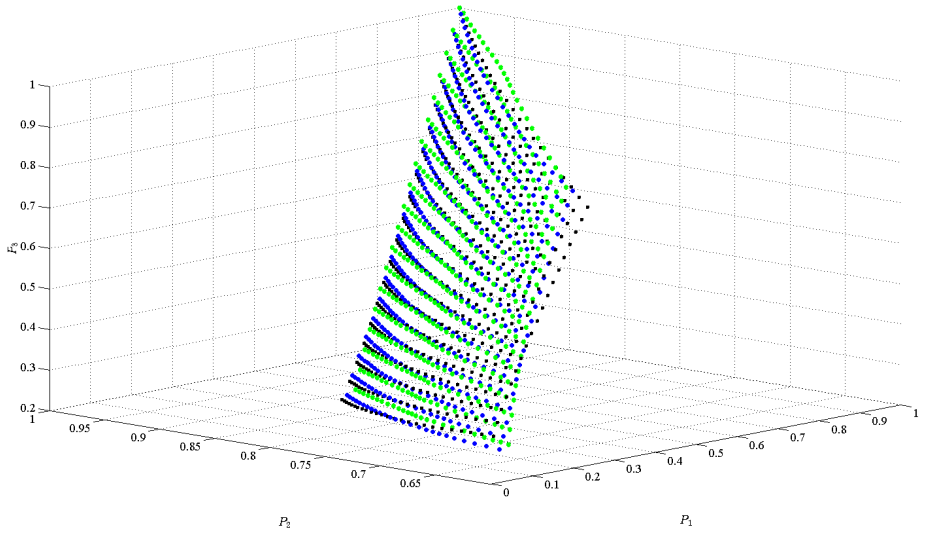


Figure 2.4: High-fidelity mathematical models for three different turbine engine-types after applying linear transformation and normalizations of data from Figure 2.3.

provides practicality in applying the following techniques for turbine condition monitoring to a large fleet of turbines.

In the examples shown above, we are utilizing five operating parameters; however, as can be seen above, the model is fully scalable if the user wishes to include new operating parameters into the model. In doing so, it is possible the rank may change but this does not alter any of the conclusions presented here. In fact, it is worth mentioning that by construction, a reduced rank model eliminates the so called *curse of dimensionality* by utilizing only the most significant axes of information produced by the non-zero singular values [JWHT13].

2.5 Condition Monitoring Applications using TLS Models

Condition based monitoring has become incredibly popular in recent years. There has been a big push in the industrial gas turbine industry to create and implement algorithms for fault identification and prediction [Sch14] . This type of data analysis demand has propelled the advancement of conditioned monitoring and condition based maintenance. Here we present some

applications of condition monitoring using the model derived in the section describing Total Least Squares modeling.

The condition of the engine can be inferred by comparing the static operating point of the engine to the TLS predictions. The errors or residuals, can be trended over time to identify gradual or step changes in the engine condition as compared to predicted performance. During this process we may implement rolling T-tests on the given window of data to ascertain whether statistical changes have occurred in the operating condition of the turbine.

For the remainder of the chapter we will perform a full analysis on two individual turbines, one that has been identified as healthy and one that has been identified as having a fault. Both turbines are of the same engine-type, and have similar configurations so that comparisons can be made.

2.5.1 Chi Squared for Goodness of Fit

Very often, when applying a model to real data, one begins the analysis by asking if the measurement is consistent with the model. We may answer this question by calculating the statistics of the measurement. If $\boldsymbol{y}$ is a measurement vector, then $\hat{\boldsymbol{y}} = C\hat{\boldsymbol{\varphi}} + \hat{\boldsymbol{\eta}} = C\boldsymbol{b}$ and the covariance matrix is $\Sigma_y = E\{(\boldsymbol{y} - \hat{\boldsymbol{y}})(\boldsymbol{y} - \hat{\boldsymbol{y}})^T\}$. From these, we have

$$0 < \rho^2(\boldsymbol{y}) = (\boldsymbol{y} - \hat{\boldsymbol{y}})^T \Sigma_y^{-1} (\boldsymbol{y} - \hat{\boldsymbol{y}}) << T$$

where T is some small, predefined threshold value. Note that the region is an ellipsoid with component mean values of 0 and our statistic, $\rho^2(\boldsymbol{y})$ is sometimes referred to as the *Mahalanobis distance* [And85]. If each component measurement follows a Gaussian distribution, then $\rho^2(\boldsymbol{y}) = \chi^2(\boldsymbol{y})$ and T may be chosen to be the corresponding value of the probability that y lies in the ellipsoid of radius $\rho(\boldsymbol{y})$. For example, with $n = 5$, then $T = \chi^2_{0.95}(\boldsymbol{y}) = 11.07$.

In Figures 2.5 and 2.6, we display the probability that a given measurement fits the

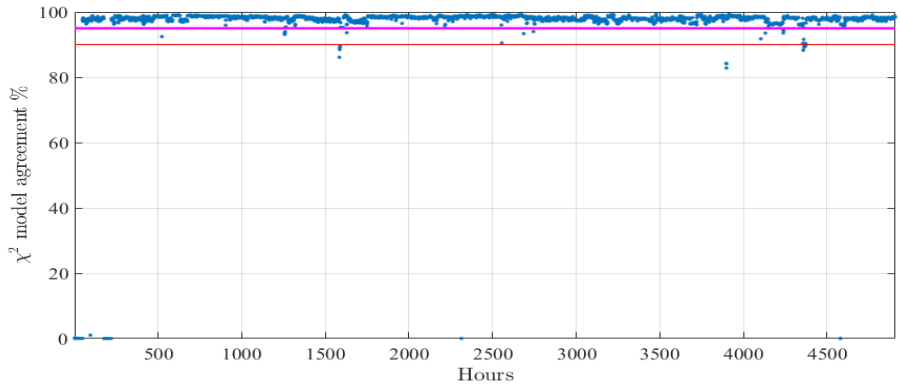


Figure 2.5: χ^2 probability of measurement to model fit applied to the case study: good engine. Note probability well above 95%.

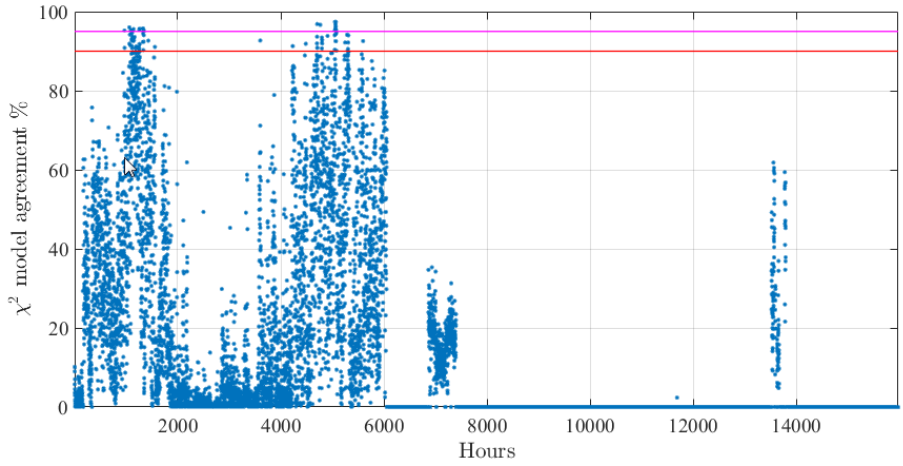


Figure 2.6: χ^2 probability of measurement to model fit applied to the case study: bad engine. Note probability rarely exceeds 90%, a strong indication that the machine is not functioning properly.

model, by evaluating the chi-squared statistic for each observation in time. These figures can be created by numerically computing the table of the chi squared values at differing probabilities, for example 0, 0.05, 0.1, ... , 1 for a given set of degrees of freedom and then interpolating a given $\rho^2(\boldsymbol{y})$ to the table. Our good turbine has probability of 95% or above, while our bad turbine is routinely close to 0%.

The chi squared statistic can be used as an overall measure of observation agreement with the model and an early predictor of turbine health changes. The data include both shutdown and startup data, which partially accounts for our good engine displaying some points below our

threshold value at 90%. In practice, the operator would use judgment on how many data points need to be below the threshold value before the determination would be that the engine warrants a closer investigation. For example, an operator may require five consecutive hours of operation where model agreement is below the threshold value before a more thorough investigation is warranted. In that scenario, our good engine would be flagged once whereas our bad engine would be flagged for more than 97% of its operation.

2.5.2 Residual Analysis

Above we noted the usefulness of the chi squared statistic; however, while this can lead an operator to the conclusion some part of the turbine is degrading or malfunctioning, it does not give any additional information as to what component of the turbine is degrading or malfunctioning. In order to facilitate a diagnosis of the root cause, we must examine the individual residuals. That is, for our inputs into the model, $p_1, ..., p_n$, our model returns $\hat{p_1}, ..., \hat{p_n}$ so that we may define the component residual, $r_i = p_i - \hat{p_i}$ for $i = 1, 2, ..., n$.

In industry, tracking of time series values of sensors is common, and some systems such as the InSight System of Solar Turbines even has virtual parameter tracking. In few instances, changes in the state of a single variable can lead to the root cause of a malfunction; however, many malfunctions or faults are a result of multiple components becoming worn, fouled or broken. In these problems, tracking of a single parameter may not provide enough resolution to identify the cause, especially early on when an operator may have a chance to fix the problem before component failure and engine shutdown.

The advantage of analyzing the residues produced from the model and measurements is that we have an easily calculated benchmark for each measurement and the combinations of residues gives greater resolution to an operators fault analysis. Our method is superior to common OLS regression, as OLS regression will return only a prediction of a single variable. Indeed, using a reduced rank model allows one to produce estimates for each input variable while not biasing

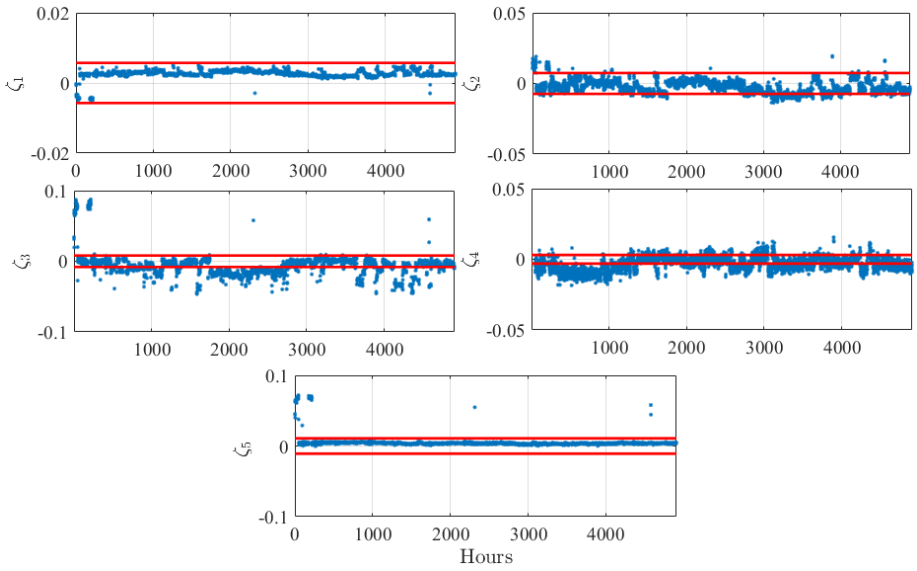


Figure 2.7: Time series residuals for our case study: good turbine. From the normalization, errors are represented as percent error. ζ_i corresponds to P_i, and is generated by taking the predicted value minus the measurement. The red lines are the 95% confidence intervals based upon the high fidelity mathematical models.

any single input variable's significance in the calculation. Continuing with our case study, we present Figures 2.7 and 2.8. We use $C = I$ since all parameters are available with data. Note that our healthy engine has operating error compared to the model within $\pm 5\%$, which shows normal operation. The threshold value for residuals is based on empirical testing of a set of 24 engines we have data for and that we can determine the health of. A different story is seen in Figures 2.9 and 2.10. Here we see departures in the residuals across multiple parameters. Specifically note the scale of the y-axis for ζ_4, [-2,2]. In many situations, it is not immediately identifiable what the cause of these changes were by simply examining the operating parameters. In our case study, we have chosen our unhealthy engine to be one with a malfunctioning sensor. The sensor is faulty from time $t = 0$ through $t = 6000$ and then malfunctions completely from then on. Note that in the range $t = 0$ to $t = 6000$, our residues show only minor deviations but our ρ^2 statistic shows very low probability that our measurements fit the model of a healthy engine. We turn to the either $\boldsymbol{\eta}$ or C to reduce the confidence in a given measurement variable or remove it entirely. We may perform a type of sensitivity analysis by varying the individual parameter's contribution and

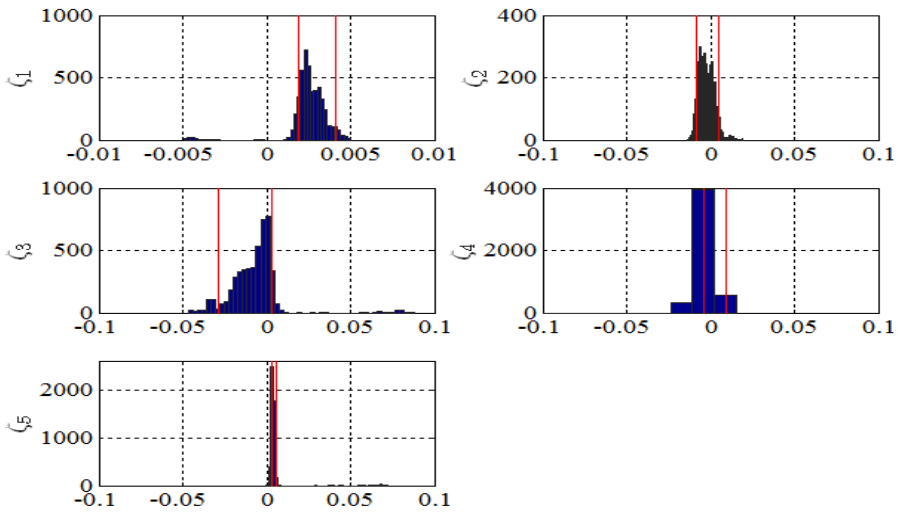


Figure 2.8: Distribution of residual values, ζ_i for our case study: good turbine. The red lines are the 95% confidence intervals based upon the high fidelity mathematical models.

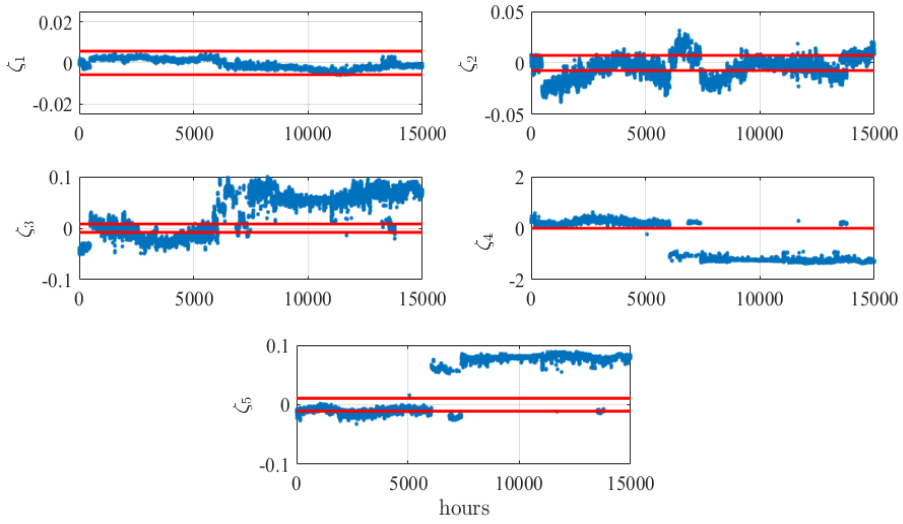


Figure 2.9: Time series residuals for our case study: bad turbine. From the normalization, errors are represented as percent error. ζ_i corresponds to P_i, and is generated by taking the predicted value minus the measurement. The red lines are the 95% confidence intervals based upon the high fidelity mathematical models.

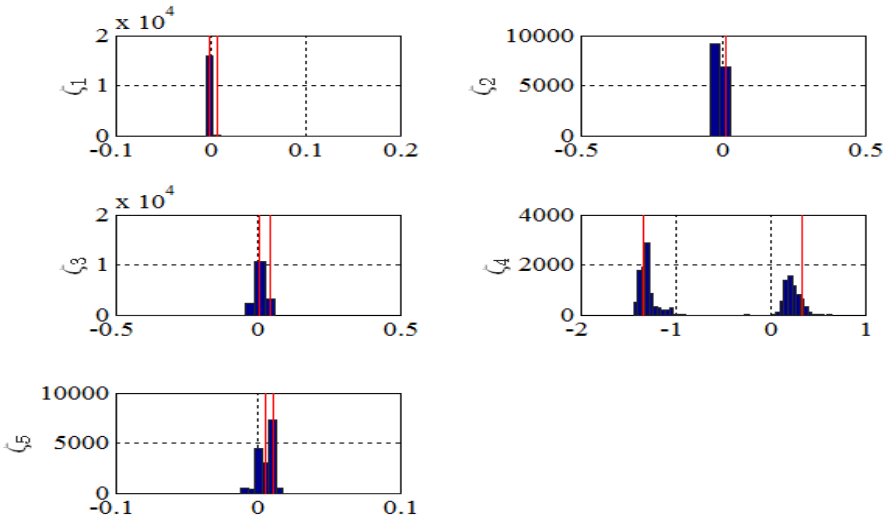


Figure 2.10: Distribution of residual values, ζ_i for our case study: bad turbine. The red lines are the 95% confidence intervals based upon the high fidelity mathematical models.

then by performing appropriate statistical tests on the average residual values for a given time interval. Returning to our case study, we see that one particular parameter seems to be the culprit causing large residue values. Once we decrease our confidence in the parameter by altering the corresponding component of $\boldsymbol{\eta}$, all other parameters fall back in line with our acceptable residue tolerances as seen in Figure2.11. Note, the parameter of interest is still significantly out of range, but this is what we expect by altering $\boldsymbol{\eta}$ if in fact, there is a fault we are trying to isolate. Hence, we conclude that the fault must be associated with this particular parameter and is independent of the other parameters. We could have tried removing the suspected culprit parameter by setting the corresponding diagonal value of C to 0 from 1. Then, $C_{3,3} = 0$, and our predictions are made without information from parameter 3. This is another key difference in our modeling technique; through the power of linear predictors and reduced rank models, we are able to predict out an original training parameter if we do not have field data corresponding to it, or if we wish to mitigate that parameter's effect on our predictions.

To summarize, we have just taken turbine sensor input, all of which are varying to different degrees, then applied our model with varying levels of confidence in the input parameter's

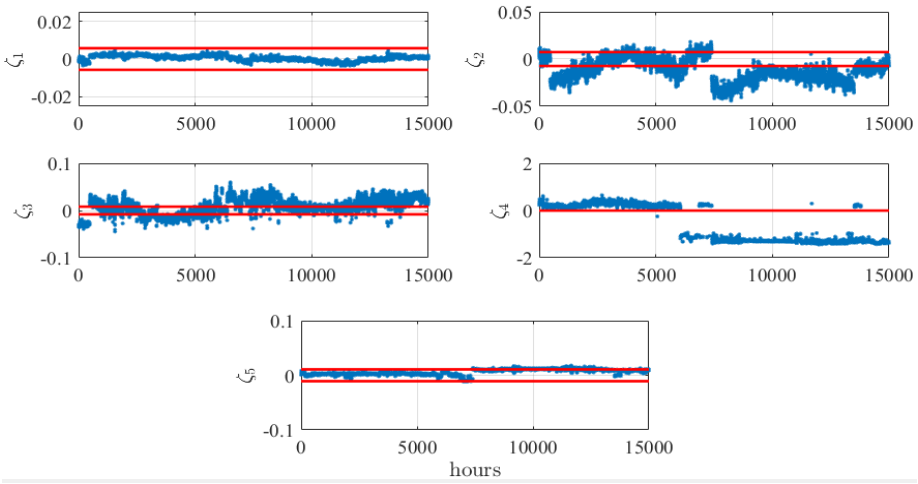


Figure 2.11: Time series Residuals for our case study: bad turbine, after altering η. Note the decrease in ζ_3 and ζ_5. ζ_4 remains well out of range, as expected based on changing η. The remaining parameters' errors are now closer to zero mean indicating normal operation.

contribution until we have found an optimal confidence such that the input parameter residues have reduced back within acceptable tolerance ranges. The fact that the optimal confidence weighting is isolated to reducing a particular input parameter's confidence weight allows us to conclude where the fault exists.

2.6 Conclusion

In this chapter, we first created a training set of data using a high-fidelity, physics based turbine models over the expected operating regime. We then identified the required complexity of the model by implementing singular value decomposition on the covariance matrix of the training data. From this, we derived a reduced rank model using Total Least Squares that maps static behavior of a turbine operating in closed loop. We then formulated a minimal variance predictor that operates on measurement data and that is capable of incorporating uncertainties and domain knowledge of measurement noise and accuracy to further refine the model. Following this, comparisons of predictions of the operational parameters to the actual measurement data were made and analysis of the residues was conducted to infer the health of the engine. We

calculated how closely the field engine is operating to the reference model by implementing the Mahalanobis distance and observing that errors from modeling were within 1% variance and zero mean as well as near Guassian in distribution. This gave us our 95% confidence intervals for our residue analysis. We discussed how binomial or non-Gaussian residual distributions of field data indicated degradation. To test our method, we selected two turbines, one healthy and one with a known fault and performed the above analysis process to both turbines and presented the results side by side for reader comparison. We showed how a sensitivity analysis could be performed on the user confidence in individual measurement inputs and how this allows a user to hone in on the root cause of a malfunction.

Future work should aim to untangle residue variation information. It is likely that while performing the sensitivity analysis, the variation of measurement confidence weights may not yield a single contributor as was shown in the case study. Some early experimentation with known faults has shown promising results. By applying additional machine learning classification methods to the individual residuals, we can describe certain faults that the classification methods have been trained for. We plan to continue investigating this track of analysis.

Chapter 2, in full, is a reprint of the material as it appears in ASME Turbo Expo 2016. Allen, Cody; de Oliveira, Maurcio, ASME 2016. The dissertation author was the primary investigator and author on this paper.

Chapter 3

Gas Turbine Machinery Diagnostics: A Brief Review and a Sample Application

3.1 Introduction

Chapter 3 proposes using machine learning algorithms in the identification of specific components of the GT system. The motivation for this idea continues to come from Sections 1.1.1 and 1.1.2. The utility of such an application is evident in instances when an operator or owner is trying to monitor the health of many of the same line of turbines in his or her fleet, and needs to know which turbines have which specific parts. A service provider of EHM falls into this category. Note that EHM providers can be both Original Equipment Manufacturers (OEMs) and non OEM's. In the case of the OEM, original engineering documents should be available in any analysis. However, it is possible that such information is incorrect or that parts have been replaced without all documentation being updated. Such a case arises when a customer has monitored the health of their turbine on their own for the first part of the turbine's life and then for the second part elected to have the OEM take over monitoring responsibility. In these circumstances, not all

of the applicable maintenance records are readily available. In the case of non OEM's, the EHM provider may not even have access to engineering documents. In either case, it can be difficult or impossible to ascertain specific component information.

This chapter seeks to highlight the importance of understanding the source of measurement data to be used in EHM algorithms. First, we tour the space of mathematical algorithms commonly used in industry for the monitoring of mechanical systems and in particular, rotating machines and gas turbines. The remainder of this chapter is a more in depth look at three 'classification type', machine learning algorithms which will be utilized in a case study that follows. In the case study, a subset of 25 gas turbines is selected, in which each fuel valve installed on each machine is known. Machine learning techniques are implemented in order to classify which fuel valves are installed on which engines, and results are examined and analyzed. The Chapter concludes with ideas to extend the accuracy of these methods as well as future research areas for such applications of machine learning methods.

3.2 Chapter Notation

ML	Machine Learning
FT	Fourier Transform
WT	Wavelet Transform
AR	Autoregression
ARMA	Autoregressive Moving Average
NN	Neural Network
HMM	Hidden Markov Modeling
SVM	Support Vector Machine
LR	Logistic Regression
DT	Decision Tree
KF	Kalman Filter

IoT Internet of Things

OEM Original Equipment Manufacturer

EHM Equipment Health Monitoring

3.3 Machine Learning Algorithms for Equipment Health Management

Statistical Learning refers to a vast set of tools for understanding data [JWHT13]. A direct definition of machine learning comes from Murphy: "Machine Learning is a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data, or to perform other kinds of decision making under uncertainty." [Mur12]. Broadly speaking, ML methods fall within two broad categories: *supervised* or *unsupervised* learning. Supervised learning refers to situations in which there are data consisting of inputs and output(s). In this case, information is obtained by relating the inputs to the corresponding outputs, where the relation is generated by the particular ML algorithm chosen. In the unsupervised case, the data only contain inputs. The association between inputs and outputs cannot be directly inferred, and instead, relations about the inputs are created.

Machine Learning as defined above is not a novel idea; in previous decades, the process of identifying relationships between inputs and outputs of a system has been called system identification, artificial intelligence, and pattern classification [Lju99]. In the remainder of this section, some common algorithms for machine monitoring applied to rotary machinery systems are discussed. This section ends with a brief review of some standard classification algorithms that will be used in the presented case study.

Lee, Wu, et al present a summary table of so called "Prognostic Health Management" tools that relate common learning/identification algorithms to typical rotating machinery components. The following sets of machine components and their common failures, algorithms for detection,

come primarily from the exquisite work of Lee, Wu et al [LWZ+13]. The addition of a valve section seeks to keep the structure of Lee, Wu et al. The technical details of the algorithms listed in each of the following subsections can be found in textbooks on machine learning and/or frequency domain analysis [Lju99][TH09][Mur12][Cha12]:

Bearings: Typical issues and failures include the outer-race, inner-race, roller or cage. Common measures include vibration signals, oil contaminant inspection and, less commonly, acoustic signals. Common features seen in data are vibration characteristic frequencies, metallic debris and sharp pulses of high intensity. Common detection algorithms include Fourier Transforms (FT), Wavelet Transforms (WT), Autoregression (AR) Frequency Spectrum, Neural Networks (NN), Hidden Markov Modeling (HMM), Support Vector Machines (SVM) and Principle Component Analysis (PCA).

Gears: Typical issues and failures include manufacturing error, teeth missing or teeth erosion, and gear cracks. Common measures include vibration signals and oil contaminant inspection. Common features seen in data are vibration characteristic frequencies and metallic debris quantity. Common detection algorithms include Fourier Transforms (FT), Wavelet Transforms (WT), Autoregression (AR) Frequency Spectrum, Neural Networks (NN), Hidden Markov Modeling (HMM), Support Vector Machines (SVM) and Kalman Filters (KF).

Shaft(s): Typical issues and failures include unbalance, bends, cracks and misalignment resulting in rub. Common measures include vibration signals and harmonic frequency components. Common features seen in data are vibration characteristic frequencies and system modal characteristics. Common detection algorithms include Fourier Transforms (FT), Autoregressive Moving Averages (AR)MA, Neural Networks (NN), and Support Vector Machines (SVM).

Lastly, we introduce a mechanical element that does not rotate, and so does not generate waveform data as the other elements do. This inclusion serves two purposes: (1) to show that algorithms and machine learning can be applied to non-rotating machinery data, and (2) to show that many of the same algorithms are used regardless of the dynamics of the system.

Valves: Typical issues and failures include deadband, hysteresis and stiction (sticking due to friction)[Rue99]. Common measures include upstream/downstream pressures, temperatures, command and position feedback signals. Common features seen in data are response frequency shifts and time domain statistical characteristics. Common detection algorithms include Neural Networks (NN), Support Vector Machines (SVM), and various Regression applications.

3.3.1 Algorithms

As seen by the definitions stated at the beginning of this section, ML algorithms are also supposed to make decisions as well as recognize patterns. Loosely speaking, this describes *classification* algorithms. These algorithms classify data sets (or rather, partition data in parameter space and associate the various partitions with some type of identifier) and can be both supervised or unsupervised algorithms. In the table below, a summary of algorithms available in the MATLAB® classification learner is presented. We discuss briefly the mechanics of the three chosen algorithms which will be presented in the case study later in this chapter. The three algorithms were chosen on the following basis: they needed to be available in the Matlab machine learning library and there needed to be both linear boundaries and non-linear boundaries.

Algorithms In Matlab Classification GUI
Decision Trees (CART)
Discriminant Analysis
Logistic Regression
Support Vector Machines
Nearest Neighbor
Ensemble Methods

Decision Trees: Decision trees are a way of stratifying the predictor space into n non-overlapping, exhaustive regions , $\mathcal{R}_k$, such that $\cup_{k=1}^{n}\mathcal{R}_k = \mathcal{R}$, where $\mathcal{R}$ is the entire predictor

space. Classification decision trees can be used to predict a qualitative or discrete response. In implementation, after the tree algorithm has been trained, an observation is assigned the value of the most commonly occurring class of training observations in the region to which it belongs [JWHT13]. To "grow" or train a classification tree, recursive binary splitting is used. To perform recursive binary splitting, all predictors $X_1, ...X_q$ and all possible division points for each predictor are considered. Choice of the predictor and division point are determined based on some form of minimized error. The most popular choice of error function for classification trees is the *Gini Index*, $G = \sum_{k=1}^{W} 2p_{wk}(1-p_{wk})$ where W is the number of classes in the response and p_{wk} is the proportion of training observations in the w^{th} region, from the kth class. Here, $2p_{wk}(1-p_{wk})$ can be seen to be the variance of a binomial random variable[And85], so the Gini Index method seeks to minimize total variance across the k classes. Another relatively popular choice for minimizing error is the *Cross-Entropy* function; see [Mur12].

Logistic Regression: Logistic regression gets its name from the underlying function that generates the classifications, the *logistic function*, $\hat{y} = \frac{e^{a_0+a_1x_1+\cdots+a_px_p}}{1+e^{b_0+b_1x_1+\cdots+b_px_p}}$, where x_k is the kth predictor and $0 < \hat{y} < 1$. To make a classification, a threshold value T is chosen, where $0 < T < 1$, such that if $0 < \hat{y} < T$, $\hat{y} = 0$, otherwise $\hat{y} = 1$. By construction, this function is a binary classifier, although there are ways to use it for more than two classes [JWHT13]. Since the function is nonlinear, finding the coefficients a_k, b_j is typically done by using a numerical minimization technique on the objective function, $\arg\min_{a_k,b_j} f(y-\hat{y})$, with $f = \|\cdot\|_l$, some specified l-norm. Popular optimization algorithms include but are not limited to 'Newton - conjugate gradient', 'Levenberg Marquardt', 'Iteratively reweighted least squares' and the 'BFGS' methods. These methods and more can be found in [JN06].

Support Vector Machines: The support vector machine algorithm is a variant of the support vector classifier, which is the implementation of the *soft margin classifier algorithm*.

Specifically, the soft margin classifier satisfies an optimization on M:

$$\max_{\beta_0,\beta_1,...,\beta_p,\varepsilon_1,\varepsilon_2,...,\varepsilon_p} M$$
$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1,$$
$$y_i(\beta_0 + \beta 1 x_{i1} + \beta 2 x_{i2} + ... + \beta p x_{ip}) \geq M(1-\varepsilon_i),$$
$$\varepsilon_i \geq 0, \sum_{i=1}^{n} \varepsilon_i \leq C$$

The support vector machine brings in *kernel* functions, $f(x) = \beta_0 + \sum_{i=1}^{n} \alpha_i K(x, x_i)$, where $K(x, x_i)$ represents the kernel function. The constraints must be adjusted accordingly; for example, $y_i(f(x)) > M(1-\varepsilon_i)$. Popular kernels are the *polynomial kernel* $K(x_i, x_i') = (1 + \sum_{j=1}^{p} x_{ij} x_{i'j})^d$ and the *radial kernel* $K(x_i, x_i') = exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2)$. For a more technical review of SVMs, see [TH09].

3.4 Case Study: Gas Valve Classification

In this section, we present a case study: the classification of the type of fuel valve in use in an industrial gas turbine, based only on standard signals measured on a gas turbine. A priori knowledge is used that the population of data has only two possible valve types, and so an individual set of data points at time t, is categorized as *valve 1* or *valve 2*.

This case study and the motivation for this chapter come from real life lessons learned from attempting to jump from raw machine output data to engine condition monitoring [C.A16]. In this previous study, uncertainty in the fuel flow measurements was attributed to insufficient knowledge about the engine systems under investigation, specifically that there were different fuel valves present in the same fleet. Recognizing that the crucial step of understanding how the data was generated had been overlooked, we sought to classify the fuel valves present in the fleet

in question.

The choice of the fuel valve classification to illustrate the proposed machine learning algorithms is indeed a practical one. From the perspective of the OEM, maintenance records are not always available. Customers can and do replace hardware on their own without leaving a sufficient paper trail. As an example, consider a customer who has monitored the health of their turbine on their own for the first part of the machine's life and then for the second part elected to have the OEM monitor the machine. In this case, it is often found that not all maintenance records are available. Furthermore, even when knowledge of the valve part is available, it is still useful to classify the operation of the valve as healthy or unhealthy. Such a classification can be done using the same techniques as proposed in the two-valve case study. These realizations provide the current problem statement. **Can specific components of a gas turbine be predicted using machine learning techniques?**

This case study attempts to provide an initial answer to this question. A fleet of 25 similar turbines is analyzed, where it is known that each turbine has one of two possible fuel valves installed. The fuel used in the sample fleet is known to be of similar composition but time varying. Using measurements of the pressures and temperatures of the gas, command of the fuel valve and standard operating parameters like shaft speed, power and ambient conditions, three separate types of ML models are trained and used to classify a given engine's fuel valve as "valve 1" or "valve 2". The first classification approach uses data from a high fidelity engine simulation, in which site conditions and engine setpoints were assumed. The second classification approach uses actual field engine data from the fleet of engines. In both cases, the three chosen ML algorithms are trained and then applied to the fleet. Lastly, results and conclusions are discussed.

3.4.1 Classification Algorithms

In this section, a subset of ML classification algorithms is used to identify the fuel valve in use, "valve 1" or "valve 2". This is a *supervised learning* problem, and requires data to "train" the

selected model to be used in creating predictions. The implementation of the selected algorithms are carried out in MATLAB® using the add-on toolbox: Statistics and Machine Learning. The process of selecting variables or (features as they are sometimes referred to) selecting and training the model and then validating the predictions is carried out through this toolbox.

The pre-programmed classifiers make base assumptions in order to provide ease of use. In certain algorithms these assumptions can be manipulated. There is additional functionality residing in the actual function calls, meaning that if one is willing to do some programming to implement the classifier functions directly, one will have more options available to change base assumptions. The more seasoned ML practitioner will be able to modify base assumptions to better tune the algorithm in order to classify more accurately.

3.4.2 Data Used in the Case Study

The data used in the first classification is generated from a high fidelity, physics-based simulation model. The data generated from this process is not always the expected value of a given operating point, but rather can be the minimum value required given other conditions of the engine. For example, a supply pressure may be x (psig), where x is the minimum value needed to provide the other operating variables supplied in the simulation. Care has been exercised in trying to reduce the number of parameters that are susceptible to this type of output, but certain parameters such as upstream pressure are outputs of the simulation like this. Identical inputs are used in the simulation except for which valve is assumed on the engine, and hence, which valve geometry is used. Ambient conditions such as elevation and pressure losses in various components of the engine are held constant for each valve simulation, and are representative of typical values at sea level.

Figure (3.1) shows some key normalized operating values for the two fuel control valves. At lower command positions, the inlet and outlet pressures are all but on top of each other, and the two valves exhibit very little deviation in operational modes. Figure (3.2) reveals that other

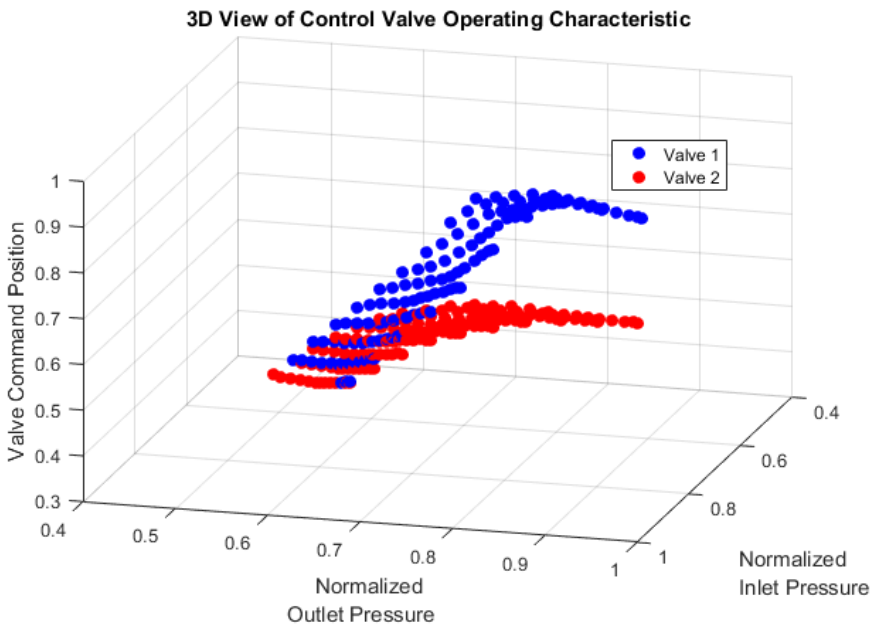


Figure 3.1: Normalized Operating Values for Control Valves.

parameters exhibit more separation. Notably, the normalized fuel flow for varying compressor discharge pressure levels and command positions yield different operating points in 3D space.

3.4.3 The Reality of Field Data

A subset of 25 engines is selected to have valve classifications performed. These engines all run on the described fuel type, and are assumed with high confidence to have one of the two valves we have trained our classification models on. For each of the given engines, the number of operating data points differs. At this point, some modeling decisions and assumptions must be made in order to arrive at our overall goal of taking a given set of measurements from an engine forward to a prediction of which valve the engine is running.

The data generated by the 25 selected machines come from real, operating Solar units around the world. As such, the systems may have degradation, may have noise on the sensor readings, or may have valves other than the two expected in this sub fleet. Additionally, not all of these units come from the same geographic location, and so, ambient conditions may be different

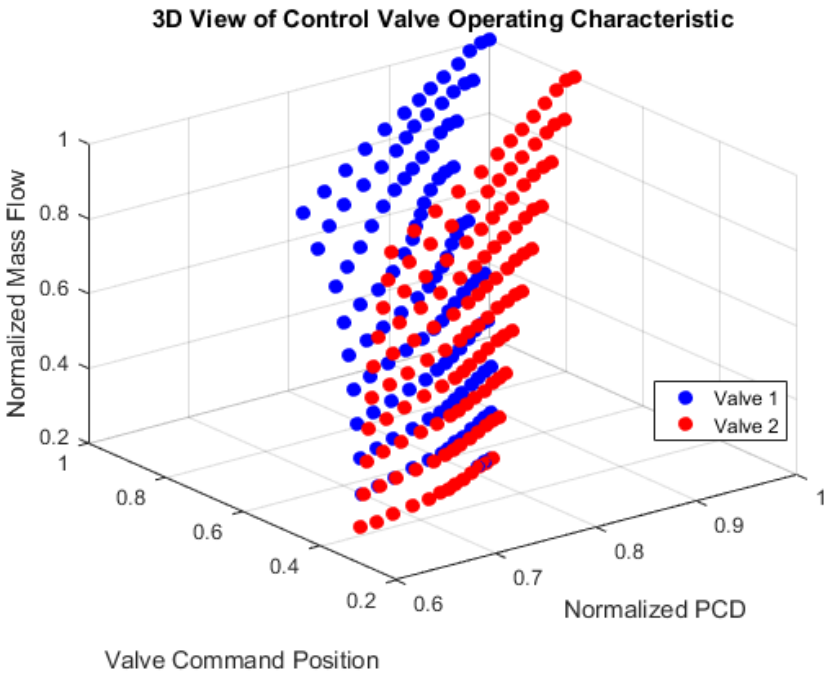


Figure 3.2: Normalized Operating Values for Control Valves.

for the given units at various times (elevations are relatively similar). With this stated, some assumptions are made regarding the data:

– The valves that were found in Solar engineering schematics are the current valve in use at customer locations. From this, we assume there are only two possible valves in use for any given engine, and every engine has one and only one valve.

– The majority of the data generated by a given engine has been generated by a clean and healthy engine, and more specifically, a clean valve, free of degradation. This fuel is known to foul injectors and valves, which will change the dynamics of the valve flow. However, we assume that this type of data is insignificant due to regular maintenance and cleaning of the selected group of engines.

– The fuel type and composition across the fleet is similar. This assumption is problematic,

as the fuel's specific gravity and lower heating values can vary. In fact, the wobbe index on coke oven gas, the fuel in this study, can vary up to 30% in extreme cases [JRS+16].

3.5 Training and Model Validation

MATLAB®'s classifier GUI makes training the ML models exceptionally easy. When starting a new session, the GUI automatically makes the user select which variables to use in the model generation process and what type of validation to perform. For each classification algorithm, 10-fold cross validation is performed. This means that for a given data set, the data is partitioned into 10 equal sized subsets, then 9 of the 10 sets are used to train while the 10th set is used to measure accuracy. This is repeated until each"hold out" set has been used once to measure accuracy. Then, the 10-fold accuracy is the simple average of each individual experiment's accuracy. This method gives an implicit sense of confidence in the algorithm's use on new data and is a standard practice in ML [TH09]

The actual process of training the algorithm and then validating is carried out behind the scenes by Matlab. Once the user has decided which algorithm to train, he or she may open the "Advanced Settings" for the given algorithm, if available. In this window, additional properties particular to the chosen learning algorithm may be changed. For example, in the DT's advanced settings, two possible selections are the number of maximum branches to grow and which splitting criterion will be implemented. In the SVM advanced settings, one may choose the type of kernel function to be implemented along with various data scaling options. After the user is satisfied with the settings to be implemented for the learning algorithm, one simply pushes the "Train" button, and as if by magic, the classifier is trained. The displayed accuracy is the 10-fold accuracy produced from the algorithm validation that Matlab performs. Of note, when implementing these algorithms in the Matlab command line, a deluge of data is returned in structure elements. One has access to the model, model coefficients, standard measures of error, accuracy metrics and

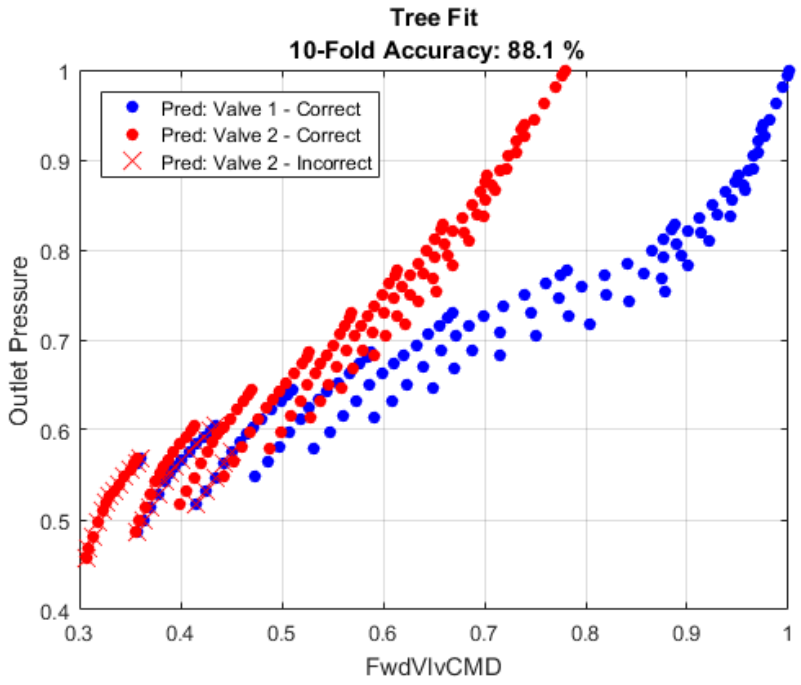


Figure 3.3: Predicted Valve based on Decision Tree method.

statistical measures of interest such as t-values and confidence intervals, when applicable.

Figure (3.3) shows the results of using the DT classifier, in which the Gini Index splitting criterion has been used, and a max number of splits was set to 10. The DT model achieves 88.1% accuracy in the 10-fold cross validation. Similar visualizations can be created for the LR and SVM models. The LR model achieves 95.8% accuracy while the SVM model achieves 94.6% accuracy. The boundary of the two valves is non-linear, so we expect SVM to do well since its kernel function is quadratic. The LR model does well, although it is a linear classifier in its boundary. The DT approximates a nonlinear boundary with a collection of linear partitions but does well at 88.1% accuracy.

The learned classifiers appear accurate, based on the 10-fold cross validation, with the lowest accuracy at 88.1% given by the DT classifier. Based on these results, it seems reasonable to think that application of the algorithms to actual field data will yield worthwhile results.

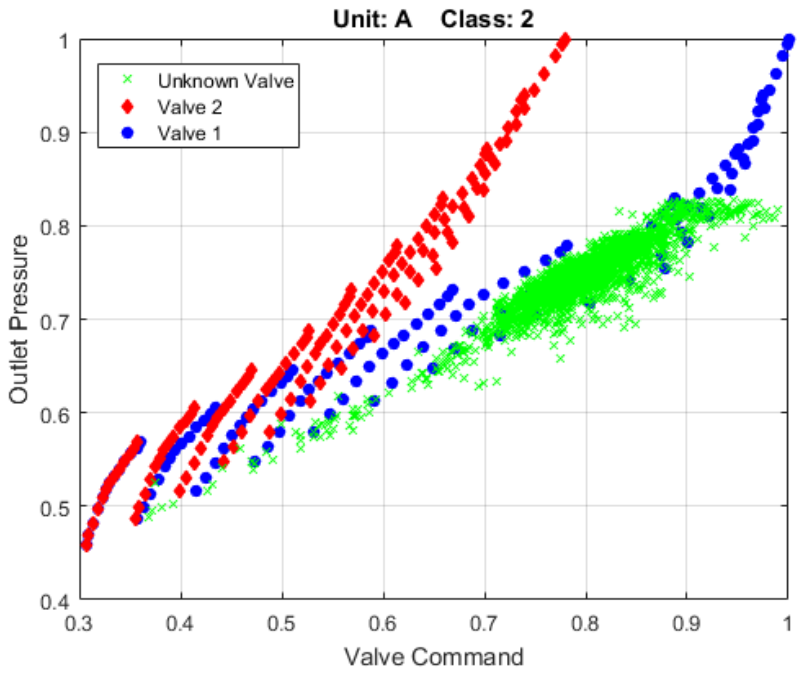


Figure 3.4: Unit A: normalized operating values compared with training data for classification algorithms.

3.5.1 Field Experiments and Results: Model Training from Simulation Data

Now that base assumptions have been addressed, a presentation of the results follows. Three specific units will be presented for illustration purposes. Unit A represents a well running unit, where the fuel system is healthy and the valve is assumed to be valve 2 based on engineering documents. Unit B represents a well running unit where the fuel system is healthy. Unit B's valve is assumed to be valve 1, although this could not be confirmed due to missing engineering documents. Lastly, Unit C represents a unit with either a fouled valve or varying Wobbe index (or both). Unit C's valve is assumed to be valve 1 according to engineering documents.

Figure (3.4) appears to be in-line with expectations of the fuel curve of valve 1, but note that the valve is actually classified as valve 2! This is a good example of how a projection from 6-D space to 2-D space can give misleading information, as it pertains to classification algorithms.

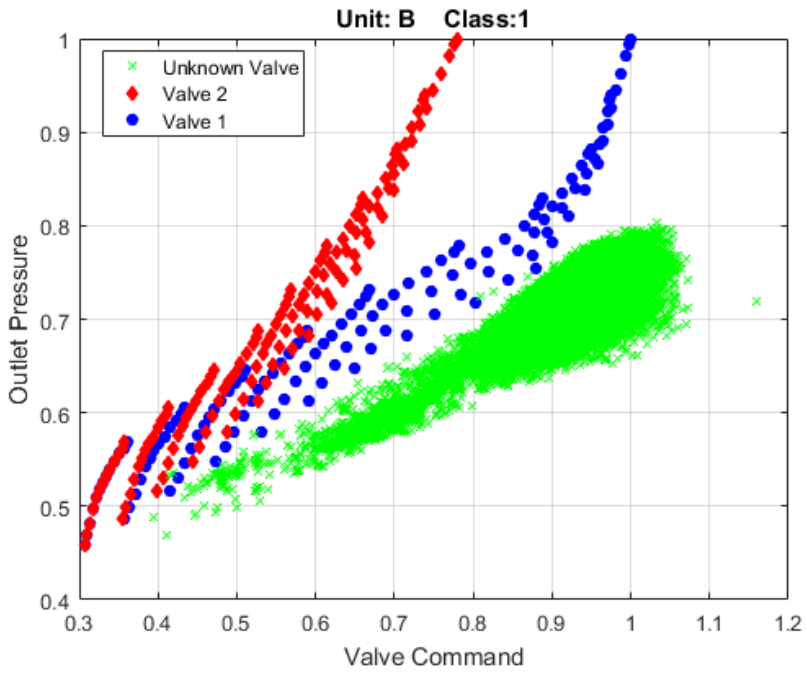


Figure 3.5: Unit B: normalized operating values compared with training data for classification algorithms.

Figure (3.5) shows Unit B has some more variation than did Unit A, but is almost completely below both training curves. The resulting classification does agree with the closer of the two valves, as Unit B is classified as valve 1.

Lastly, figure (3.6) shows Unit C with a dirty fuel system, or degraded valve. Again, the data is almost completely below both training curves but also contains decreased pressures between 0.8-1 command. This may be data pertaining to a fouled valve; however, the source of this uncharacteristic portion of the data is outside the scope of this chapter but is included to illustrate the non-ideal nature of real field data.

The reader may be wondering about how the valve command can be up to 140% open. This is due to the normalizations that occurred at the onset. In order to keep all values unbiased, normalizations based on the training data maximum values were used. Simulation values of command position never reached 100% full open, and so normalized values on real engines exceed 1.

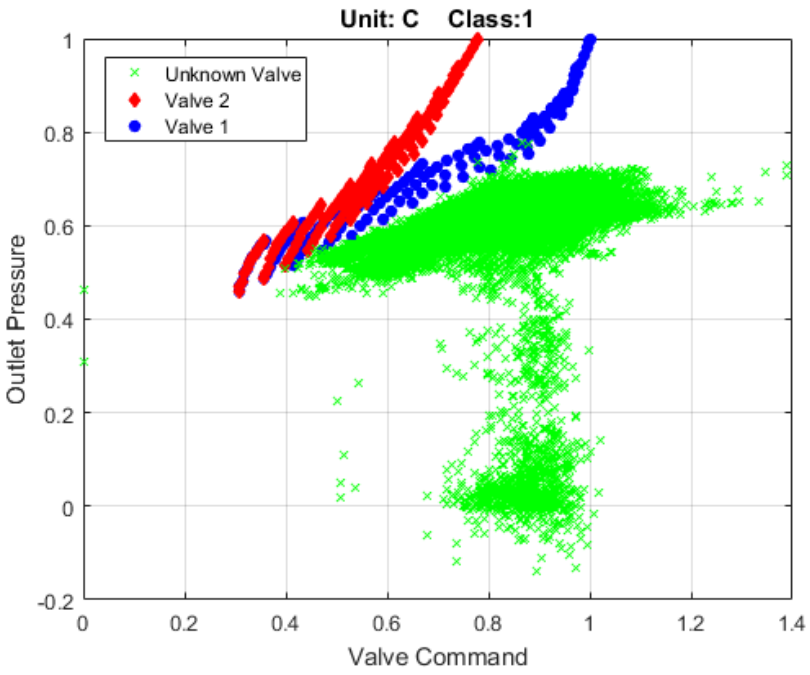


Figure 3.6: Unit C: normalized operating values compared with training data for classification algorithms.

Figure (3.1) shows the results of the experiment for all 25 units. The percentage for a given classifier and unit number is the percentage of operating points classified as "valve 2" for that unit's operating data. Any value over 50% is classified as "valve 2". A weighting algorithm is used to arrive at the final percentage for use in predicting the valve. As is seen, this implementation results in 64% accuracy in predicting the correct valve. This is a far cry from the mid-90s percent correct found when learning the training model. The most notable observation that can be made is that when the data is visualized in a 2D projection, almost all of the data is below both training curves. In other dimensions, similar results are seen. This suggests that data assumptions are at least partially incorrect. It is likely that the valves have some fouling, based on the type of fuel being run through them. It is also likely that the gas composition is not the same for each of the units. However, even with these assumptions violated, the data appears to be relatively in line with the general curves seen in the projections. That is, it seems that with some scaling or translational shifts, the data will line up better to our training curves, which should

generate higher accuracy. While this is an interesting path forward, and well worth pursuing, it will not be pursued in this chapter. Instead, new training data is chosen, the models retrained and results analyzed.

Table 3.1: Full results of the three classification algorithms. Algorithms trained on data generated by high fidelity physics simulation software. Percentage for a given classifier is the percentage of operating points classified as "valve 2". Any value over 50% is classified as "valve 2", which is indicated with a "1" instead of "0". A weighting algorithm is used to arrive at the final percentage for use in predicted valve.

Unit	TREE	LOGREG	SVM	Predicted Valve	Actual Valve	Error
1	0.0%	0.0%	84.9%	1	0	1
2	2.3%	0.0%	70.3%	0	0	0
3	2.7%	0.0%	64.0%	0	1	1
4	0.2%	0.0%	90.4%	1	0	1
5	0.3%	0.0%	95.0%	1	0	1
6	1.0%	0.0%	0.0%	0	1	1
7	1.0%	0.1%	0.1%	0	1	1
8	0.2%	0.0%	95.9%	1	1	0
9	0.3%	0.0%	98.6%	1	1	0
10	0.3%	0.0%	97.5%	1	1	0
11	0.4%	0.0%	99.8%	1	1	0
12	0.2%	0.0%	99.8%	1	1	0
13	0.2%	0.0%	99.8%	1	1	0
14	24.5%	0.5%	67.5%	0	1	1
15	9.2%	0.0%	97.4%	1	1	0
16	17.9%	0.0%	97.2%	1	1	0
17	8.7%	0.0%	91.2%	1	1	0
18	0.3%	0.0%	27.2%	0	1	1
19	0.1%	0.0%	31.9%	0	1	1
20	0.6%	0.0%	98.7%	1	1	0
21	28.6%	0.0%	99.8%	1	1	0
22	30.3%	0.0%	99.5%	1	1	0
23	15.6%	0.0%	99.1%	1	1	0
24	32.0%	0.0%	99.7%	1	1	0
25	26.9%	0.0%	98.2%	1	1	0
Total Accuracy: 64%						

3.5.2 Retraining Using Field Data from the Fleet

From the presented results, it is clear that the real data is not as easy to classify as the data generated from the high fidelity physics simulation. However, curiosity got the better of us, and we wondered if this was actually true. A decision was made to change the training data from that given by the high fidelity simulation to two selected engines from the sub fleet of 25 engines. That is, two engines were selected, one with valve 1 and one with the valve 2, that seem

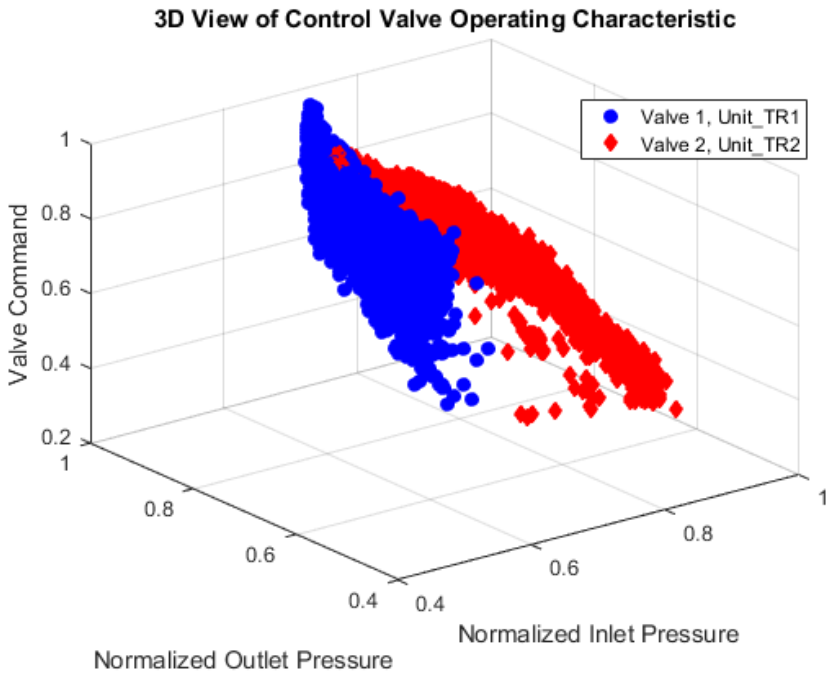


Figure 3.7: Training Data: data from two different engines, one with valve 1 and one with valve 2.

like candidates with little to no degradation, and healthy fuel systems, to be the training data for the classification algorithms. In selecting the two training engines, an effort was made to select engines that were most representative of the fleet; that is, machines that captured the mean running modes of the fleet. In selecting the valve 1 training engine, there was not much choice but in selecting the valve 2 engine, and thus such an engine was selected. Figure (3.7) displays the 3D projection of 3 of the 6-parameters used in training the classifiers. Note that there appears to be more separation between the two engines, and hence, two valves, which should allow the classifiers to determine a relatively robust boundary condition.

3.5.3 Field Experiments and Results: Model Training from Field Data

The process is as it was above, in which 10-fold cross validation is used in training the models, and the DT, LR and SVM models are trained and used for predictions. The model assumptions, data assumptions and model selection parameters are the same as well. Figure

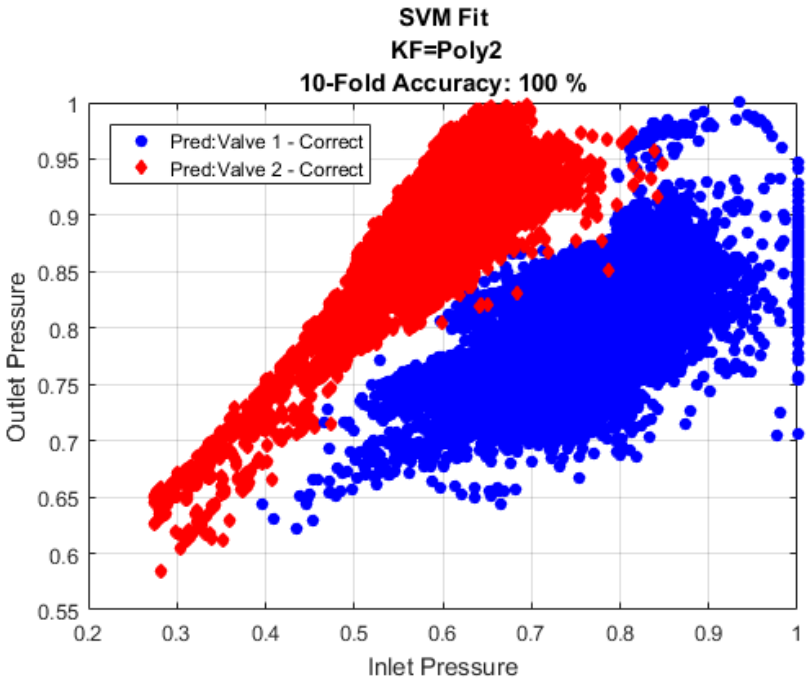


Figure 3.8: Training Data: data from two different engines, one with valve 1 and one with valve 2.

(3.8) shows the output of the trained SVM classifier; it achieves 100% accuracy in 10-fold cross validation for the two training engine's data! While this seems remarkable, enthusiasm is contained as we remember how high the accuracy was on the training set but how low the accuracy was on the real data as in our previous experiment above. Similarly, the LR model achieves full separation and 100% accuracy while the DT model has some points that are misclassified and yields an accuracy of only 99.7%. With these newly trained classifier models, the previous test units are re-examined.

Figure (3.9) shows how Unit A compares with the two training engines. After normalization of the parameters, it is seen that Unit A runs on the higher side of valve 2 in the projection of valve command versus outlet pressure. As the figure would suggest, this valve is classified as valve 2, an accurate classification by the ML classifiers.

Figure (3.10) shows how Unit B compares with the two training engines. After normalization, it is seen that this unit runs on the middle to lower side of valve 2 in the projection of valve

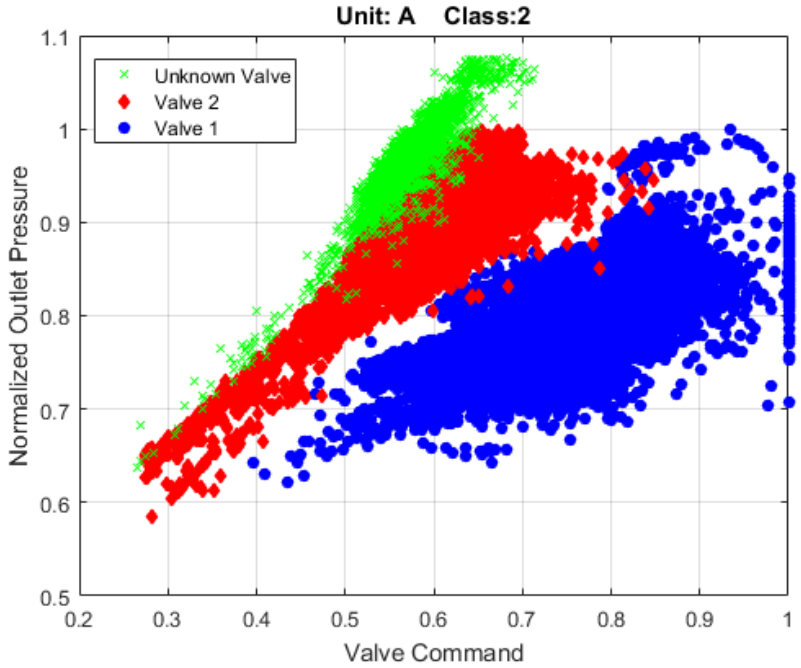


Figure 3.9: Unit A: normalized operating values compared with training data for classification algorithms. Training data from real engines.

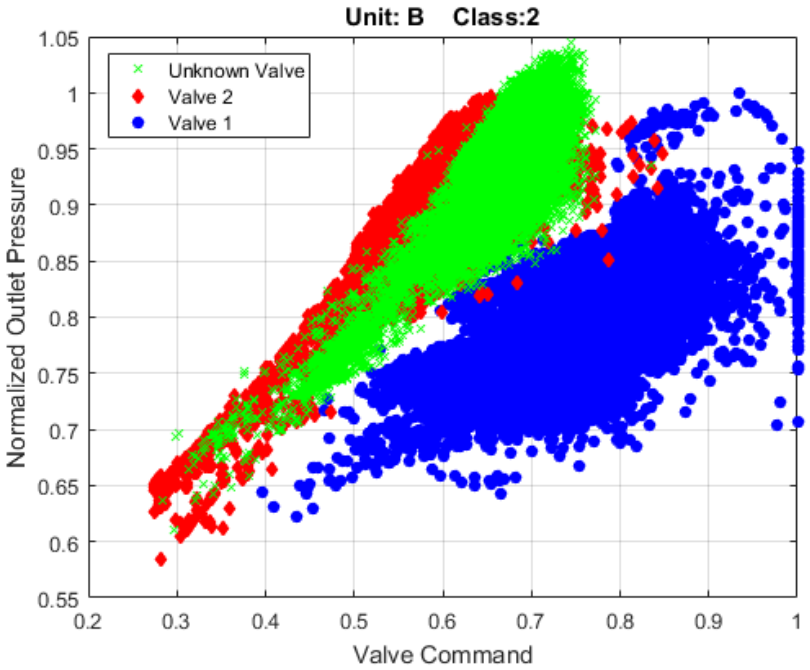


Figure 3.10: Unit B: normalized operating values compared with training data for classification algorithms. Training data from real engines.

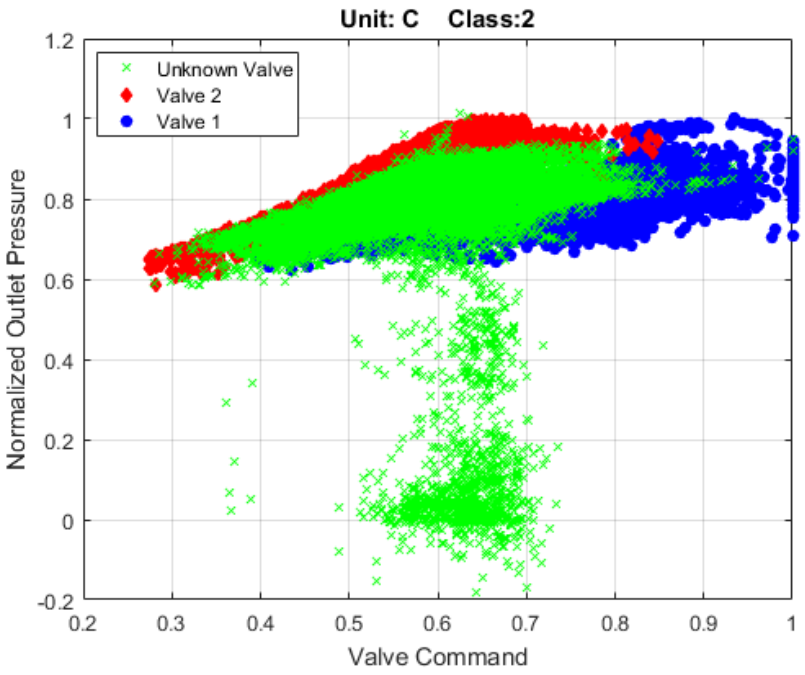


Figure 3.11: Unit C: normalized operating values compared with training data for classification algorithms. Training data from real engines.

command versus outlet pressure. This unit is classified as valve 2; however, this unit is one of the two with an uncertain valve. The authors suspect it is valve 1; however, the classifiers suggest it as valve 2. Further investigation would need to be done here to ascertain the true valve type.

Figure (3.11) shows how Unit C compares with the two training engines. After normalization, it is seen that this unit runs in the middle of valve 2 and valve 1 in the projection of valve command versus outlet pressure. This unit is the one with major noise on the data, potentially from degradation or poor running conditions. The classifiers determine that this valve is valve 2; however, its real identity is valve 1. When the suspected contaminated data is removed, the confidence in the valve being valve 2 actually increases considerably, which is not intuitive. Since our "actual" valve classifications come from original engineering drawings and schematics, it is possible a customer could have had the valve replaced without notifying Solar. This is one possible reason for the discrepancy in this valve classification. Another is simply that the classifiers are not able to fully delineate between the two valves. Even more so to question is

the selection of the representative engines for training the decision boundaries between the two valves! Recall that we only had 2 "actual" valve 1's, and two that were suspected to be valve 1, but not "confirmed". This constraint on engine and thus, valve selections for training may have lead us to train on an engine that truly is not representative of the parameter space of valve 1 engines. This uncertainty is hard to affirm without additional valve 1 data.

After running the fleet of 25 engines through the classification algorithms, we produce the same table of results as seen in table (3.1), and in table (3.2). Immediately, a much higher accuracy is noticed, 80% as opposed to 64%. The improvement suggests that the original models, trained on the high fidelity physics simulations, perhaps had wrong assumptions built into them. As noted above, there were a number of issues with the simulation data, such as gas fuel composition assumptions and site condition assumptions that would influence the simulation outputs. In addition to accuracy of classifications, a user may be interested in the percentage of false-positives or false-negatives which can be computed easily and are often useful depending on the ramifications of mis-classification. The false positive rate (classifier identifies valve 2 when actually valve 1) for the field trained classifiers is 12%, and the false negative rate (classifier identifies valve 1 when actually valve 2) is 8%.

An interesting question that arises is the question of whether fuel valve degradation is implicitly learned by the classifiers. That is, if there was some degradation in the data of the sub fleet, but degradation was also present in the two selected engines for training, perhaps these degradation states were implicitly encoded in the classifiers, so that the boundaries would account for this, something that simulation data certainly does not account for. Validation of this question would require a more detailed set of data from the selected packages, and likely an inspection of the valve, along with complete maintenance records to ascertain the state of the valve over the period of time in which data was generated. This is beyond the scope of this chapter, but would be worthwhile to understand in the future.

Table 3.2: Full results of the three classification algorithms. Algorithms trained on data from two selected engines of the sub fleet of 25 engines. Percentage for a given classifier is the percentage of operating points classified as "valve 2". Any value over 50% is classified as "valve 2", which is indicated with a "1" instead of "0". A weighting algorithm is used to arrive at the final percentage for use in predicted valve.

Unit	TREE	LOGREG	SVM	Predicted Valve	Actual Vavle	Error
1	0.0%	0.0%	0.0%	0	0	0
2	92.4%	61.5%	77.8%	1	0	1
3	95.7%	97.3%	90.8%	1	1	0
4	96.5%	94.7%	98.7%	1	0	1
5	85.2%	95.8%	94.0%	1	0	1
6	0.1%	100.0%	100.0%	1	1	0
7	2.0%	100.0%	100.0%	1	1	0
8	95.8%	7.6%	68.6%	1	1	0
9	85.8%	0.0%	32.3%	0	1	1
10	87.3%	0.7%	18.9%	0	1	1
11	74.8%	99.9%	77.3%	1	1	0
12	81.9%	100.0%	90.4%	1	1	0
13	88.8%	100.0%	97.4%	1	1	0
14	100.0%	100.0%	100.0%	1	1	0
15	99.9%	100.0%	100.0%	1	1	0
16	100.0%	100.0%	100.0%	1	1	0
17	100.0%	100.0%	100.0%	1	1	0
18	98.6%	99.9%	99.9%	1	1	0
19	99.9%	100.0%	100.0%	1	1	0
20	100.0%	100.0%	100.0%	1	1	0
21	98.6%	100.0%	100.0%	1	1	0
22	99.0%	100.0%	100.0%	1	1	0
23	96.2%	100.0%	100.0%	1	1	0
24	100.0%	100.0%	100.0%	1	1	0
25	100.0%	100.0%	100.0%	1	1	0
Total Accuracy: 80%						

3.6 Conclusion

This chapter has reviewed, in part, the current state of equipment health management and prognostic health management techniques in use in industrial settings for rotating machinery, and in particular, gas turbines. A presentation of application categories and types of components was given, along with relevant numerical algorithms and/or machine learning techniques used in analysis. In addition, it was shown how easily some of these algorithms can be implemented by using commercial computation environments. We presented a case study trying to predict the type of fuel valve in use on a given engine, which was part of a fleet of industrial gas turbines, by using only system signals such as pressures and command positions. The results suggested that it is certainly possible to identify the type of valve in use in a turbine system, if the selection class contains a small number of particular valve types and if there is enough variation in the operating space of the different valves.

The results of the case study reveal that both high fidelity model data and field data can be useful. While the high fidelity model data did not fit well with most of the field data, it was noted that gas composition variability was a salient factor. There are many advantages of the high fidelity model data though, namely that a tremendous amount of operating points can be generated with ease, the entire range of the machine can be explored, and inputs such as gas composition can be varied. Future work aims to create better simulation data that can be used as part of the training data for the learning methods.

Worthy of noting is the ability of machine learning classifiers to identify machine components based on a sample of running points. For each time t, the specified operating points were input into the various classifiers, and the output of valve type was recorded. A weighting method between the various classifier's output was devised and used to arrive at a final classification for a given engine's valve. The weighting method can be a simple average of the given classifiers or biased towards specific classifiers. This process has been performed with minimal input from an

engineering perspective. That is, for identification of a mechanical component that involves a process such as compressible gas flow through a valve, very little knowledge of fluid mechanics, thermodynamics or chemistry was needed to make a prediction on the type of valve in use, with relatively good accuracy. However, this statement excludes the requirement of knowing what valve type is actually in use, what measured parameters would be relevant to the analysis, and what type of gas fuel is in use for a given engine. Of course, to perform supervised learning, all of these items are necessary. This shows that optimal performance is found in the intersection of engineering knowledge and machine learning knowledge.

Indeed, the inclusion of additional engineering knowledge is very likely to increase the accuracy level significantly. For example, including the fuel flow or the valve CDA will provide additional information to the learning algorithms and result in better classification accuracy. Future work will begin to include additional engineering knowledge into the learning models. Fuel composition, fuel flows, enthalpy, specific heats and known valve CDAs provide additional degrees of freedom for the learning algorithms to use in creating decision boundaries, which should yield better accuracy in classification. This opens the door for going beyond classification and into the realm of component analysis, a necessary but often complicated area to study[Mat96].

Chapter 3, in full, is a reprint of the material as it appears in ASME Turbo Expo 2017. Allen, Cody; de Oliveira, Maurcio, Holcomb, Chad; ASME 2017. The dissertation author was the primary investigator and author on this paper.

Chapter 4

Gas Fuel Lower Heating Value Prediction Using Hybrid Models

4.1 Introduction

Chapter 4 continues the pursuit of robust diagnostics for monitoring the health of GTs. However, in this chapter we seek to unify physics based models with data driven models by creating hybrid models as discussed in Sections 1.2 and 1.2.1. As an application, we will estimate gas fuel lower heating value (LHV).

Gas fuel comes from myriad of sources, with many molecular composition variations. The energy content, or calorific value of a fuel is expressed in terms of its *heating value*. Generally, in GT combustion, the gas fuel LHV is of higher importance. Informally, the LHV of a gas fuel represents a measure of heat that will occur from complete combustion of the given fuel.

In GT fuel control, a common control method is to implement a calculated calorific rate of gas fuel flow into the combustor [DS18, GG19, HLC14], which requires pressures, temperatures, effective areas and fuel properties such as LHV and specific gravity (SG). With the calorific flow

rate known, the required air mass flow can be calculated to produce the stoichiometric air to fuel ratio in the combustor. The engine controller can control guide vanes as well as the fuel control valve(s) to produce this ratio at various power loads. A change in gas composition usually leads to a change in gas properties, which produces variation in mass flow rates through the combustor. In particular, variation of the enthalpy drop in expansion occurs as well as variation in the flow rate at the turbine inlet, which propagates through to affect turbine/compressor matching [CLM05]. LHV and SG may be combined into a single parameter $WI = \frac{LHV}{\sqrt{SG_{air}}}$, called the *Wobbe Index*, that provides a measure of interchangeability of fuel gases.

The higher a gas's WI, the greater the heating value of a fixed quantity of gas that will flow through an orifice of a given diameter in a given interval of time. It is customary to provide WI without units, even though it has the dimensions [Btu/scf]. The usefulness of the WI is that if two different gas fuel compositions have the same WI, the pressure drop in a given fuel system will be the same for both gases and in general direct substitution is possible where no change to the fuel system control is required. [RXK12]. A general design criterion for WI variation is to require changes in the fuel control system; that is, changes to set points or updates to fuel properties, when the WI changes from the initial value for which the controller is calibrated. Therefore, knowledge of both the LHV and SG are necessary for precise operation of GTs, especially when gas fuel supply varies in molecular composition.

In combustion theory, the *enthalpy of formation* can be defined as the energy released or absorbed when the compound is formed from its elements, where the compound and elements occur at the standard reference temperature and pressure. The enthalpy of formation is generally determined by use of statistical thermodynamics combined with observed spectroscopic data [MSBB10]. These methods are beyond the scope of this chapter, and instead we will be more interested in the *enthalpy of combustion*. The enthalpy of combustion, $\bar{h}_{RP}$, is defined as the difference between the enthalpy of the products and the enthalpy of the reactants, assuming complete combustion is achieved for a given temperature and pressure [MSBB10]; that is,

$\bar{h}_{RP} = \sum_P n_e \bar{h}_e - \sum_R n_i \bar{h}_i$ where the n's correspond to the respective coefficients of the reaction equation giving the moles of reactants and products, per mole of fuel. From this, the *heating value* can be defined as the magnitude of the enthalpy of combustion [Bej88]. The *higher heating value* (HHV) is obtained when all the H_2O formed as a result of combustion is in liquid form whereas the *lower heating value* (LHV) is obtained when all the H_2O formed as a result of combustion takes the form of a gas. The higher heating value exceeds the lower heating value by amount commensurate to the magnitude of energy that would be released were all H_2O in the products condensed into liquid, which is sometimes referred to as the *latent heat* [Sko09]. Note that the energy characterized by the latent heat is not recovered in the combustion process.

In the present situation, we would like to obtain an estimate of LHV. The fuel composition is unknown; therefore, the above calculation which utilizes the moles of reactants is not available. Instead, an approach utilizing energy balances through multiple sections of a gas turbine will be implemented, which yields an equation containing LHV as an explicit variable. Multiple sections are included so the resulting energy balance equation contains only parameters commonly measured on an industrial GT. We will take a *least squares* approach to optimization of LHV estimate. We develop a parametric model of the error produced by the general energy balance equation across many sets of varied fuel data which produces a global error function. This global error function can then be applied to new sets of data where gas fuel properties are unknown, to estimate LHV to high accuracy.

We use a high fidelity physics based simulation to test our method and build the global error function capable of spanning wide ranges of WI fuels. We validate our method using a standard machine learning technique of validation [JWHT13], *k-fold* cross validation. Figure (4.1) is presented below and provides the engine station numbering system we shall use in this presentation as well as the control volumes that will be necessary for our energy balance method.

The method developed in this section is part of a general control scheme for fuel flow in a GT, and is patent pending in the US: US Patent Application number 15852411.

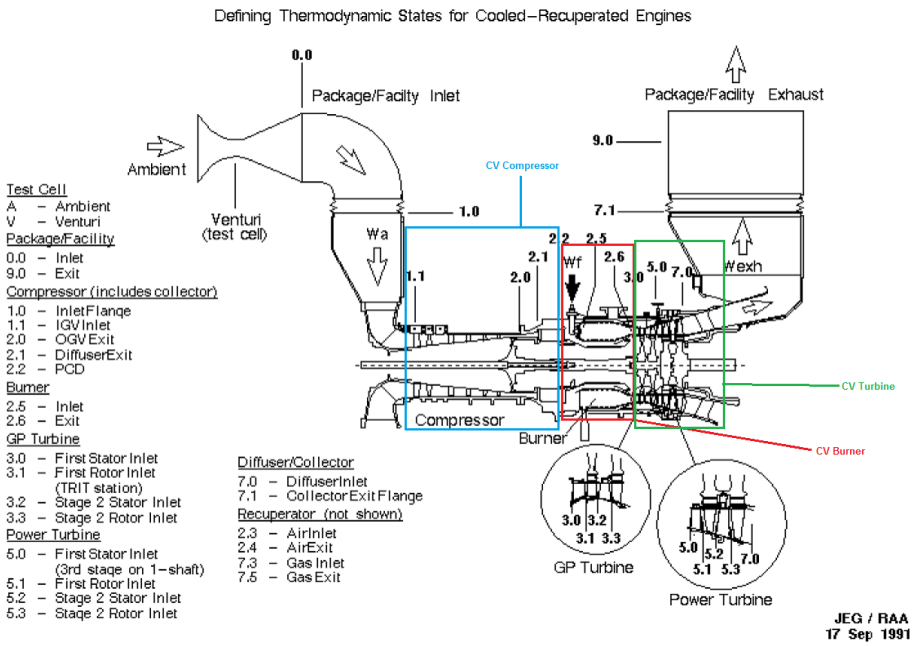


Figure 4.1: Engine station numbering

4.2 Chapter Notation

$\dot{Q}_{cv}$	Net rate of energy transfer by heat across control volume
$\dot{W}_{cv}$	Net rate of energy transfer by work across control volume excluding flow work
KE_{net}	Net kinetic energy
PE_{net}	Net potential energy
$\dot{m}_i$	Total mass flow through inlet i
$\dot{m}_e$	Total mass flow through outlet e
h_i	Total specific enthalpy through inlet i
h_e	Total specific enthalpy through outlet e
T_n	Absolute temperature at station n
$c_{p,n}$	Heat capacity of gas at station n where pressure is constant
$P_{section}$	Power produced or consumed by *section*
$\bar{h}_{RP}$	Enthalpy of combustion

LHV	Lower Heating Value of gas
η_b	Burner efficiency
WI	Wobbe Index
F	Fuel to air ratio in burner
f	LHV estimation error function

4.3 Theoretical Modelling

We start our estimation of LHV by developing a simplified energy balance which will provide us with an explicit formulation of LHV. First let us review the first law of Thermodynamics [MSBB10], applied to a 1D control volume with multiple inlets i and outlets e,

$$\frac{dE_{cv}}{dt} = \dot{Q}_{cv} - \dot{W}_{cv} + \sum_i \dot{m}_i \left(h_i + \frac{V_i^2}{2} + g z_i \right) - \sum_e \dot{m}_e \left(h_e + \frac{V_e^2}{2} + g z_e \right)$$

where $\dot{Q}_{cv}$ is the net rate at which energy is being transferred in by heat transfer at time t, and $\dot{W}_{cv}$ is the net rate at which energy is being transferred out by non-flow work at time t. Then at steady state (or quasi-equilibrium), we find

$$\dot{Q}_{cv} - \dot{W}_{cv} = \sum_e \dot{m}_e \left(h_e + \frac{V_e^2}{2} + g z_e \right) - \sum_i \dot{m}_i \left(h_i + \frac{V_i^2}{2} + g z_i \right) \tag{4.1}$$

Equation (4.1) is the basis of the energy balance equation we will develop in this section. We will apply this to each of the control volumes in our derivation. Note that in the following analyses, the net potential energy, $PE_{net} = \sum_e \dot{m}_e z_e - \sum_i \dot{m}_i z_i \approx 0$ and is dropped from all calculations. Additionally, the net kinetic energy, $KE_{net} = \left(\sum_e \dot{m}_e \frac{V_e^2}{2} - \sum_i \dot{m}_i \frac{V_i^2}{2} \right) << (\sum_e \dot{m}_e h_e - \sum_i \dot{m}_i h_i)$ and is also dropped from all calculations. Therefore, we shall use the approximate energy balance

form in the following derivations

$$\dot{Q}_{cv} - \dot{W}_{cv} = \sum_e \dot{m}_e h_e - \sum_i \dot{m}_i h_i \tag{4.2}$$

It will be fruitful to briefly review enthalpy calculation procedures. Recall, for an ideal gas with constant composition, enthalpy is only a function of temperature[Sko09], where $\frac{dh}{dT} = C_p(T)$ or $dh = C_p(T)dT$. Integration provides

$$h(T_2) = h(T_1) + \int_{T_1}^{T_2} C_p(T)dT = h_{ref} + \int_{T_{ref}}^{T_2} C_p(T)dT$$

where $h(T_1) = h_{ref} = 199.97[kJ/kg]$ and $T_{ref} = 200K$. Since integration is a linear operator, we find by superposition [Apo74], that this is an equivalent representation of finding the enthalpy change between two arbitrary stations m and n:

$$\begin{aligned}\int_{T_m}^{T_n} C_p(T)dT &= h(T_n) - h(T_m) = h(T_n) + (-h_{ref} + h_{ref}) - h(T_m) \\ &= (h(T_n) - h_{ref}) - (h(T_m) - h_{ref}) \\ &= \int_{T_{ref}}^{T_n} C_p(T)dT - \int_{T_{ref}}^{T_m} C_p(T)dT\end{aligned}$$

For each station prior to the burner, we have airflow only; hence, our specific heat function is that of air. We use the following approximate function for modeling the specific heat of air [MSBB10]

$$C_p(T) = \left(\beta_0 + \beta_1 T + \beta_2 T^2 + \beta_3 T^3 + \beta_4 T^4\right) \frac{\bar{R}}{M_{air}}$$

where T is in K and $C_p(T)$ has units $[kJ/kg]$.

To begin our energy balance derivations we first reference Figure 4.1 from which we obtain our station numbers. We have established three control volumes that partition the engine

Table 4.1: List of coefficients for specific heat function $C_p(T)$.

Coefficient	Value
β_0	3.653
β_1	$-1.337 \cdot 10^{-3}$
β_2	$3.294 \cdot 10^{-6}$
β_3	$-1.913 \cdot 10^{-9}$
β_4	$0.2763 \cdot 10^{-12}$

flow paths. Based on the control volumes, we have the following relations for mass flows

$$\dot{m}_{7.0} = \dot{m}_{3.0} + \dot{m}_b \qquad \dot{m}_{3.0} = \dot{m}_{2.0} + \dot{m}_f \qquad \dot{m}_{1.0} = \dot{m}_{2.0} + \dot{m}_b \tag{4.3}$$

Consider *CV Burner* and write down the energy balance for this control volume. Note that there is no work other than flow work for this control volume, therefore $\dot{W}_{2.4,3} = 0$ and $\dot{Q}_{burn} = (\dot{m}_{2.4} + \dot{m}_f)h_{3.0} - \dot{m}_{2.4}h_{2.4}$. The additional heat generated in the burner is a result of the combustion process. The total combustion energy is partially a function of the supplied fuel LHV, where the LHV is defined as the amount of heat released by combusting a specified quantity (initially at $25°C$) of a fuel and returning the temperature of the combustion products to $150°C$, which assumes the latent heat of vaporization of water in the reaction products is not recovered. Therefore, the theoretical heat generated in the burner can be calculated as $\dot{Q}_{burn} = \dot{m}_f LHV$.

Using these two quantities, we can construct burner efficiency. For non-recuperated engines, we let $h_{2.4} = h_{2.0} = h_2$. Burner efficiency is then given by:

$$\begin{aligned} \eta_b = \frac{\dot{Q}_{actual}}{\dot{Q}_{theoretical}} &= \frac{(\dot{m}_{2.4} + \dot{m}_f)h_{3.0} - \dot{m}_{2.4}h_{2.4}}{\dot{m}_f LHV} = \frac{\dot{m}_2((1 + \frac{\dot{m}_f}{\dot{m}_2})h_3 - h_2)}{\dot{m}_f LHV} \\ &= \frac{\left(1 + \frac{\dot{m}_f}{\dot{m}_2}\right) h_3 - h_2}{\frac{\dot{m}_f}{\dot{m}_2} LHV} \\ &= \frac{(1+F)\, h_3 - h_2}{\frac{\dot{m}_f}{\dot{m}_2} LHV} \end{aligned} \tag{4.4}$$

Where $F = \frac{\dot{m}_f}{\dot{m}_2}$ is the fuel to air ratio the burner sees. We will return to equation (4.4) shortly. We now consider the control volume CV Turbine. The turbine produces work as a result of the gas flowing through it. Using an energy balance, we have

$$-\dot{W}_{turb} \approx \dot{m}_{7.0}h_{7.0} - (\dot{m}_{3.0}h_{3.0} + \dot{m}_b h_b) = \dot{m}_{3.0}\left(\frac{\dot{m}_{7.0}}{\dot{m}_{3.0}}h_{7.0} - h_{3.0} - \frac{\dot{m}_b}{\dot{m}_{3.0}}h_b\right)$$

$$\Leftrightarrow$$

$$h_3 \approx \frac{\dot{m}_{7.0}}{\dot{m}_{3.0}}h_{7.0} - \frac{\dot{m}_b}{\dot{m}_{3.0}}h_b + \frac{P_{turb}}{\dot{m}_{3.0}} \tag{4.5}$$

where $\dot{W}_{turb} = P_{turb}$ is total power generated by the turbine. Now consider the control volume around the compressor, CV Compressor. The compressor consumes work as gas is sucked and compressed through it. Using an energy balance and noting that , we have

$$\dot{W}_{comp} = \dot{m}_2 h_2 + \dot{m}_b h_b - \dot{m}_1 h_1$$

where m_b is the mass flow that exits the compressor as bleed flow and h_b is the resulting enthalpy of this flow. Then,

$$\dot{W}_{comp} \approx \dot{m}_2 h_2 + \dot{m}_b h_b - \dot{m}_1 h_1 \qquad \text{with,} \qquad h_2 \approx \frac{P_{comp}}{\dot{m}_{2.0}} - \frac{\dot{m}_b}{\dot{m}_{2.0}}h_b + \frac{\dot{m}_1}{\dot{m}_{2.0}}h_1 \tag{4.6}$$

Using equations (4.5) and (4.6) and the mass flows found in equations (4.3), we have

$$\begin{aligned}(1+F)h_{3.0} - h_{2.0} &\approx (1+F)\left(\frac{\dot{m}_{7.0}}{\dot{m}_{3.0}}h_{7.0} - \frac{\dot{m}_b}{\dot{m}_{3.0}}h_b + \frac{P_{turb}}{\dot{m}_{3.0}}\right) - \left(\frac{P_{comp}}{\dot{m}_{2.0}} - \frac{\dot{m}_b}{\dot{m}_{2.0}}h_b + \frac{\dot{m}_1}{\dot{m}_{2.0}}h_1\right) \\ &= (1+F)\left(\frac{(\dot{m}_{3.0}+\dot{m}_b)h_{7.1} - \dot{m}_b h_b}{\dot{m}_{3.0}} + \frac{P_{turb}}{\dot{m}_{2.0}+\dot{m}_f}\right) - \left(\frac{P_{comp} - \dot{m}_b h_b + (\dot{m}_{2.0}+\dot{m}_b)h_1}{\dot{m}_{2.0}}\right) \\ &= E_P + E_E - E_A + E_b \end{aligned} \tag{4.7}$$

where,

$$
\begin{aligned}
E_P &= (1+F)\frac{P_{turb}}{\dot{m}_{3.0}} - \frac{P_{comp}}{\dot{m}_{2.0}} = \left(1+\frac{\dot{m}_f}{\dot{m}_2}\right)\frac{P_{turb}}{\dot{m}_{2.0}+\dot{m}_f} - \frac{P_{comp}}{\dot{m}_{2.0}} = \frac{P_{turb}-P_{comp}}{\dot{m}_{2.0}} = \frac{P_{shaft}}{\dot{m}_{2.0}} \\
E_E &= (1+F)\frac{\dot{m}_{3.0}+\dot{m}_b}{\dot{m}_{3.0}}h_{7.1} = (1+F)h_{7.1} + (1+F)\frac{\dot{m}_b}{\dot{m}_{3.0}}h_{7.1} \\
&= (1+F)h_{7.1} + \left(1+\frac{\dot{m}_f}{\dot{m}_2}\right)\frac{\dot{m}_b}{\dot{m}_{2.0}+\dot{m}_f}h_{7.1} = (1+F)h_{7.1} + \frac{\dot{m}_b}{\dot{m}_{2.0}}h_{7.1} \\
E_A &= \frac{\dot{m}_{2.0}+\dot{m}_b}{\dot{m}_{2.0}}h_1 = h_1 + \frac{\dot{m}_b}{\dot{m}_{2.0}}h_1 \\
E_b &= \frac{\dot{m}_b}{\dot{m}_{2.0}}h_b - (1+F)\frac{\dot{m}_b}{\dot{m}_{3.0}}h_b = \frac{\dot{m}_b}{\dot{m}_{2.0}}h_b - \left(1+\frac{\dot{m}_f}{\dot{m}_2}\right)\frac{\dot{m}_b}{\dot{m}_{2.0}+\dot{m}_f}h_b = \frac{\dot{m}_b}{\dot{m}_{2.0}}h_b - \frac{\dot{m}_b}{\dot{m}_{2.0}}h_b = 0
\end{aligned}
$$

and it follows that,

$$
\begin{aligned}
(1+F)h_{3.0} - h_{2.0} \approx E_P + E_E - E_A + E_b &= \frac{P_{shaft}}{\dot{m}_{2.0}} + (1+F)h_{7.1} + \frac{\dot{m}_b}{\dot{m}_{2.0}}h_{7.1} - \left(h_1 + \frac{\dot{m}_b}{\dot{m}_{2.0}}h_1\right) \\
&= \frac{P_{shaft}}{\dot{m}_{2.0}} + (1+F)h_{7.1} - h_1 + \frac{\dot{m}_b}{\dot{m}_{2.0}}(h_{7.1}-h_1) \qquad (4.8)
\end{aligned}
$$

Plugging equation (4.8) into equation (4.4) we have

$$
LHV = \frac{\left(1+\frac{\dot{m}_f}{\dot{m}_2}\right)h_3 - h_2}{\frac{\dot{m}_f}{\dot{m}_2}\eta_b} \approx \frac{\frac{P_{shaft}}{\dot{m}_{2.0}} + (1+\frac{\dot{m}_f}{\dot{m}_2})h_{7.1} - h_1 + \frac{\dot{m}_b}{\dot{m}_{2.0}}(h_{7.1}-h_1)}{\frac{\dot{m}_f}{\dot{m}_2}\eta_b} \qquad (4.9)
$$

as desired.

4.4 A Method for Estimating Lower Heating Value

This section discusses the energy balance equation that is the basis of our LHV estimation. We state the balance equation across the burner which can be found in most gas turbine related

technical books [Mat96][Kia02]

$$\eta_b = \frac{\dot{Q}_{actual}}{\dot{Q}_{theoretical}} = \frac{(\dot{m}_{2.4} + \dot{m}_f)h_{3.0} - \dot{m}_{2.4}h_{2.4}}{\dot{m}_f LHV} \tag{4.10}$$

From this equation, we can make use of control volumes through the compressor and turbine and apply energy balances to work out the approximation:

$$\frac{(\dot{m}_{2.4} + \dot{m}_f)h_{3.0} - \dot{m}_{2.4}h_{2.4}}{\dot{m}_f LHV} \approx \frac{\frac{P_{shaft}}{\dot{m}_{2.0}} + (1 + \frac{\dot{m}_f}{\dot{m}_2})h_{7.1} - h_1 + \frac{\dot{m}_b}{\dot{m}_{2.0}}(h_{7.1} - h_1)}{\frac{\dot{m}_f}{\dot{m}_2} LHV} \tag{4.11}$$

In equation (4.11) we are using the convention $P_{turbine} = P_{shaft} + P_{compressor} + P_{losses}$, where $P_{losses} << P_{shaft} + P_{compressor}$ and are dropped in the approximation. Written another way, we have

$$LHV = \frac{\frac{P_{shaft}}{\dot{m}_{2.0}} + (1 + \frac{\dot{m}_f}{\dot{m}_2})h_{7.1} - h_1 + \frac{\dot{m}_b}{\dot{m}_{2.0}}(h_{7.1} - h_1)}{\frac{\dot{m}_f}{\dot{m}_2}\eta_b} \tag{4.12}$$

Equation (4.12) yields a direct solution to solving for the LHV of the fuel. However, there are two unknowns in (4.12), η_b and LHV. With only one equation, and the goal being to estimate LHV, we assume a reasonable value of η_b and proceed.

With an assumption of burner efficiency and a set of N data points, we have an over-determined system. However, to find the optimal LHV given our set of data, we may perform a least squares fit. Define the following quantities:

$$x = LHV \qquad A = \frac{\dot{m}_f}{\dot{m}_{2.4}}\eta_b \qquad b = \frac{P_{shaft}}{\dot{m}_{2.4}} + \left(1 + \frac{\dot{m}_f}{\dot{m}_{2.4}}\right) h_{7.1} - h_1 + \frac{\dot{m}_b}{\dot{m}_{2.4}}(h_{7.1} - h_1) \tag{4.13}$$

The problem of solving for LHV in (4.13) takes on the familiar form $Ax = b$, which has a least

squares solution[Str88].[1]

$$x = (A^T A)^{-1} A^T b \tag{4.14}$$

so that $x = LHV_o$, the optimal LHV, in equation (4.14). While this does yield the optimal LHV in a least squares sense, we very well may have an error percentage that is unfavorable, where the error percent is given as

$$e_{LS} = 100 \left(\frac{LHV_o - LHV_{real}}{LHV_{real}} \right) \tag{4.15}$$

This is due to the approximations that were made in deriving equation (4.11). A natural question is, can we make this estimate better without introducing the complexities that were approximated? The next subsection provides a solution to this new problem in the form of modeling the error function.

4.4.1 Error Modeling and More Accurate Estimates of LHV

In the previous section, we were able to write down an energy balance that provided the base calculation for LHV. We then cast the problem as a least squares problem when given $N > 1$ sets of data points and solved for the optimal LHV value, LHV_o. We last checked the percent error, e_{LS} and noted that e_{LS} may be larger than acceptable for engineering purposes. In this section, we describe a method for reducing e_{LS} without having to alter the approximations that were made to derive equation (4.11).

We know that in solving $A\boldsymbol{x} = \boldsymbol{b}$, we found the optimal $\boldsymbol{x}$ in the least squares sense, which means $||A\boldsymbol{x} - \boldsymbol{b}||_2 = \boldsymbol{e}_{LS}$ has been minimized. Therefore, we return to equation (4.12) where the real LHV value is used to see what can be done about errors introduced from our approximation of the thermodynamics.

[1] If the columns of A are independent, $A^T A$ is invertible

Consider the scalar case, equation (4.12), then for each set j of data points. Let,

$$E_j = \left(\frac{P_{shaft}}{\dot{m}_{2.4}} + \left(1 + \frac{\dot{m}_f}{\dot{m}_{2.4}}\right) h_{7.1} - h_1 + \frac{\dot{m}_b}{\dot{m}_{2.4}} \left(h_{7.1} - h_1\right) \right)_j$$

$$|e_j| = \left| \left(LHV_{real} \frac{\dot{m}_f}{\dot{m}_{2.4}} \right)_j \eta_b - E_j \right| \geq 0 \tag{4.16}$$

where e_j is the error introduced from our approximate modeling for data set j and not the same as e_{LS}. Moving back into matrix form and rearranging equation (4.16), we find

$$A\boldsymbol{x} = \boldsymbol{b} - \boldsymbol{e} \tag{4.17}$$

and from the above discussion, we wish to find a function $f(\cdot) = e$. We choose a second order polynomial function due to the shape of modeling errors (shown in Figure 4.3).

$$e_j = f(T_1(j), P_{shaft}(j)) = \alpha_1 T_1(j) + \alpha_2 T_1^2(j) + \alpha_3 P_{shaft}(j) + \alpha_4 P_{shaft}^2(j) + \alpha_5 T_1(j) P_{shaft}(j) \tag{4.18}$$

Now that we have defined f, we need only fit the parameters α_i. This can be done easily using any scripting language such as Matlab or Python, and could even be done in Excel with enough perseverance. The essential point is that the data used to fit the parameters, that is, the " training data", must span a wide range of LHV values, and to be safe, a wide range of SG values. This can be summed up by spanning a wide range of Wobbe Index values. Once this data is collected, estimating each α_i is a simple least squares estimate.

To summarize, the process of fitting f should go as follows:

Algorithm 1 (Solution Algorithm)**.** *1. Create fuel compositions either by hand or using Solar's GFS tool*

2. *Import the set of fuels into a high fidelity simulation such as Solar's performance simulator and for each specified fuel, run the simulation and save the results*

3. *Using the true LHV (specified by the fuel), create equation (4.16) in matrix form*

4. *Concatenate all error vectors into a single error vector*

5. *Build matrix A:* $A = \begin{bmatrix} T_1 & T_1^2 & P_{shaft} & P_{shaft}^2 & T_1 P_{shaft} \end{bmatrix}$

6. *Solve the* $Ax = b$ *problem for* $x^* = LHV_o$

4.5 Model Corrections

4.5.1 Estimating LHV from a Single Set of High-Fidelity Data

For this first example, we will have a gas composition that yields $LHV_{real} = 18827$ and $SG = 0.6748$. Our goal is to estimate LHV to highest degree of accuracy. Our engine is the Taurus 60, rating 7901, conventional combustion. *If SoLoNOx engines are used, this analysis will only work at full load when the bleed valve is closed.* Note that Solar's performance simulation data needs to be converted to correct units before modeling can occur; we will show both English Engineering units as well as SI units.

To get correct values of $W_{2.4}$, we rely on bleed flow percentages developed and maintained by Solar's Engine Performance group. Then, we have $W_{2.4} = WC2 = (1 - bleed\%)WC$. When calculating the fuel to air ratio, one must use $FAR = WF/WC2$ to get the true fuel to air ratio the burner sees.

After performing the calculations above, and using equation (4.11), we can calculate the point wise LHV values shown in Figures (4.2) and (4.4).

We see that while the error is not atrocious, it is still beyond what we would normally want to see in an engineering calculation. This motivates the least squares calculation discussed

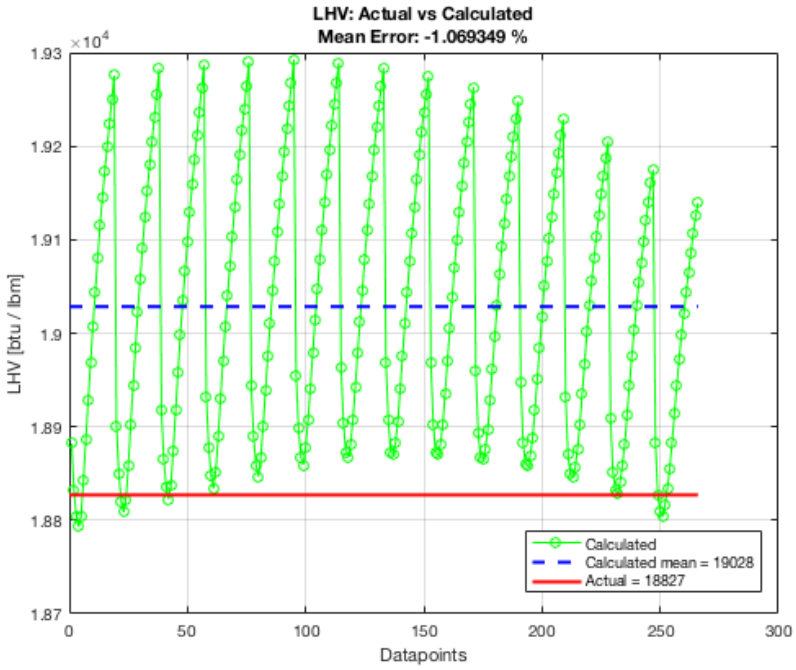


Figure 4.2: Example point wise LHV calculation from energy balance equation. Actual LHV = 18827 [btu/lbm], point wise mean value LHV = 19028 [btu/lbm].

above. The result of this calculation is seen in Figure 4.5 where the energy balance error resulting from our least squares fit is shown as a function of sequential data points.

From Figure 4.5, we see that we have approximately 1.5% error. However, we still have not found LHV to satisfactory accuracy. Therefore, we use our error model to remove the error present in our calculation. *In reviewing Figure 4.7, we find that error in our LHV calculation has dropped considerably, from 1.5% to -0.0767% error!!*

Now, the reader may be thinking, "Of course the error is gone, the model is over-fit to the data ... etc," since it was built on this specific data set. To that we respond, you are absolutely right. It is not until we train f across multiple data sets with varying fuel compositions and then apply it to a fuel it has not seen that we will really determine how valuable this approach is. We do this in the next section.

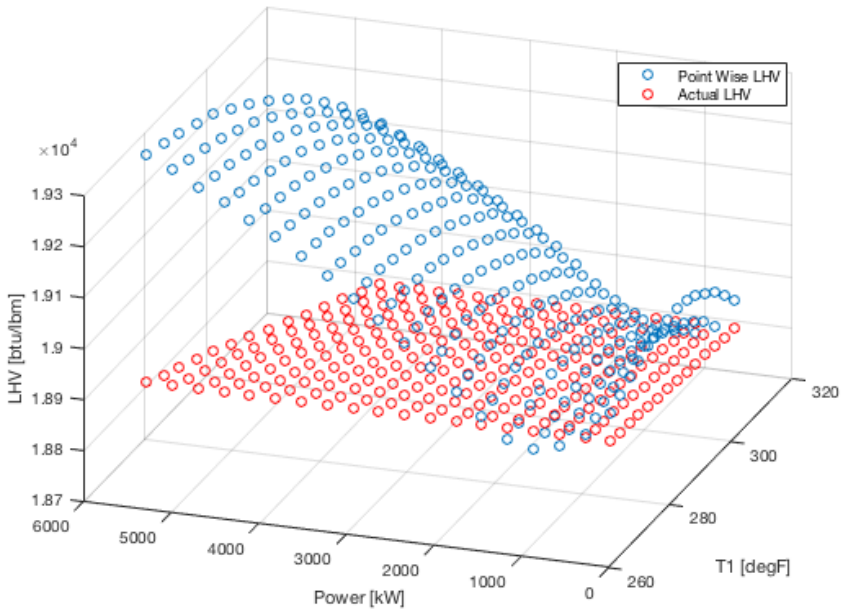


Figure 4.3: Example point wise LHV calculation from energy balance equation. Actual LHV = 18827 [btu/lbm], point wise mean value LHV = 19028 [btu/lbm]. Note the slight curvature in both the T1 and Power coordinates. This led to the choice of a quadratic function f in these variables to model the error.

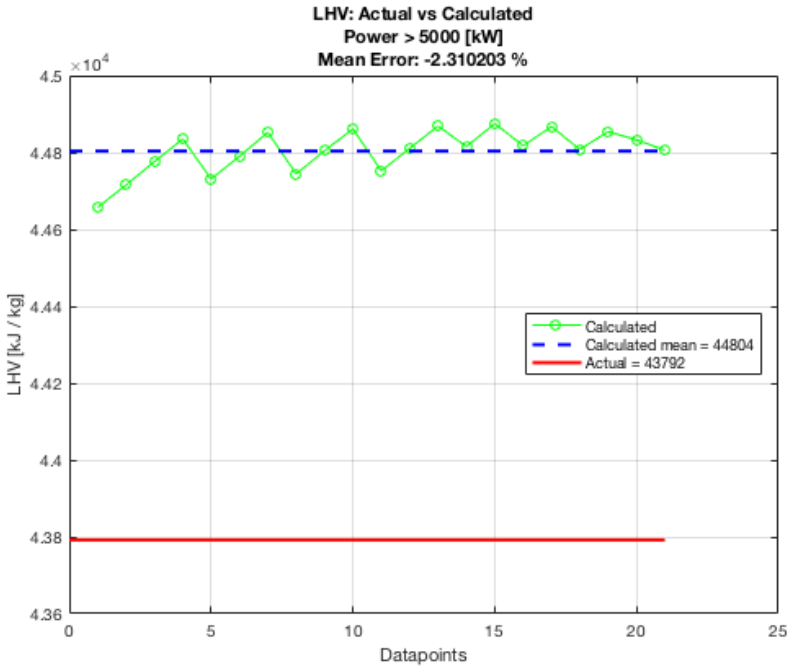


Figure 4.4: ,
point]Example point wise LHV calculation from energy balance equation using only 90% - 100% load data points. Actual LHV = 43792 [kJ/kg], point wise mean value LHV = 44804 [kJ/kg].

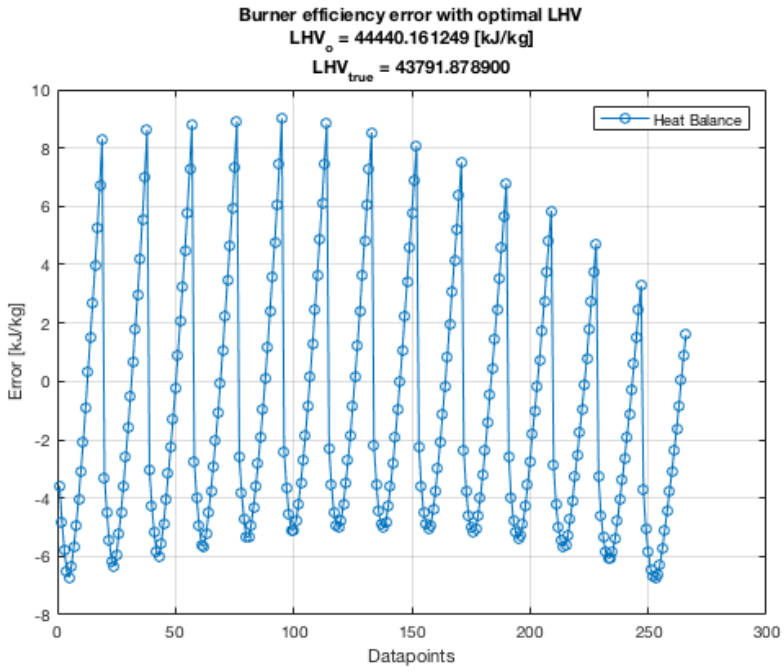


Figure 4.5: Energy balance equation error using LHV_o (see eq. 4.16). Actual LHV = 43792 [kJ/kg], least squares LHV = 44440 [kJ/kg], 1.48% error.

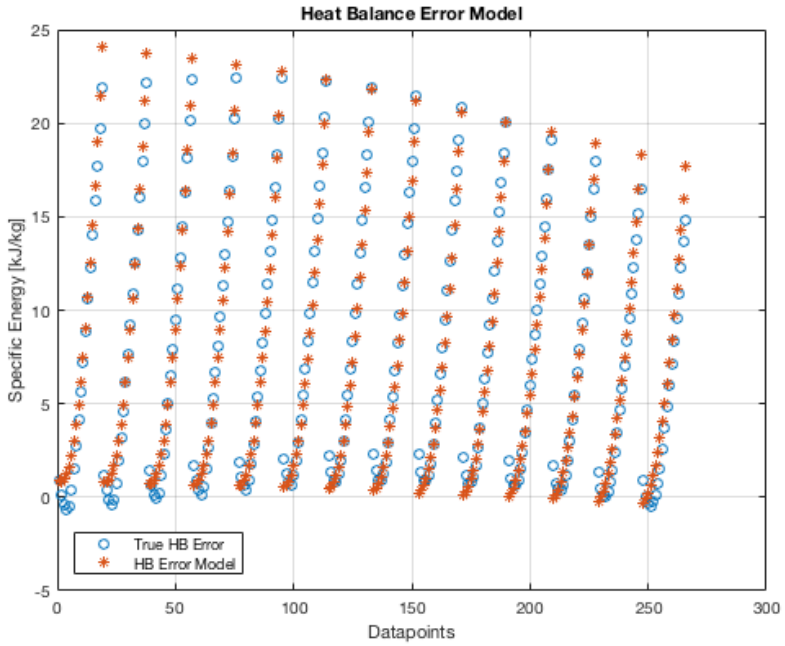


Figure 4.6: Energy balance real equation error and f model of error.

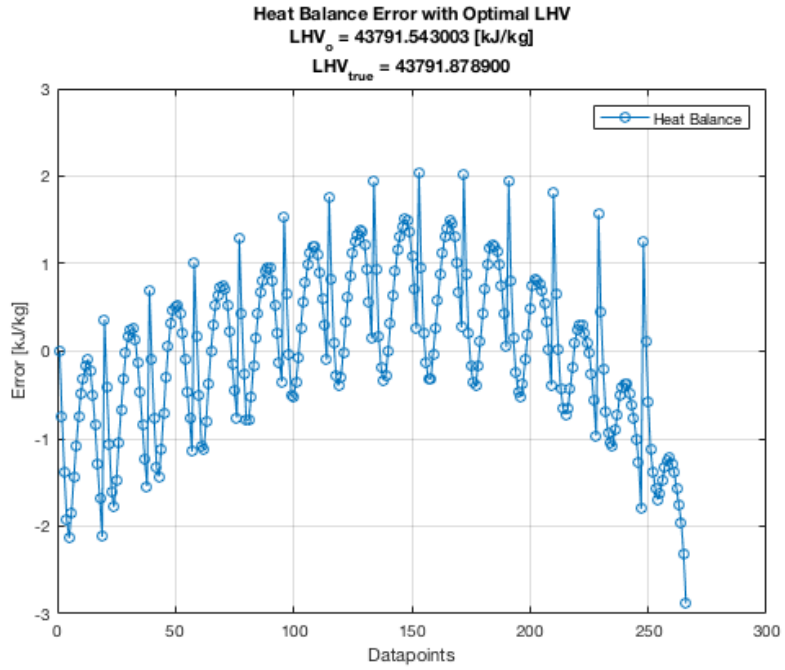


Figure 4.7: ,
least squares LHV = 43791.54]Energy balance equation error using LHV_o after f has been applied. Actual LHV = 43791.88 [kJ/kg], least squares LHV = 43791.54 [kJ/kg], -0.0013% error.

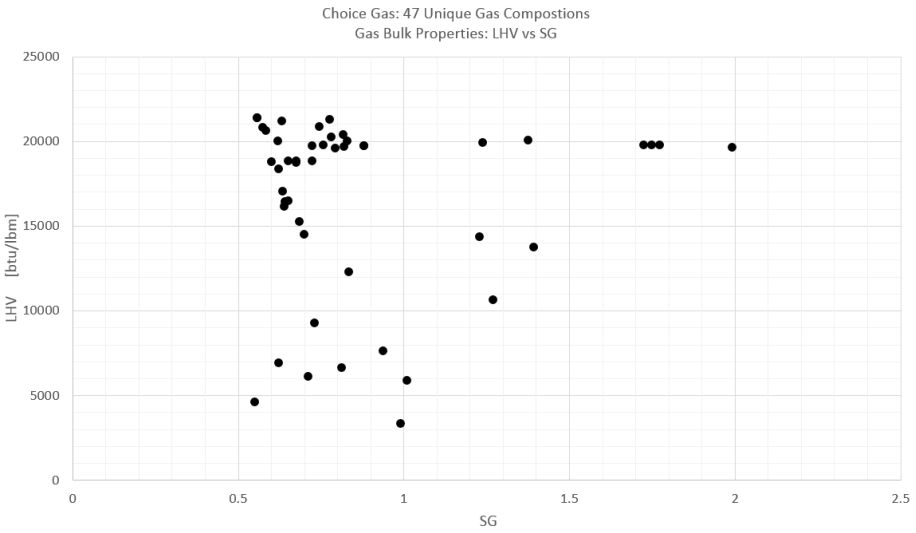


Figure 4.8: Selected fuels bulk properties, LHV and SG.

4.5.2 Fitting f Across Varying Fuels

We wish to build a global function f which can account for many variations in fuel compositions, and specifically, variations in (LHV, SG) pairs. Recall, in Section 4.4.1, we defined the function f in equation (4.18), which was

$$f(T_1, P_{shaft}) = \alpha_1 T_1 + \alpha_2 T_1^2 + \alpha_3 P_{shaft} + \alpha_4 P_{shaft}^2 + \alpha_5 T_1 P_{shaft}$$

and gave an algorithm for solving for the coefficients in Algorithm 2.

We now see why a quadratic function was chosen, since the errors appear to have slight quadratic characteristics as seen in Figure (4.3). This characteristic is persistent across varying fuels. We have assembled 47 different fuel compositions and run the corresponding Solar performance simulation with the procedure outlined at the end of Section 4.4.1 with conditions specified at the beginning of this section. Figures 4.8 and 4.9 present the fuel data, showing SG, LHV and WI for each individual fuel composition.

The process of finding gas fuel compositions is a tedious one. In order to get realistic compositions, one must mine through Solar's GFS tool. There is a search feature where one can

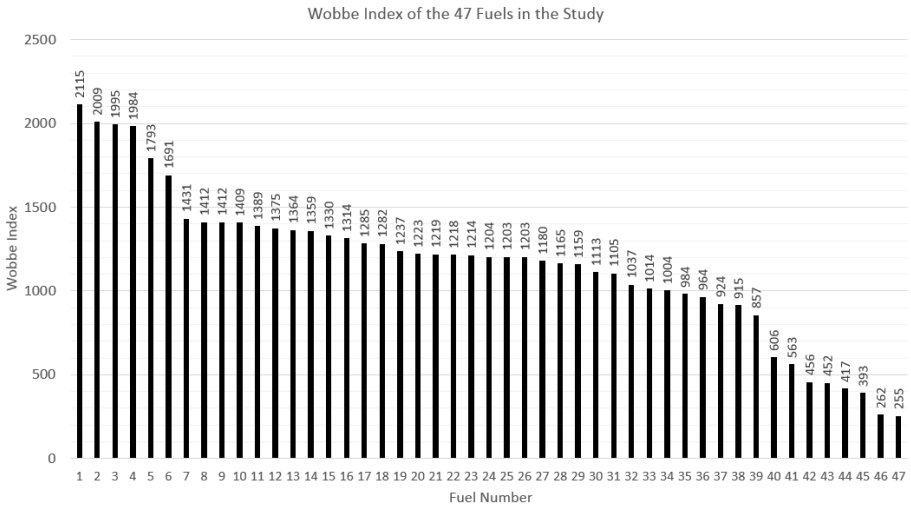


Figure 4.9: Selected fuels WI.

search for "Taurus 60" engines, but from there, must select each study separately to see the fuel specifications. In order to build the data set for f, we went through GFS studies from 5/1/17 to 8/1/17 and tried to select fuels that differed in both LHV and SG. *No attempt was made to force variation of unique molecules present in the gas composition, nor was there any attempt to limit gas compositions to non-compound molecular constituents.* Once the GFS study numbers are known, one can load these into the Solar performance simulation to specify the fuel composition and then run the simulation with the conditions given. At this point, the simulation results can be saved as .csv file format to be brought into a computing environment. We continue with the process in Section 4.4.1 to finish fitting f to the large data set.

In determining the effectiveness of our method defined in Section 4.4, we will use a global f and test it against various sets of hold out data. This process is called *cross validation*, which we briefly explain below.

We are using k-fold cross validation [JWHT13] here. We select 43 of the 47 data sets to train f, then for the remaining 4 data sets, we calculate LHV_o using f by combining results from equations (4.14) and (4.17). Therefore, for a given iteration we have,

$$LHV\frac{\dot{m}_f}{\dot{m}_{2.4}}\eta_b = \frac{P_{shaft}}{\dot{m}_{2.4}} + \left(1+\frac{\dot{m}_f}{\dot{m}_{2.4}}\right)h_{7.1} - h_1 + \frac{\dot{m}_b}{\dot{m}_{2.4}}(h_{7.1}-h_1) - f \tag{4.19}$$

with the following identifications, from equations (4.13), we find the update as

$$\bar{b} = \frac{P_{shaft}}{\dot{m}_{2.4}} + \left(1+\frac{\dot{m}_f}{\dot{m}_{2.4}}\right)h_{7.1} - ch_1 + \frac{\dot{m}_b}{\dot{m}_{2.4}}(h_{7.1}-h_1) - f$$

and then

$$LHV_o = (A^T A)^{-1}A^T\bar{b} \tag{4.20}$$

We then continue this process $k-1$ more times, each time choosing a new set of 4 hold out data sets that have not previously been used. We continue this process until all engines have been used as hold out data.

The median error in performing cross validation is -0.276% which is remarkably better than using equation (4.13) by itself.

4.6 Conclusion

In this chapter we applied a hybrid approach to the application of physics based models and purely data-driven models. We started with first principles to derive a physics based model. We then gathered a large range of WI fuels along with their LHV and SG values and used these fuels to produce an error set based on the physics only model. We then developed an error function that we fit to the error from the physics only model. Adding the error model to the physics only model gave us the hybrid model which produced superior accuracy.

The hybrid model cross validation accuracy is excellent, and shows that we can estimate

LHV from high fidelity physics simulation data for the selected engines. We note that nothing has stopped us from creating two global functions, f_1 and f_2, one for high Wobbe fuels and one for low Wobbe fuels. In doing so, the errors get smaller still, but to the small error already observed, we accept a single function, f. Further research will aim to test this method out on actual field data, with varied WI fuels.

Chapter 4, in full, has been written based on patent application number 15852411. The dissertation author was the primary investigator and author on this patent application.

Chapter 5

Axial Compressor Fouling and its Effect on Gas Turbine Fuel Consumption and Emissions

5.1 Introduction

Chapter 5 continues the hybrid modeling approach. In this chapter we develop an equation that accurately models fuel flow based on compressor performance parameters. The motivation for this chapter stems from the discussion in Section 1.2.2 as well as part of the problem defined in Section 1.3.1.

Our primary goal is to generally quantify the incremental difference in fuel flow attributed to compressor fouling. We additionally show how this can be used in optimizing compressor wash schedules and used to estimate bounds on the emissions associated with running in a fouled state. Our results provide operators of gas turbines with a set of equations that can be used to accurately calculate additional fuel consumption due to fouling but also predict the costs associated with

running in fouled states as it pertains to fuel and emissions.

Section 5.3 provides a literature review of how axial compressor fouling has been applied in simulations and how these applications compare with measurements from fouling field data. Section 5.4 contains a derivation of a physics based equation to calculate the increase in fuel consumption dependent on the compressor performance parameters: air mass flow ($\dot{m}_a$), isentropic efficiency (η_c) and pressure ratio (P_{rc}). By subtracting the nominal fuel flow for the given state of the GT and surrounding, we are able to quantify the additional fuel flow resulting from running the GT in its fouled state. In Section 5.5 we validate our equation using test cell data from Solar Turbines and a gas turbine simulation using the Numerical Propulsion System Simulation (NPSS) software. Section 5.6 applies the derived equation to optimizing compressor wash programs and predicting increases in CO_2 emissions.

5.2 Chapter Notation

Z_c	Zaba site scaling factor
$\dot{V}$	Compressor volume flow
η_p	Compressor polytropic efficiency
η_c	Compressor isentropic efficiency
η_t	Turbine isentropic efficiency
η_b	Combustor efficiency
P_a	Ambient Pressure
P_i	Stagnation pressure at station i
T_i	Stagnation temperature at station i
ΔT_{ij}	$T_i - T_j$
δ_x	Aerodynamic scaling factor for component x
$\hat{c}_{x(\cdot,\cdot)}$	Estimated specific heat (constant x) over temperature range $(\cdot,\cdot)$

R Gas specific gas constant

$\hat{\gamma}_x$ Specific heat ratio for section x from $\hat{c}_{x(\cdot,\cdot)}$

$\dot{m}_f$ Mass flow of fuel

$\hat{\dot{m}}_f$ Estimated Mass flow of fuel

K Nondimensional flow correction coefficient

Q_R Lower Heating Value of Fuel

$\dot{m}_a$ Mass flow of air

P_{rx} Pressure ratio across station x

$\dot{W}_{sh}$ Shaft Output Power

Ω Reduced order set of nonlinear equations

α_i Compressor performance parameter i

x^* Value of parameter x at nominal condition

ζ Scaling factor

n_x Number of moles of a specified substance x

M_x Molar mass of substance x

m_x Mass of substance x

K_p Chemical Equilibrium constant for gaseous mixture

$ma(24)$ Simple 24 hour moving average

5.3 Review of Fouling Relationships

It is clear that compressor fouling leads to overall loss in performance. For the industrial gas turbine, this can be of large consequence to revenue as power generation applications often rely on nominal performance to meet their production needs [HTSS11]. As mentioned in the introduction, additional fuel flow can be supplied to the combustor in an attempt to regain lost shaft power at the expense of higher turbine inlet temperatures. However, this can only occur up

to a limit, as additional heat beyond the material limits would reduce the remaining useful life of hot section components [GAF+18].

A literature review conducted by the authors led to some insight in how degradation is generally imposed by researchers and the effect it typically has. Often, to simulate degradation, multiplicative constants are applied to typical compressor map parameters to scale the compressor map [MHFW89, KBW09, MHBS13, K+18, BK18]. However, one must be careful in how these factors are applied as physical laws must be maintained for a simulation's results to be valid with reality. To ensure this, we have also included data from field studies [BK18, Zab80, IPF+14] as well as field data from Solar Turbines. In the papers we reviewed, most applications of degradation have been in line with actual field data. We present results compiled from a number of published papers as well as data from a Solar Turbines' engine in Figures 5.1 and 5.3. We first introduce the following notation and definition:

Definition 1 (Compressor Performance Parameters)**.** *Define η_c^*, $\dot{m}_a^*$ and P_{rc}^* to be the nominal compressor performance parameter values at full load. Then, the compressor performance parameters are*

$$\alpha_1 = \dot{m}_a/\dot{m}_a^* \qquad \alpha_2 = P_{rc}/P_{rc}^* \qquad \alpha_3 = \eta_c/\eta_c^*$$

As an example, applying a 1% reduction in compressor pressure ratio yields,

$$P_{rc} = P_{rc}^* - 0.01P_{rc}^* = 0.99P_{rc}^* = \alpha_2 P_{rc}^* \qquad \text{and} \qquad \alpha_2 = 0.99 = P_{rc}/P_{rc}^*$$

The data shown in Figure 5.1 indicates that the relationship between reduction in compressor air mass flow percent and reduction in compressor pressure ratio percent is approximately 1:1.

The data shown in Figure 5.2 has wider variation than Figure 5.1. However, there is strong agreement between the two field data sources, namely Zaba et. al. [Zab80] and Igie et. al. [IPF+14]. From the field data and simulations, we conclude that the relationship between

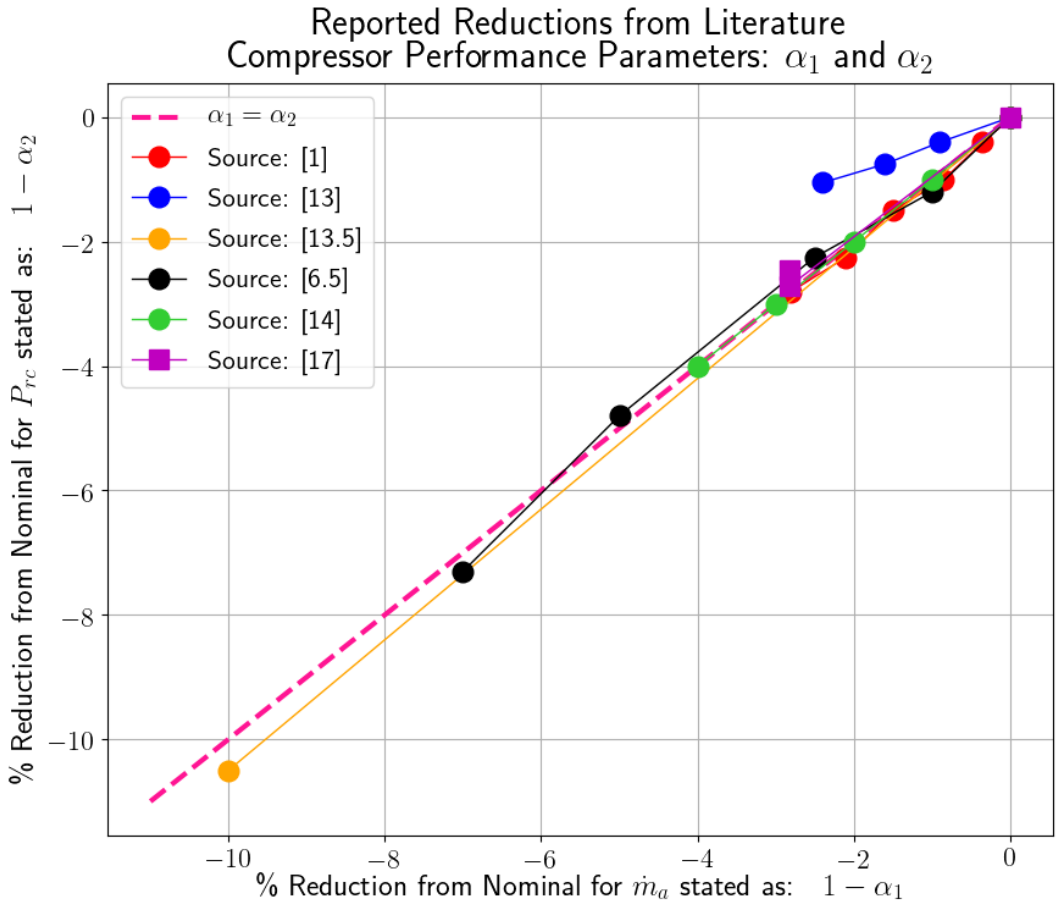


Figure 5.1: Relationship of multiplicative constants applied to compressor mass flow of air and compressor pressure ratio. Note that square markers indicate actual measured data, whereas circle markers are values from simulations. Sources with $X.5$ indicate when a source had two different simulations.

Table 5.1: Figures 5.1, 5.2 and 5.1 References

[1]	[MHFW89]
[6][6.5]	[MHBS13]
[13][13.5]	[KBW09]
[14]	[K$^+$18]
[15]	[BK18]
[17]	[IPF$^+$14]

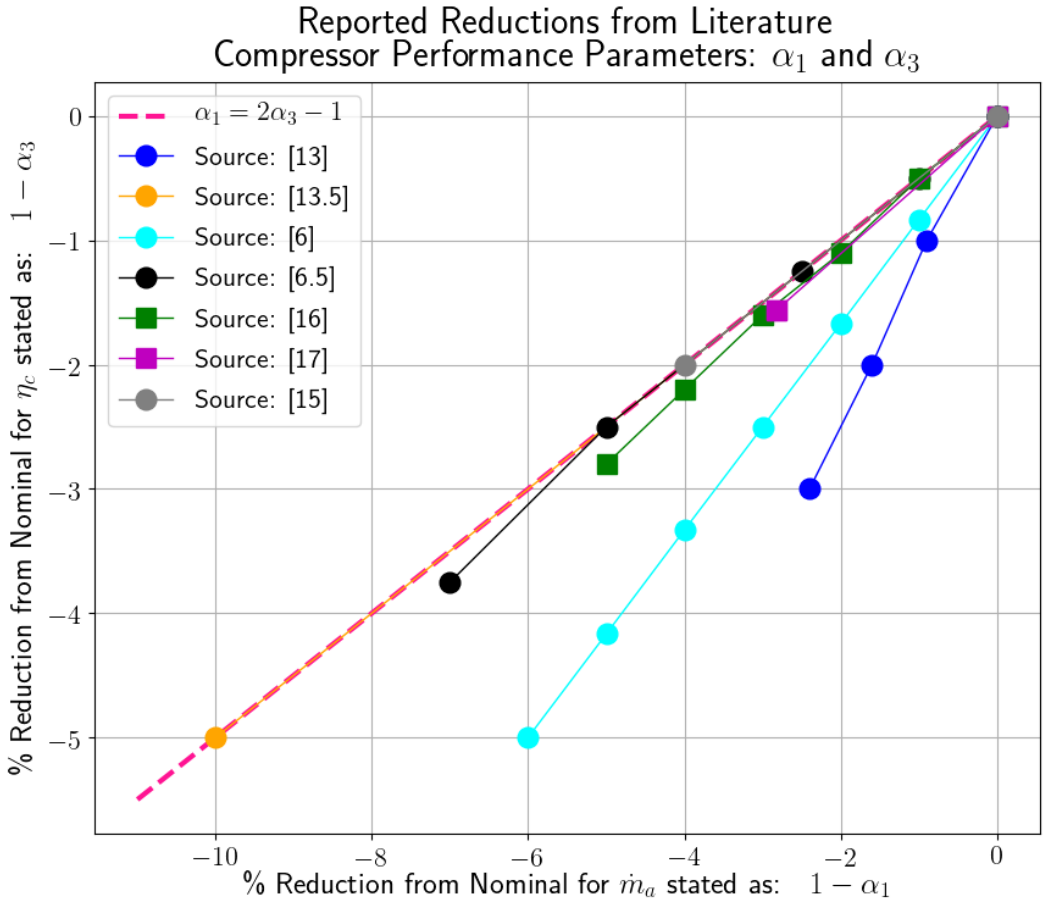


Figure 5.2: Relationship of multiplicative constants applied to compressor mass flow of air and compressor isentropic efficiency. Note that square markers indicate actual measured data, whereas circle markers are values from simulations. Sources with $X.5$ indicate when a source had two different simulations.

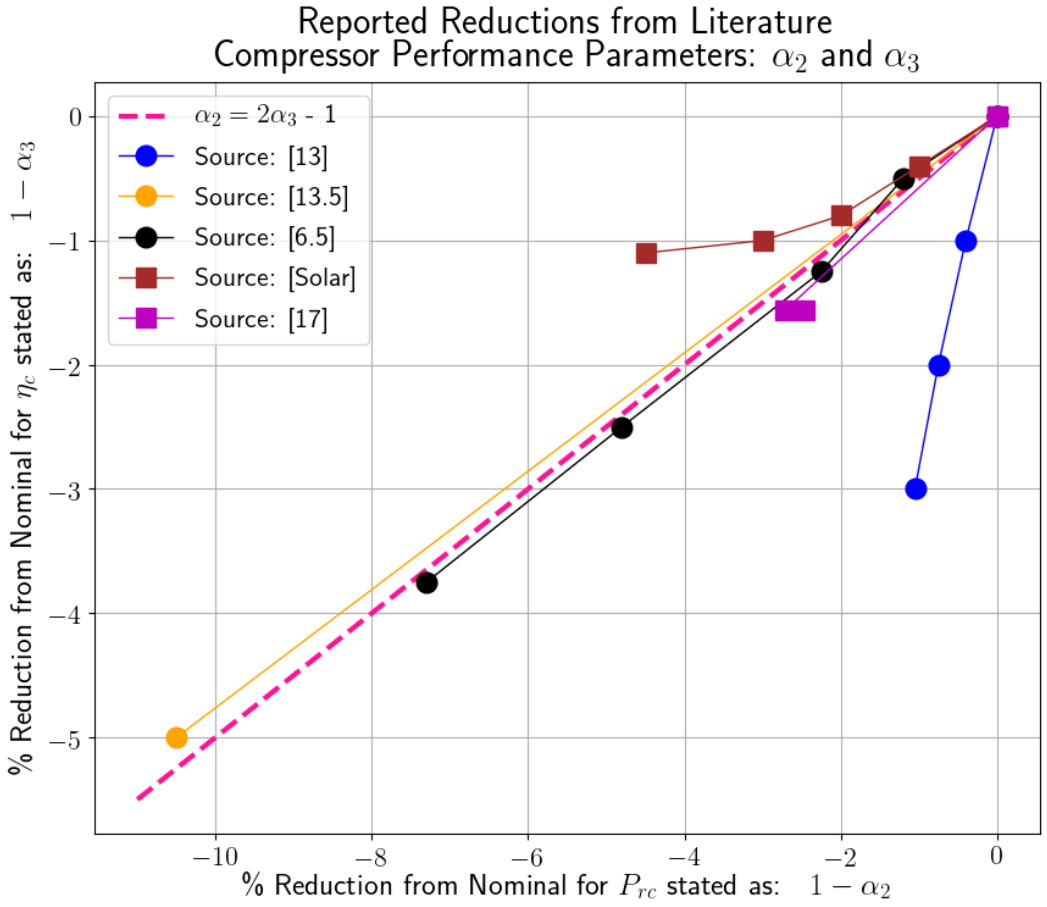


Figure 5.3: Relationship of multiplicative constants applied to compressor pressure ratio and compressor isentropic efficiency. Note that square markers indicate actual measured data, whereas circle markers are values from simulations. Sources with $X.5$ indicate when a source had two different simulations.

compressor air mass flow percent reduction and compressor isentropic efficiency percent reduction is approximately 2:1.

The data shown in Figure 5.3 again has wider variation. However, when limiting the compressor pressure ratio percent reduction to 3%, the field data appear linear, with slopes ranging from approximately 1 to 3. In general, we conclude the relationship between compressor pressure ratio percent reduction and compressor isentropic efficiency percent reduction is approximately 2:1, which, when coupled with the conclusions from Figure 5.3, further supports the 1:1 relationship between compressor air mass flow percent reduction and compressor pressure ratio percent reduction found from Figure 5.1.

Stating these observations in terms of Definition 1, the relationships identified in the literature are given in the following remark.

Remark 1. *Empirical compressor fouling rules estimated from previous compressor fouling simulations and field data can be established according to Definition 1 as:*

$$\alpha_1 \approx \alpha_2 \approx 2\alpha_3 - 1. \tag{5.1}$$

The primary conclusion we have drawn from the literature review is that fouling multiplicative factors have approximately linear relationships in the 0%-5% fouling regime. Indeed, the linear relationship between efficiency and airflow was studied by Zaba et. al., in which they derived an empirical relationship between compressor volume flow decrease ($\Delta\dot{V}$) and polytropic efficiency decrease ($\Delta\eta_p$) due to fouling, expressed as

$$\frac{\Delta\dot{V}}{\dot{V}} = Z_c \frac{\Delta\eta_p}{\eta_p} \tag{5.2}$$

Isentropic efficiency can be written as a function of polytropic efficiency and pressure ratio (assuming constant specific heat ratio). We can relate (5.2) to isentropic efficiency, by holding pressure ratio constant. The effect is that for a given pressure ratio, small changes in polytropic

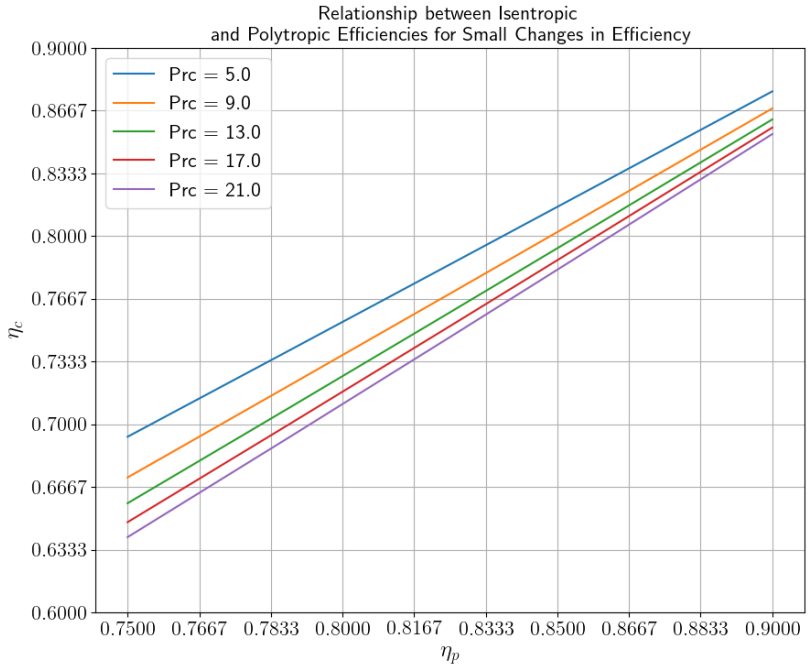


Figure 5.4: Relationships between polytropic and isentropic efficiency for constant pressure ratios. The relationships are linear and the slopes have small changes for wide ranges of pressure ratio.

efficiency correlate linearly with changes in isentropic efficiency. If we then allow for small changes in pressure ratio, the linear relationship persists, with nearly negligible changes in slope. These effects are observed in Figure 5.4.

When taken together, we conclude that a linear relation adequately models small changes in η_p and P_{rc}, which lead to small changes in η_c. Therefore (5.2) supports the linear relationships found in (5.1).

We note that site specific causes influence these relationships. Hepperle et. al studied some of these effects as they relate to (5.2) and concluded that the linear relationship was indeed valid and that the slope varied between 1.5-2.5, depending on site conditions [HTSS11].

We conclude this section with a look at the Solar engine data reported among the findings in Figure 5.5 with data from a Titan 250 engine operating in Asia. In the figure, an online water wash is observable 05-2019.

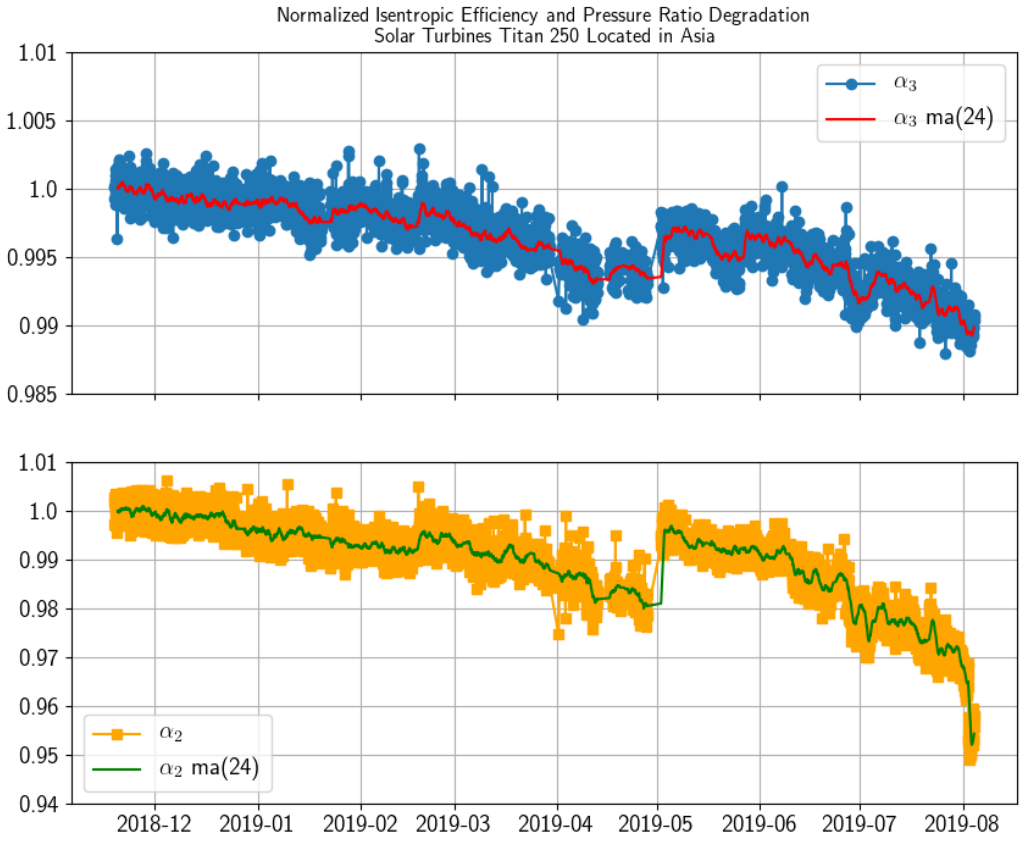


Figure 5.5: Solar Turbines Titan 250 compressor fouling that occurred at site. The fouling rate can be seen to be approximately linear until the final days of data, when the pressure ratio fouling rate becomes nonlinear.

Consider the moving average data before the water wash. In terms of (5.1), Figure 5.5 shows that $\hat{\alpha}_3 \approx 0.9925$, which leads to a calculation of $\hat{\alpha}_2 = 0.985$, quite close to the observed value $\alpha_2 = 0.981$.

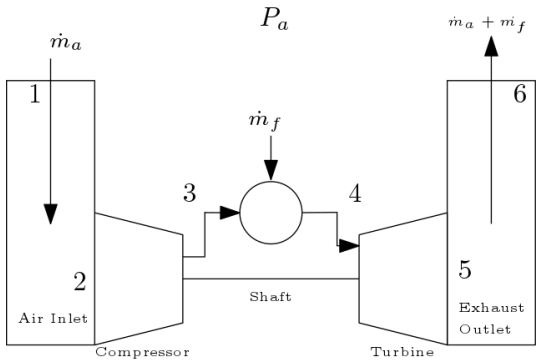


Figure 5.6: Standard station numbering

5.4 Calculating Fuel Mass Flow Based on Fouling Parameters

Industrial gas turbines produced by OEMs typically undergo myriad tests before delivery to the customer. During these tests, the test rig is usually outfitted with a wide array of calibrated measurement devices, providing a richer set of valuable data upon completion of these tests. We will utilize this augmented data set to calculate *least squares* estimates for various parameters of our model at the nominal state. We shall use the standard [Mat96] numbering system throughout this chapter, presented in Figure 5.6.

We approximate the pressure drop across the combustor as a percentage of the compressor discharge pressure by scalar δ_{comb} in (5.3). The exhaust outlet pressure drop can similarly be expressed with scalar δ_{ex} in (5.4), applied to ambient pressure. This type of approximation can be applied to the inlet as well; however, compressor inlet pressure is monitored on most industrial gas turbines in operation, alleviating the need for this approximation.

$$P_4 = \delta_{comb} P_3 \qquad \text{with,} \qquad 0.9 < \delta_{comb} < 1 \tag{5.3}$$

$$P_5 = \delta_{ex} P_a \qquad \text{with,} \qquad 1 < \delta_{ex} < 1.10 \tag{5.4}$$

$$P_{rc} = P_3 / P_2 \tag{5.5}$$

These scalar values are assumed to negligibly vary at full load between nominal operation and degraded operation; thus, we set these as constants, determined from an engine's test certification data.

The equations used to simulate steady state GTs can be found in various textbooks [Mat96, SRC01, Boy12] and are not reproduced here. Note that bleed air is assumed a small fraction of the total flow through the gas path, and is neglected. The error imposed by this assumption is mitigated when we consider the reduced order model, by inclusion of a flow correction coefficient, K, in section 5.4.1.

Since we are interested in fouling at full load operation, we assume a constant value for η_t, allowing us to isolate compressor fouling. Both Q_R and R are known constants. Then, given the values of P_a, $\dot{m}_a$, P_{rc}, η_c, and $\dot{W}_{sh}$, a solution vector $\bar{\boldsymbol{s}}$ can be obtained by numerical simulation, where

$$\bar{\boldsymbol{s}} = (\dot{m}_f, P_3, P_4, P_5, T_3, T_4, T_5, c_{p23}, c_{p34}, c_{p45}, c_{v23}, c_{v34}, c_{v45}, \gamma_c, \gamma_t)$$

yielding all gas path parameters of interest.

5.4.1 Steady State Gas Turbine Simulation: Reduced Order

Our first goal is to derive a closed form equation that quantifies the incremental change of fuel flow resulting directly from compressor fouling. Such an equation can be used by operators of GTs who do not have engine design software or full thermodynamic models of their machinery, to track changes in fuel consumption attributable to compressor fouling.

The full set of equations used to simulate a GT cannot be solved analytically to give fuel flow as a function of compressor performance parameters, due to nonlinearities in temperature variables in the specific heat equations. In order to find a reduced set of equations, we fit a linear regression model to the specific heat functions over the two temperature intervals of interest,

namely (T_2, T_3) and (T_5, T_4) as seen in (5.6)-(5.8), where the regression variables are α_2 and α_3.

$$\hat{c}_{p23}(\alpha_2, \alpha_3) = 0.24986 + 0.00472\alpha_2 - 0.00889\alpha_3 \tag{5.6}$$

$$\hat{c}_{p34}(\alpha_2, \alpha_3) = 0.30599 + 0.00450\alpha_2 - 0.03758\alpha_3 \tag{5.7}$$

$$\hat{c}_{p45}(\alpha_2, \alpha_3) = 0.32912 - 0.00439\alpha_2 - 0.04778\alpha_3 \tag{5.8}$$

The data used to fit these functions comes from numerically solving the standard set of equations for 48 unique fouling cases, where α_1, α_2 and α_3 have been allowed to independently decrease by a maximum of 4%.

To validate the specific heat functions, a simple gas turbine was modeled using the Numerical Propulsion System Simulation (NPSS) environment. We will discuss the specifics of the model in a later section. The maximum error from (5.6)-(5.8) compared with the full NASA CEA function values yielded from the NPSS simulation gave

$$\max_{T \in (T_i, T_i + \Delta T)} |c_{p,NASA} - \hat{c}_p| \leq 1 \times 10^{-3}\%$$

where i is an integer indicating a GT station number. This result indicates excellent agreement in the temperature ranges of the GT simulation.

The low error magnitude is to be expected. As the compressor fouls, the temperature intervals (T_2, T_3) and (T_5, T_4) become slightly elongated, but since the $\hat{c}_p$ equations (5.6)-(5.8) are fit over this range, the slight increase has minimal effect on accuracy.

We substitute the corresponding $\hat{c}_p$ equations into the standard equations to find our reduced set of equations Ω. Note that the primary difference between Ω and what we refer to as a full set of equations are those equations linking the C_p, C_v and γ equations with the ideal gas model.

$$\Omega = \begin{cases} 0 = T_2 P_{rc}^{\frac{\hat{\gamma}_c - 1}{\hat{\gamma}_c}} - T_2 - \eta_c \Delta T_{32} \\ 0 = \Delta T_{54} - T_4 \eta_t \left(P_{rt}^{\frac{\hat{\gamma}_t - 1}{\hat{\gamma}_t}} - 1 \right) \\ 0 = (\dot{m}_a + \dot{m}_f)\hat{c}_{p45}\Delta T_{45} - \dot{W}_{sh} - \dot{m}_a \hat{c}_{p23}\Delta T_{32} \\ 0 = \hat{c}_{p34}((\dot{m}_a + \dot{m}_f)T_4 - \dot{m}_a T_3) - \eta_b Q_R \dot{m}_f \\ 0 = \delta_{comb} P_3 - P_4 \\ 0 = \delta_{ex} P_a - P_5 \\ 0 = P_3 - P_{rc} P_2 \end{cases}$$

where $\hat{\gamma} = \hat{c}_p/(\hat{c}_p - R)$ and $\hat{c}_p = \hat{c}_p(\alpha_2, \alpha_3)$ as defined in equations (5.6)-(5.8). Then, Ω is a system of 7 equations in 7 unknowns, and in particular, is linear in $\dot{m}_f$. This allows us to solve the system in closed form for $\dot{m}_f$ as a function of fouling parameters and parameters typically measured on an industrial GT.

5.4.2 Incremental Fuel Mass Flow as a Result of Compressor Fouling

A closed form equation for the fuel mass flow, based on compressor pressure ratio, compressor isentropic efficiency, and the mass flow of air, is derived by solving Ω algebraically.

The solution is given as the *mass flow function* (MFF) and relies only on inputs that are generally measured on a typical industrial gas turbine with exception to compressor airflow, and is presented below in equation (5.9),

$$\hat{\dot{m}}_f(\dot{m}_a, P_{rc}, \eta_c, T_2, P_2, P_a, \dot{W}_{sh}) = K \frac{\hat{c}_{p34}\left(a + \frac{bc}{d}\right)}{P_{rc}^{\frac{1}{\hat{\gamma}_c}} \eta_b \eta_c Q_R} \tag{5.9}$$

where

$$
\begin{aligned}
a &= \dot{m}_a T_2 \left(P_{rc} + (\eta_c - 1) P_{rc}^{\frac{1}{\hat{\gamma}_c}} \right) \\
b &= \delta_{comb} P_2 P_{rc} \left(\frac{\delta_{ex} P_a}{\delta_{comb} P_2 P_{rc}} \right)^{\frac{1}{\hat{\gamma}_t}} \\
c &= \hat{c}_{p23} \dot{m}_a T_2 \left(P_{rc} - P_{rc}^{\frac{1}{\hat{\gamma}_c}} \right) + \eta_c \dot{W}_{sh} P_{rc}^{\frac{1}{\hat{\gamma}_c}} \\
d &= \hat{c}_{p45} \eta_t \left(-\delta_{ex} P_a + \delta_{comb} P_2 P_{rc} \left(\frac{\delta_{ex} P_a}{\delta_{comb} P_2 P_{rc}} \right)^{\frac{1}{\hat{\gamma}_t}} \right) \\
K &= \text{ non-dimensional flow correction coefficient}
\end{aligned}
\tag{5.10}
$$

where the terms not appearing as arguments to the function are known constants.

To isolate the effect of compressor fouling, we hold P_a and $\dot{W}_{sh}$ constant. We expect $T_a \approx T_2$. We assume P_2 variation is negligible as fouling occurs in the range considered and that the significant portion of changes in air mass flow come from changes in air velocity. This assumption has been confirmed using the compressor stage stacking technique, in which inlet velocity triangles have airflow reductions represented by reductions in the magnitude of absolute and relative velocity vectors [MHFW89, LS86]. Furthermore, application of Bernoulli's equation through the inlet supports this claim. Taken together, the result provides us with the fuel mass flow equation dependent only on compressor parameters:

$$\hat{\dot{m}}_f = \hat{\dot{m}}_f(P_{rc}, \dot{m}_a, \eta_c)$$

and the change in fuel mass flow resulting solely from compressor fouling is given by the

incremental fuel mass flow equations,

$$\Delta\hat{\dot{m}}_f = \hat{\dot{m}}_f(\dot{m}_a, P_{rc}, \eta_c) - \hat{\dot{m}}_f(\dot{m}_a^*, P_{rc}^*, \eta_c^*) \tag{5.11}$$

$$\%\Delta\hat{\dot{m}}_f = 100\left(\frac{\hat{\dot{m}}_f(\dot{m}_a, P_{rc}, \eta_c)}{\hat{\dot{m}}_f(\dot{m}_a^*, P_{rc}^*, \eta_c^*)} - 1\right) \tag{5.12}$$

Taken together, (5.9)-(5.12) fully quantify the incremental difference in fuel flow due to compressor fouling, for a given load and ambient condition.

5.5 Validation of the Mass Flow Function with Test Cell Data and Simulations

The mass flow function (5.9) has a non-dimensional flow correction coefficient K (5.10). This value can be estimated from test cell data by implementing a least squares criterion given a data set with varying loads and a fuel blend similar to the expected site fuel blend. The coefficient calibrates for modeling simplifications, such as bleed flows out of the compressor and into the turbine, in solving Ω for $\hat{\dot{m}}_f$.

5.5.1 Test Cell Validation

To validate (5.9), we have collected 18 Solar Turbines Titan 130 engines' test cell data and used the least squares approach to estimate individual values of K for each engine. Every engine is from the same engine power rating, where both conventional and SoLoNOx engines have been included. Furthermore, we have assembled one complete test run per engine that will be used to estimate that engine's K value. Typical results are shown in Figure 5.7. We have included data ranging from 50%-100% maximum load to show how well (5.9) performs even when the full load assumption is violated.

As can be seen in Figure 5.7, the agreement of equation (5.9) is best at full load, which is

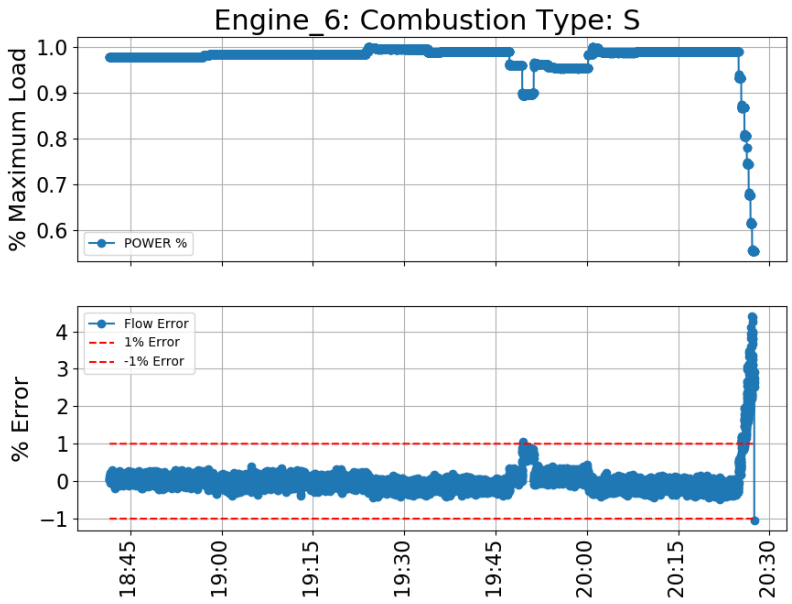


Figure 5.7: Engine # 6 test cell data. The top plot shows the % maximum load throughout the test. The bottom plot shows percent error between measured flow and calculated flow from equation (5.9), where $\alpha_1 = \alpha_2 = \alpha_3 = 1$. Note that for the duration of the test, T_2, P_2 and P_a vary less than 0.5%.

expected since full load data dominates the data set with which we apply least squares. That is to say, equation (5.9) has excellent agreement with measured data at or slightly below full load operation as intended.

We show each individual engine's K value in Figure 5.8. The conventional data set contains two outliers, for which we have not yet identified the cause. Once the outliers are removed, both the conventional and SoLoNOx engines have nearly identical estimated K values.

Remarkably, there is strong evidence that the K value is not dependent on the combustion system, nor is it dependent on the specific engine. This suggests that equation (5.9) captures the general physics of the GT and that our modeling assumptions are not too restrictive.

The total error is presented in Figure 5.9, given in %, from all 18 of the test cell data sets (approximately 20 hours running time, with a 2Hz sampling rate with load between 100%-50% of maximum load). The error is calculated as $\left(\frac{\hat{\dot{m}}_f - \dot{m}_f}{\dot{m}_f}\right) \times 100$ where $\hat{\dot{m}}_f$ is the calculated flow

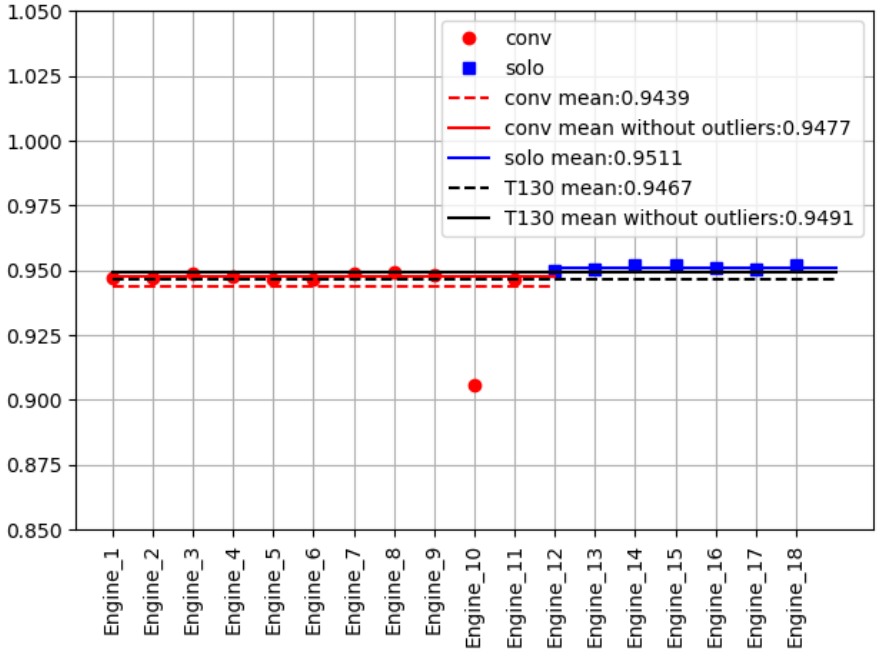


Figure 5.8: Estimated K values from Solar Turbines' test cell data using least squares estimation.

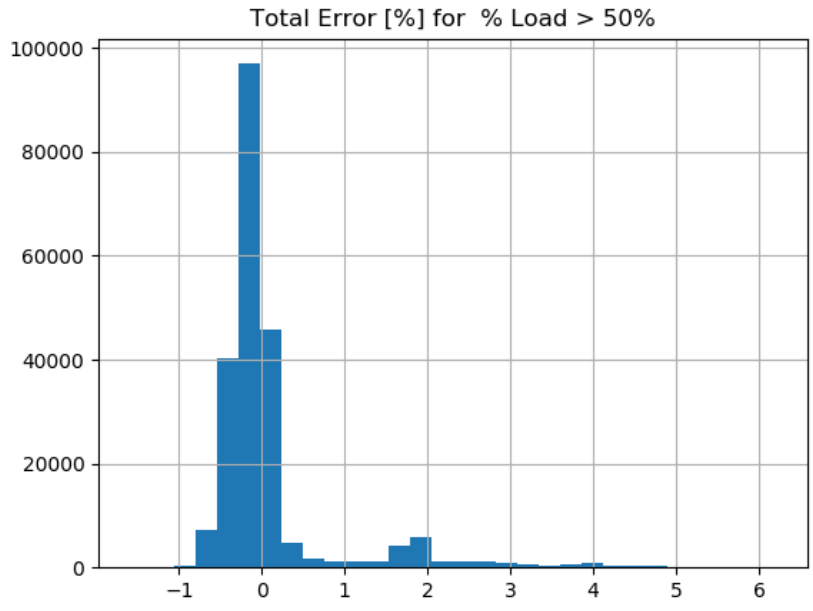


Figure 5.9: Error from test cell data. 93.8% of error is within $\pm 1\%$ of zero. Errors breaching this band occur when the engine is running at part load.

and $\dot{m}_f$ is the measured flow. As can be seen, the significant majority of error is within $\pm 1\%$ error, indicating excellent agreement at nominal conditions. Surprisingly, the deviation observed between 90% - 50% load is quite small. The error making up the "tails" of the distribution in Figure 5.9 are traceable to non-full load operation, where the largest error is observed at 50% load and is approximately 5.5%.

5.5.2 NPSS© Simulation

We develop a NPSS model of a simple cycle turbo shaft with attached generator to further validate equation (5.9) and perform fouling simulations. The industrial gas turbine model requires a number of initial conditions or design values to be specified initially which are listed in Table 6.1. Note that the turbine design pressure ratio is presented as P_4/P_5 in the table. The values found in table 6.1 will remain constant unless otherwise noted.

Table 5.2: NPSS Model Initial Conditions.

Ambient Static Pressure	14.696 [psia]
Ambient Static Temperature	519 [$^\circ R$]
Inlet Pressure Loss %	20%
Cmp. Design Pressure Ratio	14.7
Cmp. Design Air Mass Flow Rate	45.2 [lbm/s]
Cmp. Design Isentropic Efficiency	0.85
Fuel LHV	21496 [btu/lbm]
Combustor Pressure Loss %	2.5%
Combustor Efficiency	0.98
Turbine Design Pressure Ratio	11.07
Turbine Design Efficiency	0.895
Exhaust Outlet Pressure Loss Percent	2.68%
Shaft Design Speed	10000 [rpm]
Generator Design Efficiency	0.95
Electric Load In	5395 [kW]

NPSS Simulation Notes

In the design simulation, there is no assumed fouling and the NPSS model is in "Design Mode". The NPSS solver is set up with independent variable as fuel mass flow rate and dependent variable as overall shaft net torque. The fuel mass flow rate is varied to achieve an overall shaft net torque of zero at the design condition. Requiring zero net shaft torque ensures that a mechanical steady state condition has been reached.

In the off-design simulation, the NPSS model is set to "off-design" mode and is put into "audit-mode" and prescribed fouling conditions are given as initial conditions. Here, the NPSS audit factors are invoked as positive scalar values less than unity, which represent the compressor performance parameters in Definition 1. The NPSS solver adds dependent variable, static exhaust outlet pressure, and independent variable, turbine pressure ratio. Requiring a specific exhaust outlet pressure constrains the flow through the gas path to be physically consistent based on design assumptions. Varying the turbine pressure ratio to an off-design point ensures that the turbine efficiency remains constant, which isolates the effect of various degrees of compressor fouling.

NPSS Simulation

The NPSS simulation results are found by allowing α_1, α_2, and α_3 to vary independently of each other and without respect to time. In general, the fouling parameters increase with the simulation number. While these factors are certainly coupled through aerodynamic and thermodynamic relationships as shown in Figures 5.1, 5.2, and 5.3, by allowing them to vary independently we are able to validate the agreement between NPSS and (5.9).

Figure 5.10 presents the set of 216 different fouling regimes and their corresponding fuel flow, as a percent of nominal flow, calculated with the NPSS fouling simulation. The nominal fuel flow is given in the first simulation and carried through as the dotted orange line for reference.

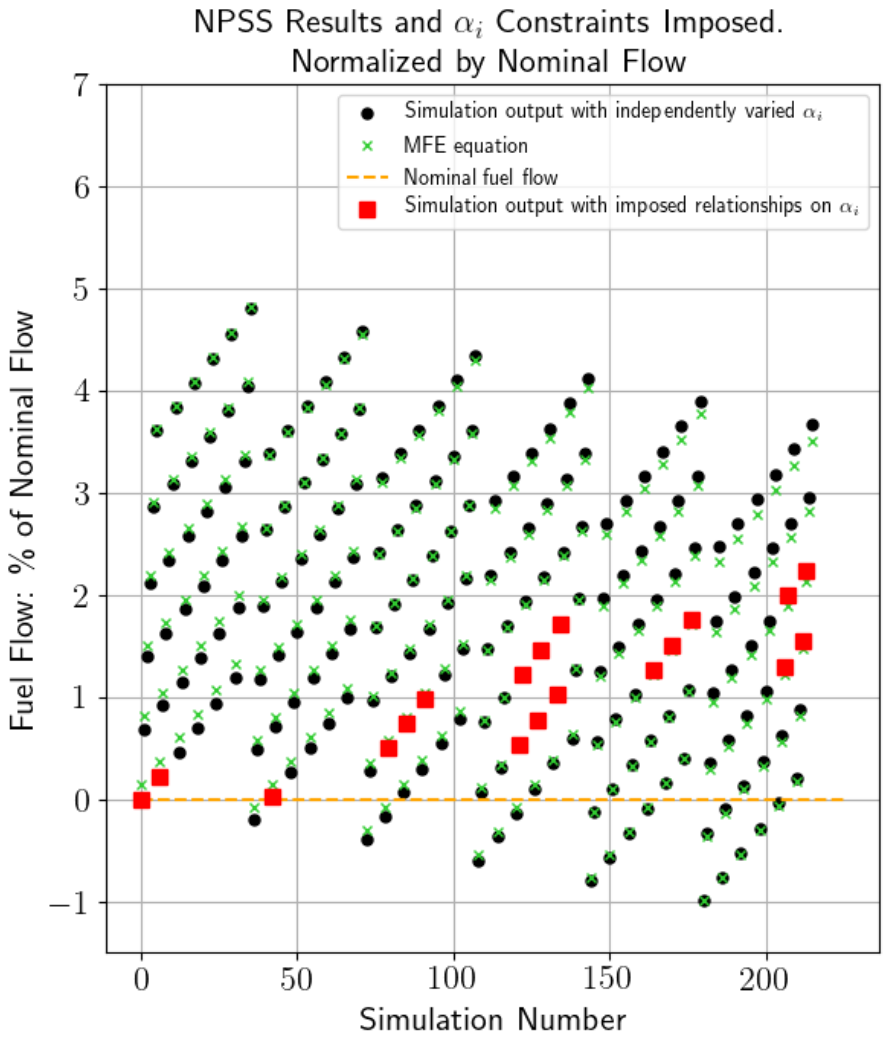


Figure 5.10: NPSS simulation output with α_i varied independently along with predictions from (5.9). Note the red squares indicate points where the fouling relationships (5.1) are satisfied.

Black circles represent the NPSS output and green x's are the output from the MFF equation (5.9). The K value was fit by applying a least squares criterion to the total NPSS output. The total error follows a Gaussian distribution with zero mean and maximum error of less than $\pm 0.2\%$, showing remarkably good agreement between (5.9) and NPSS.

We highlight with red squares the points consistent with the observations in (5.1), to show the effect of aerodynamic and thermodynamic couplings gleaned from data. These points are required to satisfy $|\alpha_1 - \alpha_2| \leq 0.01$ and $0.02 \geq \alpha_3 - \alpha_1 \geq 0$ for the range of $\alpha_1 \in [0.95, 1]$. Select constrained simulation results are given in Table 5.3. It can be immediately concluded that as fouling becomes more severe, additional fuel flow must be supplied to supplement the losses in the compressor when holding output power and ambient conditions steady.

As seen by the results, simply allowing the compressor performance parameters to vary independently produces a wide grid of possible changes in fuel flow, both positive and negative. However, only a small band of simulation values appear in line with data found in the literature.

For the most severe fouling case, where $(\alpha_1, \alpha_2, \alpha_3) = (0.95, 0.95, 0.97)$, equation (5.12) gives

$$\%\Delta \dot{m}_f = 2.14\%$$

which appears in agreement with values reported in the literature.

5.6 Diagnostics, Compressor Wash Optimization and Estimated Emissions

Section 5.5 has shown that equation (5.9) accurately calculates fuel flow. Therefore, both (5.11) and (5.12) can be trended in time to provide a diagnostic of incremental fuel flow attributable to compress fouling. This provides operators of GTs a diagnostic that identify and/or

Table 5.3: Simulation Results With α_i Constraints. Mass flow units in [lbm/s].

α_1	α_2	α_3	$\dot{m}_{f,NPSS}$	$\hat{\dot{m}}_f$	$\%\Delta\hat{\dot{m}}_f$
1.00	1.00	1.00	0.8155	0.8167	0.15%
0.98	0.98	0.99	0.8215	0.8220	0.81%
0.98	0.97	0.99	0.8235	0.8239	1.04%
0.97	0.97	0.99	0.8218	0.8220	0.80%
0.97	0.97	0.98	0.8275	0.8274	1.47%
0.97	0.96	0.98	0.8294	0.8294	1.71%
0.96	0.96	0.98	0.8278	0.8274	1.47%
0.96	0.95	0.98	0.8298	0.8294	1.71%
0.95	0.95	0.98	0.8281	0.8275	1.47%
0.95	0.96	0.97	0.8317	0.8309	1.89%
0.95	0.95	0.97	0.8338	0.8329	2.14%

corroborate the severity of compressor fouling. One implementation is to solve for the α_i for $i = 1,2,3$ parameters necessary to match an observed fuel flow; that is, solve the minimization problem,

$$\min_{\alpha_1,\alpha_2,\alpha_3} ||\dot{m}_f - \hat{\dot{m}}_f||_2$$

for a given data set to estimate the severity as exhibited by the α_i parameters. However, this approach may have more than one solution. Another implementation is to use the rules found in (5.1) and substitute them into (5.9) to find $\hat{\dot{m}}_f(\alpha_3, T_2, P_2, P_a, \dot{W}_{sh})$ which then depends only on commonly measured parameters. Therefore, multiple diagnostics can be constructed using (5.9) in some form.

In this section, we use the rules in (5.1) substituted into (5.9) unless otherwise specified. We find a first order approximation for the percent change in compressor isentropic efficiency and percent change in fuel flow, holding ambient conditions and output power constant. In the range of parameters observed on smaller industrial gas turbines (30 MW or less), we can state (5.12) as,

$$\%\Delta\hat{\dot{m}}_f \approx -\zeta(\%\Delta\eta_c) \tag{5.13}$$

where $\%\Delta\eta_c = \alpha_3 - 1 \leq 0$ and $\zeta \in [0.65, 0.75]$. Considering the example from section 5.5.2, $\zeta = 0.687$.

Equation (5.13) offers a linear function such that we can impose $\alpha_3 = \alpha_3(t)$, so that changes in fuel can vary in time with varying levels of compressor fouling. This time varying property becomes necessary when making predictions into the future, such as when an operator decides how to create a compressor wash schedule or tries to predict how much additional emissions could come from running in a fouled state. We discuss details of both scenarios in the next sections.

5.6.1 Optimizing Compressor Wash Schedules

An optimal compressor wash program can be found by solving the general minimization problem [ARDM12, AHdO19],

$$\min_n LP(t_0, t, n) = \min_n \left(\int_{t_0}^{t} C_{deg}(\tau, n) d\tau + nC_M \right)$$

where $LP(t_0, t, n)$ is the lost profit function representing total cost, $[t_0, t]$ is the interval of interest, n is the number of washes in the interval, $C_{deg}(\tau)$ is some function describing economic cost of fouling and C_M is a fixed cost of maintenance. In the case $C_{deg}(\tau, n)$ does not include varying fuel flow attributed to fouling, we can augment the problem statement by including (5.9) or (5.13). To do this, we must first parameterize fouling in time.

It is well documented in the literature that compressor fouling can be approximated by a decreasing linear function [HTSS11, BK18, MMG14, ARDM12, Igi17]. Using Definition 1 and a set of field data, we can estimate the decline of isentropic efficiency by fitting θ to data such that

$$\alpha_3(t)\eta_c^* = \eta_c(t) = \eta_c^* - \theta t \tag{5.14}$$

where $\theta > 0$ is now the unit of fouling for the specified unit of time and we require that $0 < \eta_c^* - \theta t < 1$ to be consistent with physical reality. We substitute (5.14) into (5.13), to find

$$\%\Delta\hat{\dot{m}}_f \approx -\zeta(\%\Delta\eta_c(t)) = -\zeta\frac{\eta_c(t)-\eta_c^*}{\eta_c^*}\times 100 = \zeta\frac{\theta t}{\eta_c^*}\times 100 \tag{5.15}$$

Equation (5.15) provides the approximate ratio of observed fuel flow to nominal flow per unit time, at the constant ambient conditions. Therefore, we can integrate the result in time, for some initial time t_0, with reference to (5.12)

$$\int_{t_0}^{t}\hat{\dot{m}}_f(\tau)d\tau \approx \dot{m}_f^*\int_{t_0}^{t}\left(\frac{\zeta\theta}{\eta_c^*}\tau+1\right)d\tau = \frac{\zeta\theta\dot{m}_f^*}{2\eta_c^*}(t^2-t_0^2)+\dot{m}_f^*(t-t_0) \tag{5.16}$$

where the first term in (5.16) represents the accumulated flow due to fouling (the fouling term) and the second term is the nominal accumulated flow.

Given some function $P(t)$ representing the cost of fuel $[\$/lbm]$ in time, the fouling term can be directly plugged into the general compressor wash optimization problem statement to capture the effect of fouling in time,

$$\min_n LP(t_0,t,n) = \min_n\left(\int_{t_0}^{t}P(\tau)\frac{\zeta\theta\dot{m}_f^*}{\eta_c^*}\tau + C_{deg}(\tau,n)d\tau + nC_M\right)$$

Note that the fouling term in (5.16) produces quadratic growth in time. Therefore, for a linear rate of fouling in isentropic efficiency, θ, and a fixed fuel price, $P(t)=P$, the fuel cost associated to fouling will grow quadratically in time. Should the fuel price be rising over the period of consideration, the cost will grow faster than quadratically.

It is worth mentioning that the time interval considered $[t_0,t]$ is not constraining. Intermediate intervals can be used, where piecewise linear functions can approximate non-steady ambient conditions and corresponding changes in fuel flow. Then, $[t_0,t]=\bigcup_{i=0}^{N}[t_i,t_{i+1}]$ with $t=t_N$ where

N parameterizes the number of subintervals considered. With this, we have,

$$\int_{t_0}^{t_N} \hat{\dot{m}}_f(\tau)d\tau = \sum_{i=0}^{N-1} \int_{t_i}^{t_{i+1}} \hat{\dot{m}}_{f,i}(\tau)d\tau$$

where $\hat{\dot{m}}_{f,i}$ represents (5.9) for given conditions in interval $[t_i, t_{i+1}]$.

Consider the example in section 5.5.2 where $\dot{m}_f^* = 0.8167[lbm/s]$, $\eta_c^* = 0.85$, and $\zeta = 0.687$. Suppose isentropic efficiency degrades linearly by 4% over six months and that the GT runs continuously. We can calculate an approximate cost of fouling by converting $\dot{m}_f^*$ to $[lbm/day]$ and assuming this value is a good proxy to daily fluctuations. If not, first create subintervals as discussed above. Continuing, we find $\dot{m}_f^* = 70,563[lbm/day]$. Plugging these values into equation (5.16),

$$\frac{\zeta\theta\dot{m}_f^*}{2\eta_c^*}(t^2 - t_0^2) = \frac{0.687(2.2)(70563)}{2(0.85)(10^4)}(180^2) = 203,259[lbm]$$

therefore an approximately additional 203,000 $[lbm]$, or 1.6%, of fuel are consumed due directly to fouling over the six month period. From US Energy Information Administration, the US Natural Gas Industrial price had an average value of 3.43 $[\$/10^3 ft^3]$ from May to October in 2019. If we approximate natural gas here to be methane, with density $\rho = 0.03547[lbm/ft^3]$, then $P = 0.0967[\$/lbm]$ and the final cost of fuel due directly to fouling over six months is $\$19,655$. Industrial GTs used for power generation are often dual fuel, capable of running on either natural gas or diesel fuel. For a GT running primarily on diesel, the total cost due to fouling could be significantly higher. The example shows that the cost of additional fuel due to fouling is certainly germane in the compressor wash optimization problem and should be included.

The above connection with compressor wash optimization uses (5.13), which is derived from (5.9) and (5.1). Using (5.9) directly and without imposing (5.1), requires each of the compressor performance parameters of Definition 1 to become time varying, which is a topic of further research.

5.6.2 Estimating Additional CO_2 Due to Fouling

Equation (5.9) provides an estimate of fuel flow and (5.11) provides the incremental difference between nominal and fouled flow rates. These equations can be used to calculate bounds on additional emissions of CO_2 by applying chemical and mass equilibrium equations when the fuel composition is known. At high enough firing temperatures, CO exists only in trace amounts and almost every carbon atom either binds to oxygen to form CO_2 or exits as an unburned hydrocarbon molecule. This conclusion comes from analysis of the equilibrium constant for the equilibrium equation,

$$\mathrm{CO_2} \rightleftharpoons \mathrm{CO} + 0.5\,\mathrm{O_2} \quad \text{where,}$$
$$K_p(T_{comb}) \leq K_p(1800K) \approx e^{-8.497} \approx 2.04 \times 10^{-4}$$

where the adiabatic flame temperature is $T_{comb} \leq 1800K = 3240°R$. Note that $K_p = f(T_{comb})$ for some function f and experimental values of K_p show that f is an increasing function [MSBB10] of temperature and very small values of K_p imply equilibrium favors the reactants [AJ08], which is CO_2 in this case.

Recall that $m_x = n_x M_x$ where n_x is the number of moles of substance x and M_x is the molecular weight of substance x. Consider n_x moles of a hydrocarbon fuel of molecular form $C_v H_w$. This fuel molecule contains v carbon atoms, therefore an upper bound on the number of moles of CO_2 emitted by combustion of n_x moles is given by

$$n_{CO_2,act} \leq n_x v \tag{5.17}$$

and correspondingly, an upper bound on the mass $[lbm]$ of CO_2 emitted by combustion of n_x

moles is

$$m_{CO_2,act} \leq n_x v \frac{M_{CO_2}}{M_{C_V H_W}} \tag{5.18}$$

where $n_{CO_2,act}$ is the actual number of moles of CO_2 and $m_{CO_2,act}$ is the actual mass $[lbm]$ of CO_2 produced by combustion.

We can approximate the total additional mass $[lbm]$ of CO_2 due to fouling by utilizing equation (5.16). Taking the time derivative, we have $\dot{n}_x = \dot{m}_x / M_x$. Utilizing the bounds in (5.17) and (5.18), we have,

$$\dot{m}_{CO_2,act} = \dot{n}_{CO_2,act} M_{CO_2} \leq \dot{n}_x v M_{CO_2} = \left(\frac{\hat{\dot{m}}_f}{M_{C_V H_W}} \right) v M_{CO_2} \tag{5.19}$$

and by integrating in time, it follows,

$$m_{CO_2,tot} = m_{CO_2,nom} + m_{CO_2,foul} \leq v \frac{M_{CO_2}}{M_{C_V H_W}} \int_{t_0}^{t} \hat{\dot{m}}_f(\tau) d\tau \tag{5.20}$$

where $m_{CO_2,tot}$ is the total accumulated mass of CO_2 such that $m_{CO_2,nom}$ is the nominal accumulated mass and $m_{CO_2,foul}$ is the accumulated mass attributable to fouling. Applying (5.20) and utilizing the fouling term found in (5.16), it follows

$$m_{CO_2,foul} \leq v \left(\frac{M_{CO_2}}{M_{C_V H_W}} \right) \frac{\zeta \theta \dot{m}_f^*}{2\eta_c^*} (t^2 - t_0^2) \tag{5.21}$$

Continuing with the example in section 5.6.1, the fuel considered is CH_4 so that $v = 1$, and the total mass of CO_2 emitted due to fouling is calculated to be

$$m_{CO_2,foul} = \frac{44.01}{16.04} (203,259) = 557,695 [lbm].$$

which is a 1.6% increase in CO_2 emissions attributable directly to compressor fouling.

5.7 Conclusion

This chapter has accomplished multiple goals. First, we have established approximate relations between axial compressor performance parameters and their evolution during compressor fouling based on a literature review. Second, we have derived a physics based, hybrid equation (5.9) for predicting the increase in fuel consumption given a specified level of axial compressor fouling and scrupulously validated our equation with both field and simulation data.

Next, we used the derived function (5.9) and rules extracted from literature, (5.1), to deduce a simple approximation, (5.13), which can be directly placed into compressor wash optimization objective functions, to increase the accuracy of the cost minimization problem, as shown in section 5.6.1. We found the increase in accumulated fuel flow attributable to fouling scales quadratically for set fuel prices, and produces appreciable increases in total fuel cost given a long enough time period and an aggressive enough fouling rate. Future research should assess the magnitude of the quadratic growth for various rates of fouling.

The final contribution showed how (5.9) can be used to approximate upper bounds on CO_2 emissions. Section 5.6.2 derived upper bounds on the expected CO_2 mass attributable to fouling by utilizing equation (5.9). The accumulation of CO_2 was shown to similarly be quadratic assuming a general hydrocarbon fuel molecule.

Future research should validate or amend the rules in (5.1) with data and underlying physics as well as find methods to make all of the parameters in Definition 1 time varying.

Chapter 5, in full, is a reprint of the material as it appears in ASME Turbo Expo 2020. Allen, Cody; de Oliveira, Maurcio; Balaji, Puru; Holcomb, Chad; ASME 2020. The dissertation author was the primary investigator and author on this paper.

Chapter 6

Estimating Recoverable Performance Degradation Rates and Optimizing Maintenance Scheduling

6.1 Introduction

Chapter 6 changes topics from the earlier chapters. In this chapter the attention turns to optimization. In particular, we will be interested in finding optimal rates of degradation and using these rates to create optimized maintenance schedules. These ideas were introduced in 1.3.1 of the Introduction.

Section 6.3 will discuss a statistical method for extracting the constant rate of degradation, α, from observed data. By identifying maintenance sessions in the data and removing those corresponding data points, we will show that when measurement noise is distributed normally, α is simply the expected value of the observations.

Section 6.4 deals with optimizing maintenance schedules. In section 6.4.1, we derive

a total lost profit function based on recoverable degradation of a component and forecasts of economic conditions. In section 6.4.4, we discuss a method of dissecting the overall optimization interval into smaller independent intervals where we enforce uniform spacing which allows us to provide a closed form solution to the maintenance scheduling problem within a smaller interval. We briefly discuss the operator's choice at the beginning of each interval to perform maintenance or continue operating; however, included in this work is an optimal scheduling algorithm assuming an operator always starts a new interval. This results in the overall optimization method, in which we solve each sub-problem over a specified interval and combine the solutions to yield the final solution.

We conclude the chapter with a numerical example which compares the optimal solution to a "best case" scenario of uniform intervals to aid in understanding the process of determining the optimal maintenance schedule as well as illustrate the efficacy of the methods described in this chapter.

6.2 Chapter Notation

IID	Independent and Identically Distributed
$E[\cdot]$	Expected Value
$\sim N(\cdot,\cdot)$	Follows Gaussian Distribution
$(t_i,...,t_n)$	Vector of Specified Times
$\delta(t)$	Discrete Delta Function
$e(t)$	White Noise Function
$u(t)$	Unit Step Function
$D_T(t)$	Difference Function Over Set T
$erf(t)$	Error Function
α	Actual Rate of Degradation

$\hat{\alpha}$	Estimated Rate of Degradation
β_i	Actual ith Wash Recovered Performance Percent
$\hat{\beta}_i$	Estimated ith Wash Recovered Performance Percent
n_i	Number of Washes in Interval i
n_i^*	Optimal Number of Washes in Interval i
C_m	Cost of Materials for One Maintenance Session
C_d	Cost of Downtime for One Maintenance Session
C_M	Total Cost of 1 Maintenance Session
$P_{max}(\cdot)$	Theoretical Maximum Power Function
$P_{meas}(\cdot)$	Observed Maximum Power Function
$MP(\cdot)$	Market Price of Energy Function
$C(\cdot)$	Specific Cost of Energy Function
$Q(\cdot)$	Relative Profit Function
$R(\cdot)$	Realized Relative Profit Function
$C_{deg}(\cdot)$	Cost of Degradation Function
$PM_T(\cdot)$	Performance Metric Function
$LP(\cdot)$	Lost Profit Function
$t_{i,p}$	Maximum Postponement Time for Interval i

6.3 Degradation Rate Identification

In this section we discuss the need to estimate a degradation rate, or slope, necessary for the optimized maintenance schedule developed in section 6.4. We start by describing a general problem statement, then discuss a statistical solution. We then show that the task of estimating a degradation slope of a given component is a special case of the general problem and the result

follows immediately.

Before describing an identification process, we discuss the importance of an accurate estimation of the degradation rate. Our degradation rate, α, will be a major component of our optimization schedule, and final solution. We will show later in the chapter that the optimal number of maintenance sessions varies non-linearly on α; in particular, we will find the optimal number of maintenance sessions in interval i, n_i^*, varies as $n_i^* \sim \alpha^{1/2}$ which highlights the importance of accurately estimating a robust degradation rate. Overestimates of α will result in more frequent maintenance sessions and additional maintenance costs whereas an underestimate of α will result in less frequent maintenance sessions and lost profit due to loss of efficiency.

For increased clarity, we summarize the method of identifying α before describing it in mathematical detail. We first establish a mathematical model of the degradation and recovery process. Then we will use simulated data to fit the parameters of the degradation model, and specifically, to estimate the degradation rate, α. We do this by making an assumption that the noise from the measured degradation signal is normally distributed about the identified mathematical model. From this assumption, we may utilize the mathematical properties of the expectation and variance operators to show that our reset (or maintenance sessions) will tend to be the largest spikes when looking at consecutive differences in the performance metric function. Once these maintenance sessions have been identified, they can be removed from the data set. The remaining data then represents observations of the degradation rate plus noise. From our assumptions, α will be the expected value, or mean, of this data set. The remainder of this section provides the necessary rigor to support these claims.

6.3.1 Estimating Degradation Model Parameters

We begin with a parameterization of a constant degradation rate *Performance Metric function*, $PM(t)$. Let $T = (t_0, t_1, t_2, ..., t_n)$ such that $t_0 < t_1 < \cdots < t_n$, and consider a function of

the form

$$g_T(t) = -\alpha t + \sum_{i=0}^{n} u(t-t_i)\beta_i$$

where $\alpha > 0$, $\beta_i > 0$ for each $i = 0, 1, 2..., n$ scalars and $u(t)$ is the *Heaviside step function*, or unit step function[Rob12],

$$u(t) = \begin{cases} 0 & t < 0 \\ 1 & 0 < t \end{cases}$$

Then, $g_T(t)$ creates a "sawtooth" function when plotted in 2D. For example, Fig. 6.1 shows a function $g_T(t)$, hiding behind actual observations, which have been simulated and represent measurement noise.

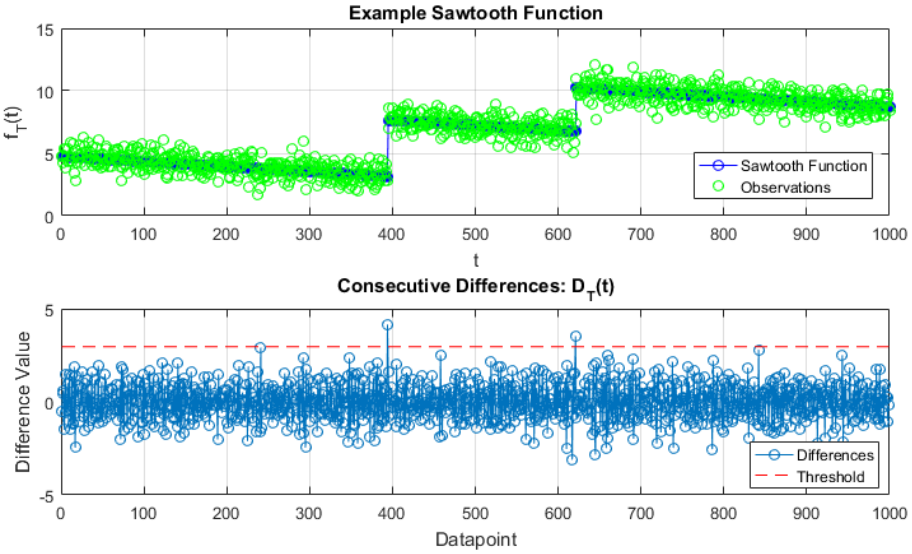


Figure 6.1: Top: Example sawtooth function and observations. The function was generated with $T = (0, 392, 619)$, N = 2, $\alpha = 0.004$, $\beta_0 = 4.732$, $\beta_1 = 4.4967$ and $\beta_2 = 3.4919$. The observations are generated by adding zero-mean Gaussian white noise with 0.5625 variance to the function. Bottom: Difference values. Threshold value set to $3\sigma_D$, where $\sigma_D = 2.961$ is the standard deviation of the set of difference values.

Note that if we further specify an interval over which $g_T(t)$ is valid, $[T_0, T_1)$ for example, where we set $T_0 = t_0$ and $T_1 = t_n$ and we require for each $0 < i \leq n$, $\beta_i = \alpha(t_i - t_{i-1})$, then we

define

$$PM_T(t,n) = \alpha\left(-t + \sum_{i=1}^{n} u(t-t_i)(t_i - t_{i-1})\right) \tag{6.1}$$

and call this the constant degradation rate, *performance metric function*. This function will be our parameterized version of $PM(t)$ in section 6.4.1. We analyze the more general case of unknown structure on the β_i values in this section, and note that equation (6.1) is a specific case of the general problem.

In the case of a function without measurement noise, identification of the constant slope α is trivial, as the slope is given by almost any consecutive pair of points; that is, consecutive points that do not include a maintenance session. However, now consider the function when measurement noise is present,

$$f_T(t) = g_T(t) + e(t) \tag{6.2}$$

where $e(t)$ is zero mean white noise, i.e., a sequence of IID random variables sampled from a normal (Gaussian) distribution[Lju99]. Hence, $e(t) \sim N(0,c)$, where $c > 0$. Now, the difference between a consecutive pair of points leaves α plus or minus some measurement noise, $f_T(t) - f_T(t-1) = -\alpha + e(t) - e(t-1)$ for almost any t. Recall that the difference of two IID random variables is itself an IID random variable. Indeed, $X \sim N(\mu_x, \sigma_x^2)$ and $Y \sim N(\mu_y, \sigma_y^2)$ then for $Z = X - Y$ we find,

$$E[Z] = E[X-Y] = E[X] - E[Y] = \mu_x - \mu_y$$
$$Var[Z] = Var[X-Y] = Var[X] + Var[Y] = \sigma_x^2 + \sigma_y^2$$

Hence, if $e(t) \sim N(0,c)$, then $e(t) - e(t-1) \sim N(0,2c)$. Define the difference function as

$D_T(t) = f_T(t) - f_T(t-1)$, then we have,

$$\begin{aligned} D_T(t) &= f_T(t) - f_T(t-1) \\ &= -\alpha t + e(t) + \sum_{i=0}^{n} u(t-t_i)\beta_i - \left(-\alpha(t-1) + e(t-1) + \sum_{i=0}^{n} u(t-(t_i+1))\beta_i \right) \\ &= -\alpha + e(t) - e(t-1) + \beta_i \delta(t-t_i) \end{aligned}$$

and if we consider the expected value of $D_T(t)$,

$$E[D_T(t)] = -\alpha + E[e(t) - e(t-1)] + \beta_i \delta(t-t_i) = -\alpha + \beta_i \delta(t-t_i)$$

where $\delta(t)$ is the *discrete* delta function.

We have just shown that the expected value of the difference of $f_T(\cdot)$ at two consecutive time values, t and $t-1$, is almost always $-\alpha$, except when $t = t_i$, in which case we have $-\alpha + \beta_i$ instead. Hence, if we can find each β_i, we will be able to calculate α by subtracting β_i at $t = t_i$ from the difference of $f_T(t)$ and $f_T(t-1)$ and calculating the mean value.

We now turn our attention to estimating the beta values. We again look at the difference

$$D_T(t) = -\alpha + \tilde{e}(t) + \beta_i \delta(t-t_i)$$

where $\tilde{e}(t) = e(t) - e(t-1) \sim N(0, 2c)$. In the case that $t = t_i \in T$, we have $D_T(t) = -\alpha + \tilde{e}(t) + \beta_i$ or $\beta_i = D_T(t) + \alpha - \tilde{e}(t)$.

We have $0 < \alpha \ll 1$, which is generally true for reasonable time scales. If we expect $\beta_i \gg Var(\tilde{e}(t))$, then from this simple representation, we see that there is a high likelihood that $\beta_i \approx D_T(t)$, since $\tilde{e}(t)$ is zero mean. Therefore, for most data sets we can set,

$$\hat{\beta}_i = D_T(t_i) \tag{6.3}$$

as our approximation to the true β_i. Then, $\beta = (\beta_0, ..., \beta_n)$ is constructed by choosing the J largest values, where J is a user decision, and then setting $n = J$.

Usually, a reasonable choice of J can be made by plotting the difference values along with the actual degradation measurements, and superimposing a threshold value. A good starting point for the threshold value is usually given by

$$Threshold = k\sigma_D = k\sqrt{2c} \tag{6.4}$$

where $k \in (2,3)$, $c = var(e(t))$ at some specified time t, and $n << N$, where N is the number data points. It can be shown (see [AW05]) that the probability a normal deviate lies outside $(\mu - k\sigma, \mu + k\sigma)$ is $P_k(|x - \langle\mu\rangle| \geq k\sigma) = 1 - erf(k/(\sqrt{2}))$, where $erf(\cdot)$ is the *error function*. For example, let $\tilde{e}(t) = e(t) - e(t-1) \sim N(0, \sigma^2)$ and consider $k = 3$. Then, $erf(3/\sqrt{2}) \approx 0.9973$ so the probability that a deviate, $\tilde{e}(t)$ at specified time t, lies outside of the range $(-3\sigma, 3\sigma)$ is $1 - 0.9973 = 0.0027$, which is the familiar result taught in elementary statistics classes. The bottom of Fig. 6.1 shows an example of a threshold value of $3\sigma_D$.

The threshold value yields two values, which we set equal to $\hat{\beta}_1$ and $\hat{\beta}_2$, and provides corresponding time locations, t_1 and t_2. From these values, we are able to remove the $\hat{\beta}_i$ values from $D_T(t)$ at $t = t_i$ to yield the set D_T^*. We take the mean value of the resulting set and find

$$\hat{\alpha} = E[D_T^*] = mean(D_T^*)$$

Additionally, one could choose to use the median value or the mean value of the inner quartiles of data to estimate $\hat{\alpha}$. These two alternatives are easier to implement but do not identify the β_i values. The $\hat{\beta}_i$ values are a direct estimate of the recoverable performance from a maintenance session and are usually desirable to identify, which is why we perform the process outlined.

The final result of this process applied to our simulated data is given in Fig. 6.2 where we have used our estimates $\hat{\beta}_i$ and $\hat{\alpha}$ to reconstruct the sawtooth function. The results from our

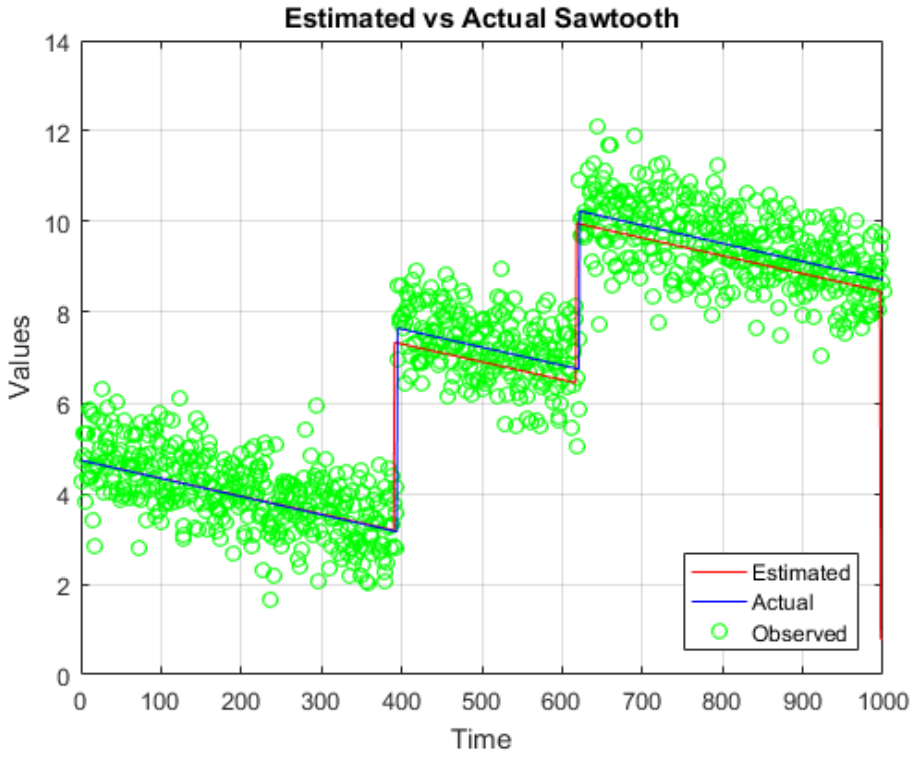


Figure 6.2: Example sawtooth function, observations and estimated sawtooth function. From the method outlined, we find $\hat{\alpha} = 0.00395$, $\hat{\beta}_0 = 4.732$, $\hat{\beta}_1 = 4.15$ and $\hat{\beta}_2 = 3.528$.

method, $\hat{\alpha} = 0.00395$, $\hat{\beta}_0 = 4.732$, $\hat{\beta}_1 = 4.15$ and $\hat{\beta}_2 = 3.528$ are well within a reasonable range and provide estimates that can be used in our economic analysis.

Lastly, consider if the function starts from $f_T(0) = 0$, with the expectation that maintenance will return full performance, then this is simply a special case of the general sawtooth function. Therefore, in the problem of degradation estimation, $\beta_i = \alpha(t_{i+1} - t_i)$, and the selection threshold for $\hat{\beta}_i$ values returned from $D_T(t)$ can be imposed based on how much time has passed since the previous wash. Simple iteration starting with an initial set of $\hat{\beta}_i$'s and estimation of $\hat{\alpha}$ leads to a robust statistical solution.

6.4 Condition Based Maintenance and Optimized Scheduling

In this section we describe a method of finding the optimal maintenance schedule based on an operator's relative cost model. Recall that the goal of the chapter is to present both a method of degradation rate identification as well as a framework for finding an optimal maintenance schedule. The methods do not place a restriction on the complexity or form of a cost model. However, the optimization problem complexity will vary based on the complexity of the economic model. We provide a derivation of the relative cost model and then optimize which provides a simple and tractable result.

6.4.1 Deriving the Overall Lost Profit Function

For the given component, the maintenance required to restore recoverable performance is usually a repeatable process with little variation. Therefore, the process generally has a fixed cost of material as well as an average amount of time to complete the task. Let C_m be the cost of materials for one maintenance session, including all costs for parts and the overall average cost of the labor of the technician (since the time to perform maintenance is fixed, if the technician is not working overtime, this cost should have little variation from its average). Let C_d be the cost of downtime; that is, the average lost revenue from not operating while maintenance is performed. Note that the difference in time scales between a single maintenance session versus the interval of time between maintenance is usually large enough that averaging C_d is justified. Then,

$$C_M = C_m + C_d \tag{6.5}$$

is the cost of performing one session of maintenance. For the remainder of this section, we assume that the component of interest is the axial compressor of a gas turbine in order to write down a specific cost model. The process developed could be repeated with small variations for different, critical components, such as a heat exchanger.

Define $P_{max}(t)$ to be the maximum power output of the gas turbine at standard conditions [Mat96] at time t. This value generally comes from a high fidelity gas turbine model. Let $P_{meas}(t)$ be the maximum measured power corrected to standard conditions. Define the *performance metric* as

$$PM(t) = \frac{P_{meas}(t)}{P_{max}(t)}$$

It is clear that PM should be between 0 and 1. Due to measurement noise and model simplifications, this value is sometimes slightly above 1; however, this is of minor concern in the present conversation.

In application to a specific component, the user must have this measure as it relates to the component's degradation, or a function that relates the component degradation to the power degradation. Note that in our example for compressor degradation, we have the following

$$\frac{Pr_{meas}(t)}{Pr_{max}(t)} \propto \frac{P_{meas}(t)}{P_{max}(t)}$$

that is, the ratio of compressor discharge pressures is linearly related to the ratio of powers. Therefore, one may be substituted for the other in the lost profit function without fundamentally changing the optimization output.

Now, define $Q(t)$ to be the relative profit generated at time t by utilizing non-degraded maximum power, $P_{max}(t)$. Then, the following model which has been adapted from [ARDM12]

describes time t profit,

$$Q(t) = (MP(t) - C(t))P_{max}(t)$$

where $MP(t)$ is the market price of energy, $C(t)$ is the specific cost of energy production from the turbine and $P_{max}(t)$ is the energy produced by the turbine, all at time t. Similarly, define $R(t)$ be the relative profit generated at time t by utilizing measured maximum power, that is, the realized relative profit,

$$R(t) = (MP(t) - C(t))P_{meas}(t)$$

The function $C(t)$ will depend on fuel costs at time t as well as fuel consumed at time t. We assume that fuel properties are constant, which is reasonable for a power plant running on sales gas. The fuel consumed at time t is partially dependent on the state of degradation present in the compressor. However, these incremental increases in fuel mass flow are low enough in the range of $PM(t)$ where most operators allow the turbine to run at, that we assume they can be parameterized in time by the operator. Hence, we assume the function $C(t)$ can be created based on an operator's forecast fuel purchase prices and is dependent only on time. Therefore at time t, the lost profit, or *cost of degradation*, is

$$C_{deg}(t) = Q(t) - R(t) = (MP(t) - C(t))(P_{max}(t) - P_{meas}(t)) = Q(t)(1 - PM(t))$$

and the *total lost profit* as a result of recoverable degradation during the interval (t_0, t) is

$$\begin{aligned} LP(t_0, t, n) = \int_{t_0}^{t} C_{deg}(\tau)d\tau + nC_M &= \int_{t_0}^{t} Q(\tau)(1 - PM(\tau))d\tau + nC_M \\ &= \int_{t_0}^{t} Q(\tau)(1 - PM_T(\tau, n))d\tau + nC_M \end{aligned} \tag{6.6}$$

where we use the parameterization of the performance metric function (6.1) to see the dependence of the LP function on the number of maintenance sessions.

We have established an operating, relative cost function for the time interval (t_0, t), namely, $LP(t_0, t, n)$. The optimization problem is then easily stated,

$$\min_n LP(t_0, t, n) \qquad \text{such that,} \qquad n^* = \arg\min_n LP(t_0, t, n) \tag{6.7}$$

and the minimizing argument, n^* represents the optimal number of maintenance sessions given our lost profit function and is ultimately what we aim to calculate.

6.4.2 Using Forecasts in Bounded Time Intervals

The functions $MP(t)$ and $C(t)$ are very specific to location and plant operation and must be developed by the operator. Usually, there are discrete forecasts of each over a given period of time or parts of each over a given period of time, e.g., forecasts for both electricity and fuel prices. Development of these functions is beyond the scope of this chapter.

For a given plant, suppose $MP(t)$ and $C(t)$ can be evaluated for given discrete times, $(\hat{t}_0, \hat{t}_1, ..., \hat{t}_k)$ using forecasts of spot price and fuel cost. These can be used to produce sets $MP = (MP(\hat{t}_0), MP(\hat{t}_1), ..., MP(\hat{t}_k))$ and $C = (C(\hat{t}_0), C(\hat{t}_1), ..., C(\hat{t}_k))$. Using a *zero-order hold* method [Lju99], we can make piecewise constant functions for both $MP(t)$ and $C(t)$. The zero-order hold method converts a discrete-time signal to a continuous time signal by holding each sample value constant for one sample interval. We can similarly estimate average values of $P_{max}(t)$ in each interval, since it is a reference value generated from a high fidelity gas turbine model. Then for $t \in (\hat{t}_i, \hat{t}_{i+1})$ as described in section 6.3.1, we have,

$$Q(t) = (MP(t) - C(t))P_{max}(t) = (MP(\hat{t}_i) - C(\hat{t}_i))P_{max}(\hat{t}_i) = Q_{i,i+1}$$

where $Q_{i,i+1}$ is constant in interval $(\hat{t}_i, \hat{t}_{i+1})$. By combining adjacent times (i.e., sample intervals) where $Q(t_i) \approx Q(t_{i+1}) \approx \cdots \approx Q(t_{i+j})$, we can set $Q(t) = Q_{i,j} = \frac{1}{j+1}\sum_{q=0}^{j} Q(t_q)$ where we are taking the simple average of $Q(t)$ over the time interval $(\hat{t}_i, \hat{t}_{i+j})$.

Then, the end result of this process is a set of constants $Q_{i,i+1}$ which correspond to intervals defined by $(t_0, t_1, ..., t_z)$ with $z \leq k$, where usually $z \ll k$. For example, in the interval defined by $t \in (t_2, t_3)$, we have $Q(t) = Q_{2,3}$ which is constant.

Let us now return to the lost profit model (6.6), where we set $t = t_z$ to be the last unit of time in which we have forecasts. From equation (6.1), $PM_T(t,n) = \alpha(-t + \sum_{i=1}^{n} u(t-t_i)(t_i - t_{i-1}))$. Utilizing the additive property of the Riemann integral [Apo74], we find,

$$\begin{aligned}
LP(t_0, t_z, n) &= \int_{t_0}^{t_z} Q(\tau)(1 - PM_T(\tau, n))d\tau + nC_M \\
&= \sum_{i=0}^{z-1} \int_{t_i}^{t_{i+1}} Q_{i,i+1}(1 - PM_T(\tau, n_{i+1}))d\tau + nC_M \\
&= \sum_{i=0}^{z-1} Q_{i,i+1} \int_{t_i}^{t_{i+1}} (1 - PM_T(\tau, n_{i+1}))d\tau + nC_M \\
&= \sum_{i=0}^{z-1} Q_{i,i+1} \int_{t_i}^{t_{i+1}} \left(1 - \alpha\left(-\tau + \sum_{j=1}^{n_{i+1}} u(\tau - t_j)(t_j - t_{j-1})\right)\right) d\tau + C_M \sum_{i=0}^{z-1} n_{i+1} \quad (6.8)
\end{aligned}$$

In equation (6.8), index i represents intervals of constant $Q(t)$, whereas index j represents wash intervals, which occur within a given interval defined by index i. That is, for specified i, we have interval (t_i, t_{i+1}) and within this interval, a wash schedule that partitions the interval into a set of n_{i+1} subintervals.

We see that the total lost profit function due to recoverable degradation is now primarily an integral of the performance metric in time.

6.4.3 Evaluating the Total Lost Profit Function

We would like to solve the global minimization problem given in equation (6.8). With the conditions imposed from section 6.4.2, we were able to set $Q(\tau) = Q_{i,i+1}$ over the interval (t_i, t_{i+1}), allowing us to pull that term out from under the integral. In doing so, all that remains to integrate is $1 - PM_T(t, n_{i+1})$ in the interval of interest. This causes problems, since our parameterization of $PM_T(t, n_{i+1})$ involves the unknown number of maintenance sessions, n_{i+1}, which is to be optimized. We propose solving the local optimization problem in each interval and treating the global minimization as a shortest path problem where the path depends on washing or not washing at the transition of interval to interval. This does not guarantee a globally optimal schedule, but does provide locally optimal schedules which when combined provide a global schedule.

In order to carry out the minimal path problem, we first introduce an equivalent form of $PM_T(t, n_{i+1})$. Consider the interval (t_i, t_{i+1}) where i is fixed and let $\Delta t_{i+1} = t_{i+1} - t_i$. The definite integral can be represented by a transformed version of the function $PM_T(t, n_{i+1})$; namely, a shift down one unit, and a reflection of the function across the x-axis in Cartesian coordinates. Geometrically, this produces a sequence of triangles based on the x-axis. Along each triangular segment, we have a linear function shifted right on the x-axis by the starting point of the segment, yielding,

$$\int_{t_i}^{t_{i+1}} \left(1 + \alpha\tau - \sum_{j=0}^{n_{i+1}} u(\tau - t_j)\alpha\Delta t_j\right) d\tau = \sum_{j=0}^{n_{i+1}-1} \int_{t_j}^{t_{j+1}} \alpha(\tau - t_j) d\tau \tag{6.9}$$

where $t_i = t_j\Big|_{j=0}$ and similarly, $t_{i+1} = t_j\Big|_{j=n_{i+1}}$, upon integrating,

$$\sum_{j=0}^{n_{i+1}-1} \int_{t_j}^{t_{j+1}} \alpha(\tau - t_j) d\tau = \frac{\alpha}{2} \sum_{j=0}^{n_{i+1}-1} \left(\tau^2\Big|_{t_j}^{t_{j+1}} - 2t_j\tau\Big|_{t_j}^{t_{j+1}}\right) = \frac{\alpha}{2} \sum_{j=0}^{n_{i+1}-1} \Delta t_{j+1}^2 \tag{6.10}$$

Therefore, we find a sum of quadratic terms that represent an equivalent area as the original form over the fixed interval (t_i, t_{i+1}). Then, in this interval, the $(i+1)$-interval, we have the area under the integral as,

$$Q_{i,i+1}\frac{\alpha}{2}\sum_{j=0}^{n_{i+1}-1}\Delta t_{j+1}^2$$

where each t_j, with $j = 0, 1, ..., n_{i+1} - 1$ are times when a maintenance session is conducted. If we require that each of the sub-intervals defined by (t_j, t_{j+1}) have uniform length, $q_{i+1} = \Delta t_{j+1}$ for each j, and it follows that $n_{i+1}q_{i+1} = t_{i+1} - t_i = \Delta t_{i+1}$. Utilizing this fact,

$$\begin{aligned} Q_{i,i+1}\frac{\alpha}{2}\sum_{j=0}^{n_{i+1}-1}\Delta t_{j+1}^2 &= Q_{i,i+1}\frac{\alpha}{2}\sum_{j=0}^{n_{i+1}-1}q_{i+1}^2 = Q_{i,i+1}\frac{\alpha}{2}q_{i+1}^2\sum_{j=0}^{n_{i+1}-1} \\ &= Q_{i,i+1}\frac{\alpha}{2}\left(\frac{\Delta t_{i+1}}{n_{i+1}}\right)^2 n_{i+1} \\ &= Q_{i,i+1}\frac{\alpha}{2}\Delta t_{i+1}^2\frac{1}{n_{i+1}} \end{aligned} \tag{6.11}$$

which is a *convex function* in n_{i+1} when $Q_{i,i+1} > 0$; that is, when the relative profit is positive. Note that since i is fixed, $\Delta t_{i+1} = t_{i+1} - t_i$ is a known quantity, therefore equation (6.11) can be evaluated. We bring this result into the context of the lost profit model, equation (6.8), for the given interval to find,

$$LP_{i+1}(n_{i+1}) = Q_{i,i+1}\frac{\alpha}{2}\Delta t_{i+1}^2\frac{1}{n_{i+1}} + n_{i+1}C_M \tag{6.12}$$

6.4.4 Optimizing the Total Lost Profit Function

In the previous section, we derived an equivalent form of the LP function that is dependent only on n_{i+1}, where the other values are constants for the interval (t_i, t_{i+1}). Since $n_{i+1}C_M$ is clearly convex, $LP_{i+1}(n_{i+1})$ is convex in n_{i+1}, and can be minimized by simply taking the derivative with

respect to n_{i+1} and setting the result equal to zero.

$$0 = \frac{dLP_{i+1}}{dn_{i+1}}\Big|_{n_{i+1}=n^*_{i+1}} \qquad \text{and,} \qquad n^*_{i+1} = \arg\min_n LP_{i+1}(n_{i+1})$$

In solving for the optimal n_{i+1}, we find,

$$n^*_{i+1} = \left(\frac{\alpha Q_{i,i+1}(t_{i+1}-t_i)^2}{2C_M}\right)^{1/2} \tag{6.13}$$

By convexity, (6.13) is the global minimizer[BV04], and if we shift to the closest integer about n_{i+1}, we are still guaranteed optimality of the solution. Thus, we have solved the local minimization problem for the interval (t_i, t_{i+1}).

In the solution for the local minimizer, the interval (t_i, t_{i+1}) was general; therefore, this solution can be applied to every interval determined by a constant value, $Q_{i,i+1}$. In parameterizing this optimization, we have implicitly required that at the beginning of every new interval defined by $Q_{i,i+1}$, we perform a maintenance session. As a result of this, we are actually imposing a lower bound on number of maintenance sessions. In fact, the minimum number of maintenance sessions will be equal in count to the number of intervals where $Q_{i,i+1}$ is constant; that is, the z intervals we created in section 6.4.2 and correspondingly, we will perform at least z maintenance sessions.

Within a given interval (t_i, t_{i+1}), where we have calculated the solution n^*_{i+1}, there is some flexibility in timing when maintenance is performed. From the geometry of the solution, within a given interval, the *LP* function is a set of same size triangles as discussed in the previous section. The triangle congruency within a given interval means that the overall area will not change if we slide the triangles in the given interval. See figure 6.3 for a depiction of this geometry. This realization which is dependent on the constant linear degradation, allows us to slide a given triangle to match the previous interval's final triangle height. Note that the number of maintenance sessions does not change; hence, this gives the operator the option of postponing a wash. Therefore, a beginning interval wash may be postponed a maximum amount of time

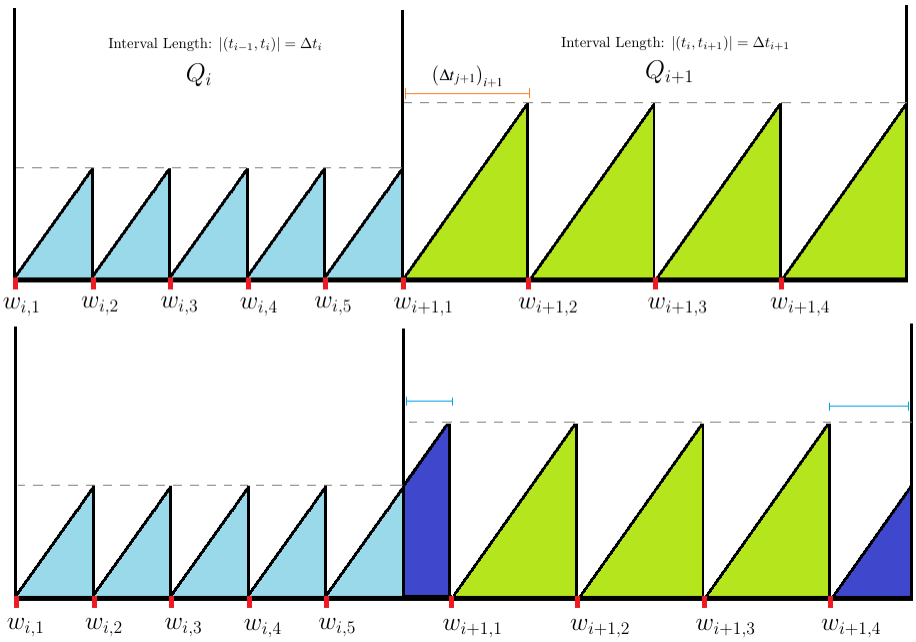


Figure 6.3: Note that once the triangle areas are specified, congruency allows sliding within an interval without change in overall area. The blue length scales sum to the orange length scale which shows how we can slice the final triangle and slide a portion to the beginning of the interval allowing for some flexibility in when the first maintenance session is performed for a new interval. Note that the number of washes is unchanged. The bottom plot shows the maximum postponement length $t_{i+1,p} = (\Delta t_{j+1})_{i+1}$.

equal to the width of a subinterval, which is known once n^*_{i+1} has been calculated. That is, for interval (t_i, t_{i+1}) and corresponding solution n^*_{i+1}, the maximum postponement time from the interval start is

$$t_{i+1,p} = \frac{\Delta t_{i+1}}{n^*_{i+1}} = \left(\Delta t_{j+1}\right)_{i+1}$$

where $t_{i+1,p}$ is the maximum time an operator can postpone the initial interval maintenance session from the beginning of interval (t_i, t_{i+1}).

The imposition of a minimum number of maintenance sessions is the main drawback of requiring a maintenance session at or close to the beginning of each period. However, it is this exact requirement that has allowed us to decouple the intervals where $Q_{i,i+1}$ changes. This yields a local and independent solution to the optimization problem in each interval. We then combine

the solutions to form the global solution. With this in mind, when defining periods over which $Q(t)$ is constant, the operator should keep z as small as possible.

For a given interval, (t_i, t_{i+1}), if the solution (6.13) results in $n^*_{i+1} < 1$, then the selection of intervals should be revisited. In the case that the intervals cannot be approximated by constants over a sufficient length such that $n^*_{i+1} \geq 1$, then our solution method will not guarantee optimality and the operator must make a judgment of whether to perform maintenance at the beginning of the session or not.

6.4.5 Example of Optimal Maintenance Schedule

We end this section with a short example. The reference data is mostly taken from [ARDM12] where we have modified some parameters to better reflect current pricing and costs. In table 6.1, we give the problem initial values. Note that the degradation rate would have been calculated from field data using the method in the previous section.

Table 6.1: Reference Example Data.

Total time duration	365 Days
Washing duration	5 hours
Plant Nominal Power	5.7 MW
Fuel Flow for Nominal Power	0.5556 lbm/s
Wash Cost	3000 US$
Other Costs	525 US$
Degradation Rate	0.0095 % / day

In this example, we allow the fuel price and market price of energy to vary throughout our year time frame

$$C(t) = 5.7 + \frac{1}{2}\cos\left(\frac{2\pi}{365}(t-200)\right) \qquad \left[\frac{\$}{MMBTU}\right]$$

$$MP(t) = 55 + 5\cos\left(\frac{2\pi}{365}(t-180)\right) \qquad \left[\frac{\$}{MWh}\right]$$

In Fig. 6.4, the relative profit function $Q(t)$ is given along with our approximations using the zero order hold constant model. Recall, a zero order hold simply sets a constant value for a specified range of time until the constant is updated, a common technique in signal processing.

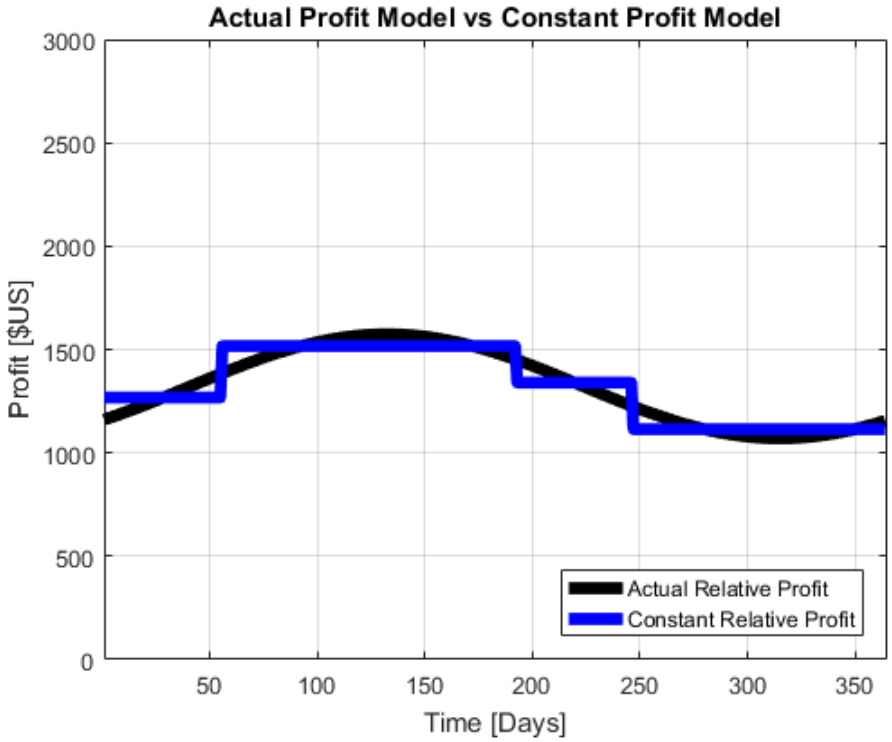


Figure 6.4: Using data from table 6.1 we produce $Q(t)$ and $Q_{i,i+1}$. Here, there are 4 intervals we have chosen where we approximate Q with the constant model. Interval selection is a user choice.

In Fig. 6.5, we see the lost profit functions $LP_{i+1,w}$. Note that for each interval, $LP_{i+1,w} < LP_{i+1,nw}$ so that n^*_{i+1} is found from the wash model, which is shown in the figure. The optimal number of washes per interval comes from the minimum integer of each of these curves.

In finding our optimal set of washes per period, namely, $\boldsymbol{n_w} = (n^*_1, n^*_2, n^*_3, n^*_4) = (2,6,2,5)$, we see that over the year period, 15 washes are called for. Additionally, the times between washes for the given interval can be calculated in days, $(28,23,27,24)$.

These wash points are plotted in Fig. 6.6 against our profit models and show the optimal washing schedule.

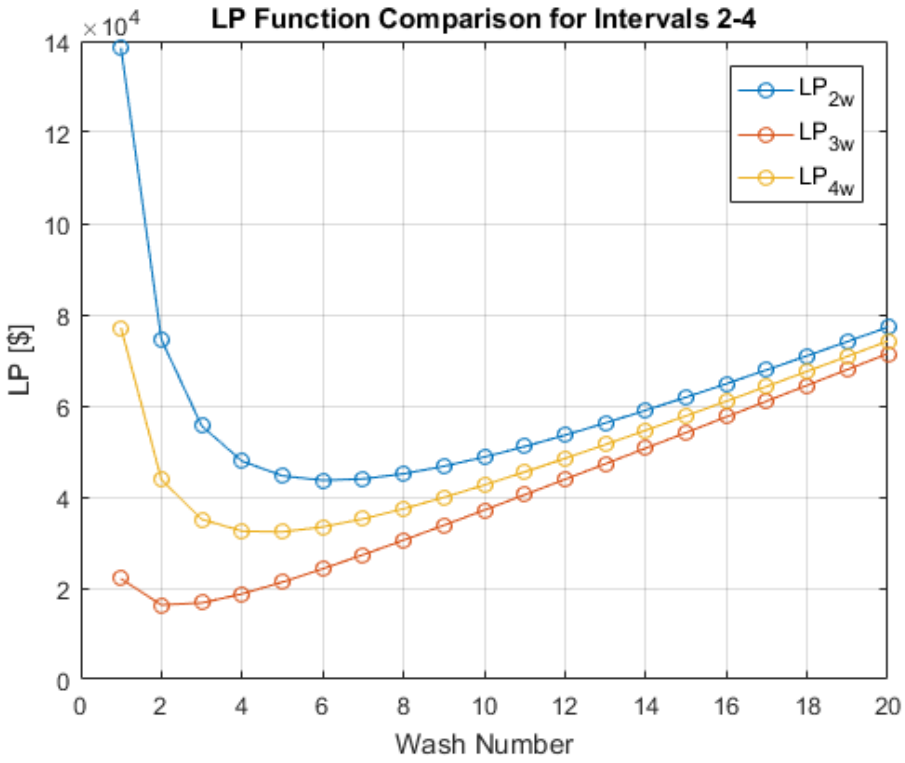


Figure 6.5: $LP_{i+1,w}$ models given over each interval for $i > 1$. The optimal number of washes per interval comes from the minimum integer of each of these curves. Note that each curve is convex, a key characteristic necessary for optimality.

For comparison, we will consider a static model where the wash intervals are equally spaced throughout the year. The operator would need to choose a method for determining the time between washes to implement such a schedule. In the example above, the *LP* function is calculated on a daily basis and then summed for the entire period, where the total number of washes in the 365 days has been predefined, and hence, so has the wash times. Figure 6.7 shows the effect of different sized, constant length wash intervals.

The values shown are the percentage increase in the lost profit model from the value of the lost profit model using the optimal washing schedule. For this particular example, the best case, constant interval wash schedule, results in 2.14% loss of the year's relative profit as compared with 0% loss of relative profit if using the optimal wash schedule. To be clear, this 2.14% is the incremental difference in lost profit for using the best, albeit not optimal, wash schedule in which

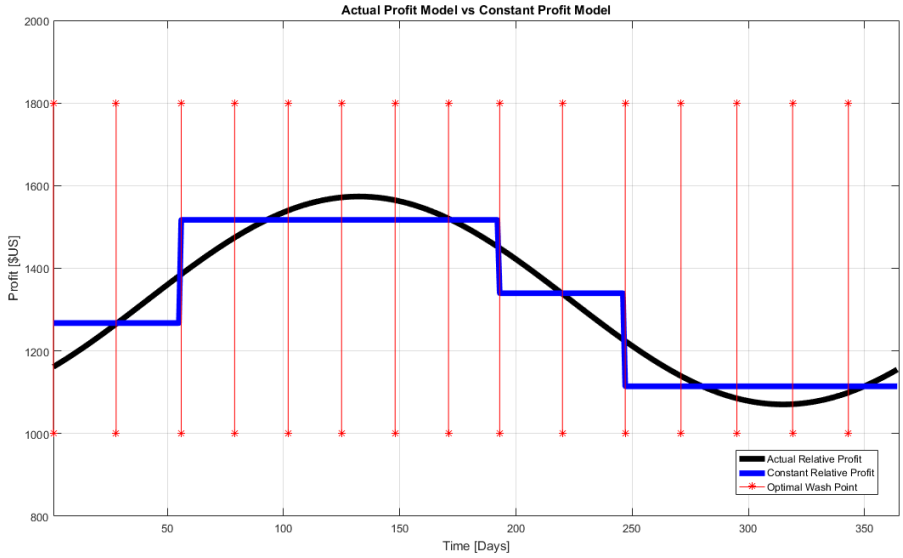


Figure 6.6: Depiction of overall wash schedule. Red vertical lines indicate washes or maintenance sessions. Within each sub-interval, time between washes varies, based on the solution of the optimization problem.

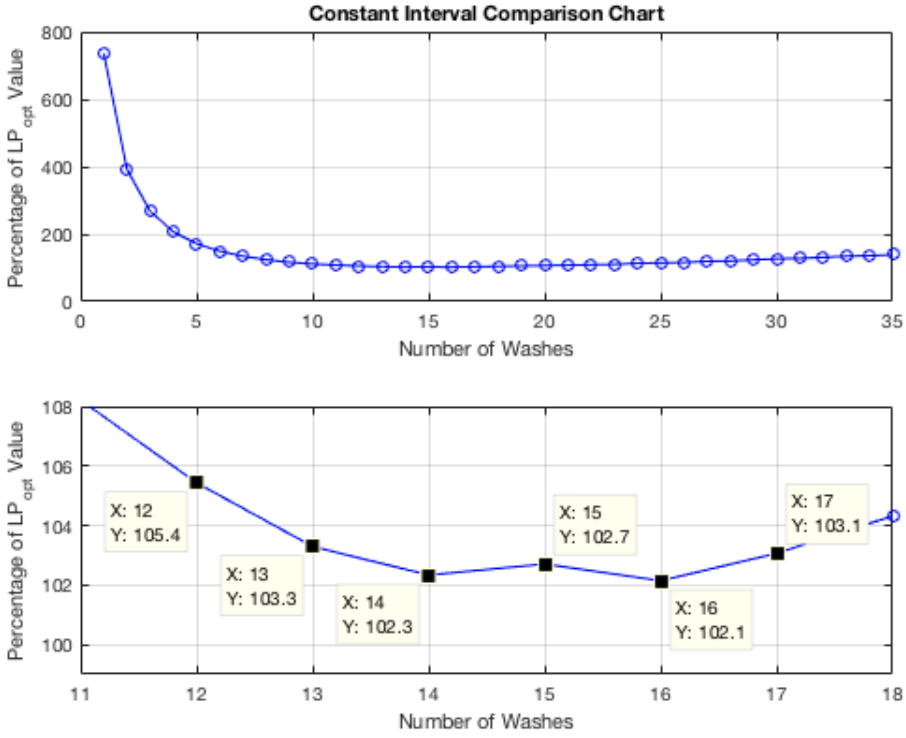


Figure 6.7: Comparison of percentage of LP functions. Shown is the value $(LP_{preset}/LP_{opt} - 1) \cdot 100$. Top is full view, bottom is zoomed into area of interest.

the intervals are set constant throughout the year.

Note that while 2.14% may not seem like a major savings, we have used the best case scenario for such an approach. This interval length is not known a priori by the operator and would have to be arrived at before such a maintenance schedule was implemented. If the operator chooses longer or shorter intervals between washes, then the percentage can grow considerably. For example, if the operator chooses to wash twice a month, or 24 times in the year, the percentage increases to 13.6%. Similarly, if the operator washes every other month, for 6 washes throughout the year, the percentage jumps to a staggering 49.4%. These values are of course for this specific example, however, the "take home message" remains: a non optimal number of maintenance sessions can severely increase lost profits due to the nonlinear nature of the cost model, or put plainly, a non optimal number of maintenance sessions can have a large impact on a company's net income.

The example in this section, in which the optimal maintenance schedule is compared with possible preset maintenance schedules of interest, elucidates the opportunity that an optimal, condition based maintenance schedule can provide the gas turbine operator.

6.5 Conclusion

In this chapter we have presented a framework for finding the optimal number of maintenance sessions for an asset that has recoverable degradation over time. We have shown that when forecasts of revenue and running costs are available, an economic lost profit model can be developed based on recoverable degradation. We then made some assumptions regarding operation, which allowed a single variable lost profit economic model to be derived, and we recast the optimization problem into a form that can be solved analytically. This lost profit model was shown to be convex in n, where n was the number of maintenance sessions in a given interval of time. The solution to each local lost profit model minimization solved the local scheduling

problem. Combining these locally optimal solutions produced a global maintenance schedule. We concluded with providing an example case of the optimal scheduling solution and provided a numerical simulation where all intervals are chosen to be of uniform size for comparison.

Our closed form solution is qualitatively equivalent to that given by Aretakis and Roumeliotis et al. [ARDM12] where they solved the problem iteratively. This provides additional validation of our method and solution. Quantitatively, our solution differs due to differences in the cost models used.

In addition, a method for asset degradation rate identification was developed. A discussion of the merit of the degradation rate identification followed, in which statistical expectations were discussed at length and general guidelines of rate identification were presented. These two main thrusts together compose a total solution to the maintenance scheduling problem as proposed in this chapter.

Lastly, the method outlined in this chapter has two drawbacks; namely, the requirement of a linear degradation rate and that we cannot guarantee global optimality, only local optimality. Future research will generalize the linear degradation rate, allowing for more accuracy in modeling true component degradation; in particular, generalizing the linear degradation rate to a quadratic degradation rate or a piece-wise linear degradation rate, as these types of profiles have been observed at various field applications. Additionally, future work aims to cast the degradation rate identification into an optimization problem, as opposed to solely relying on statistical methods.

Chapter 6, in full, is a reprint of the material as it appears in Journal of Gas Turbines and Power. Allen, Cody; de Oliveira, Maurcio, ASME 2018. The dissertation author was the primary investigator and author on this paper.

Chapter 7

A Minimal Cardinality Solution to Fitting Sawtooth Piecewise Linear Functions

7.1 Introduction

One inherent difficulty of fitting a sawtooth function to data is the fact that a perfect fit is always possible if one does not constrain the maximum number of discontinuities or jumps. See for example, the bottom plot in Figure 1.3 and Remark 3. One approach to limit the number of jumps is to solve a least-squares problem with an additional constraint on the cardinality of the set of jumps. The resulting problem, discussed Section 7.3, is non-smooth and nonconvex, given its combinatorial nature.

Made popular by techniques like LASSO regression [Tib96], we proceed by relaxing this cardinality constraint into a convex 1-norm constraint on the vector of jumps. Unlike LASSO regression, we prove that the cardinality of the optimal solutions obtained through the proposed relaxation is a monotonic function of the parameter of the relaxation. Our main result, Theorem 1, along with discussion on connections with LASSO regression is presented in Section 7.4. In

Section 7.5, we explore this result to propose an algorithm for the solution of sawtooth least-square estimation problems based on a series of convex relaxations. Technical proofs of the main result and supplemental results appear in Sections 7.6 and 7.7. We close the chapter with some conclusions and discussion on possible extensions of the main result to broader function approximations.

7.2 Chapter Notation

n	Total number of sample points available
$f_{\alpha\beta}$	General sawtooth piecewise linear function
$\|\cdot\|_0$	Non zero cardinality of $\cdot$
$\|\cdot\|_v$	Standard $v-$norm for $v > 0$
J	Removal Set
$\bar{J}_i$	The ith element in set $\bar{J}$
X	Matrix X
X_i	The ith column of matrix X
$\boldsymbol{x}$	General vector in $\mathbb{R}^s$ where s is given in text
$(\boldsymbol{x})_i$	The ith component of vector $\boldsymbol{x}$
$\boldsymbol{x}_m^*$	Optimal $\boldsymbol{x}$ vector for given bound m
$\mathbb{E}$	Expectation operator of random variable
$\mathcal{A}(\cdot)$	The active set for the given solution
$\boldsymbol{z} \odot \boldsymbol{x}$	Component-wise product, result is a vector
λ	Lagrangian constant
$\boldsymbol{e}_y$	Zero vector with solitary 1 at component y

7.3 Problem Formulation

The sawtooth function fitting problem considered in the present paper is a special form of the general PWL function fitting problem.

Definition 2 (Sawtooth function)**.** *Let $r \in \mathbb{N}$ and the discontinuity points t_j, $t_{j+1} > t_j$, $j = 0, \cdots, r-1$, be given. Define $f_{\alpha\beta} : [t_0, \infty) \mapsto \mathbb{R}$ such that*

$$f_{\alpha\beta}(t) = \begin{cases} \beta_0 - \alpha(t - t_0), & t \in [t_0, t_1) \\ \beta_0 + \beta_1 - \alpha(t - t_0), & t \in [t_1, t_2) \\ \vdots & \\ \sum_{j=0}^{r-1} \beta_j - \alpha(t - t_0), & t \in [t_{r-1}, \infty) \end{cases} \tag{7.1}$$

and $\alpha \geq 0, \beta_j \geq 0$ for $j = 0, 1, .., r-1$.

Remark 2. *Without loss of generality, we can assume that $f_{\alpha\beta}(t_0) \geq 0$, in which case we can impose $\beta_0 \geq 0$. If $f_{\alpha\beta}(t_0) \leq 0$ we can choose $c \geq f_{\alpha\beta}(t_0)$ and consider the shifted sawtooth function $\bar{f}_{\alpha\beta} = f_{\alpha\beta} + c$.*

Definition 3 (Cardinality)**.** *Let $\boldsymbol{x} \in \mathbb{R}^n$ and define $\|\boldsymbol{x}\|_0$ to be the number of non-zero entries of $\boldsymbol{x}$.*

Let $f_{\alpha\beta}$ be given as in Definition 2. Suppose that we are given n, possibly noisy ordered samples of $f_{\alpha\beta}$, that is, $\hat{\mathcal{Y}} = \{(\hat{t}_i, \hat{y}_i),\ i = 0, \cdots, n-1\}$, in which $t_0 \leq \hat{t}_i < \hat{t}_{i+1}$, $i = 0, \cdots, n-1$. Later in the paper we will constrain these samples to be drawn periodically. One data fitting problem is then

$$\min_{\alpha \in \mathbb{R}_+, \beta \in \mathbb{R}_+^r} J(\alpha, \beta) = \frac{1}{2} \sum_{i=0}^{n-1} (f_{\alpha\beta}(\hat{t}_i) - \hat{y}_i)^2 \tag{7.2}$$

where $J(\alpha, \beta)$ is the *residual sum of squares* (RSS) [TH09]. The optimization problem in (7.2) has a convex quadratic cost and linear constraints, making it a *convex quadratic program* [JN06], which can be efficiently solved if the r discontinuity points t_j, $j = 0, \cdots, r-1$, are known *a priori*.

However, we are interested in the much more challenging problem in which we do not have prior knowledge of the discontinuities. In order to address this problem we shall let $r = n$,

$\beta \in \mathbb{R}^n$, $t_i = \hat{t}_j$, that is we shall allow one discontinuity for every sample point, and equip the above least-squares problem with an additional *cardinality* constraint. The resulting problem of interest is the following constrained optimization problem

$$\min_{\alpha\in\mathbb{R}_+,\beta\in\mathbb{R}^n_+} \left\{ \frac{1}{2}\sum_{i=0}^{n-1} \left(f_{\alpha\beta}(\hat{t}_i) - \hat{y}_i\right)^2 : \quad t_i = \hat{t}_j, \quad \|\beta\|_0 \le s \le n \right\}, \tag{7.3}$$

in which $s \le n$ limits the number of discontinuities in the sawtooth function. Problem (7.3) is no longer convex due to the cardinality constraint.

Remark 3. *In practice $s \ll n$, because when $s = n$ problem* (7.3) *is trivial. Indeed, if $s = n$, $\hat{y}_0 \ge 0$ and $\alpha = \max\{\max_i(\hat{y}_{i-1} - \hat{y}_i)/(\hat{t}_i - \hat{t}_{i-1}), 0\} \ge 0$ then*

$$\beta_0 = \hat{y}_0 \ge 0, \qquad \beta_i = \hat{y}_i + \alpha(\hat{t}_i - \hat{t}_0) - \sum_{k=0}^{i-1} \beta_i \ge 0, \quad i = 1, \ldots, r-1$$

is feasible[1] *with cost*

$$0 \le J(\alpha,\beta) = \frac{1}{2}\sum_{i=0}^{n-1} \left(f_{\alpha\beta}(\hat{t}_i) - \hat{y}_i\right)^2 = \frac{1}{2}\sum_{i=0}^{n-1} \left(\beta_i - \alpha(\hat{t}_i - \hat{t}_0) - \hat{y}_i\right)^2 = 0$$

so that it must also be optimal. An example of such a solution is presented at the bottom plot in Figure 1.3. Clearly large values of s will lead to over-fitting.

Before proceeding, and for simplicity, we shall further constrain the sample points, $\hat{t}_i$, $i = 0, \cdots, n-1$, to be drawn periodically, that is $\hat{t}_i = \hat{t}_0 + iT_s$, where $T_s > 0$ is the *sampling period*. As we will show later, it seems that this assumption can be safely removed, but doing so leads to distracting technical complications. With that assumption in mind, define matrices $A \in \mathbb{R}^{n\times n+1}$

[1] The proof follows trivially by induction and is omitted for brevity.

and $E \in \mathbb{R}^{n\times n+1}$, and vectors $\boldsymbol{x} \in \mathbb{R}^{n+1}$ and $\boldsymbol{y} \in \mathbb{R}^n$, as follows

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ -1 & 1 & 1 & 0 & \cdots & 0 \\ -2 & 1 & 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1-n & 1 & 1 & 1 & \cdots & 1 \end{bmatrix}, \quad E = \begin{bmatrix} 0 & I \end{bmatrix}, \quad \hat{\boldsymbol{y}} = \begin{pmatrix} \hat{y}_0 \\ \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_{n-1} \end{pmatrix}, \quad \boldsymbol{x} = \begin{pmatrix} \alpha \\ \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{n-1} \end{pmatrix} \tag{7.4}$$

in which I is the $n \times n$ identity matrix so that $E\boldsymbol{x} = \beta$. Using the above notation Problem (7.3) can be rewritten as

$$\min_{\boldsymbol{x}} \quad \left\{ \frac{1}{2} \|A\boldsymbol{x} - \hat{\boldsymbol{y}}\|_2^2 : \quad \|E\boldsymbol{x}\|_0 \leq s, \quad \boldsymbol{x} \geq 0 \right\}. \tag{7.5}$$

It is this compact formulation that will be used in the rest of the paper.

7.4 Main Result and Connections to LASSO Regression

The problem of explicitly imposing a constraint on the *cardinality* of the optimal solution is a difficult problem in general. Perhaps the best, well known heuristic is the LASSO (Least Absolute Shrinkage and Selection Operator) regression [Tib96]. The LASSO heuristic replaces the cardinality constraint in a least-squares regression problem by a 1-norm constraint, as in:

$$\min_{\boldsymbol{x}} \quad \left\{ \frac{1}{2} \|A\boldsymbol{x} - \hat{\boldsymbol{y}}\|_2^2 : \quad \|\boldsymbol{x}\|_1 \leq m \right\} \tag{7.6}$$

where $m \in \mathbb{R}_+$, which is a constrained convex quadratic program. Here the matrix A can have arbitrary structure. The LASSO approach has one goal: to restrict how large the coefficients may grow, and one side-effect, which is to limit how many of the coefficients are non-zero. The inclusion of the 1-norm constraint *promotes* sparsity in $\boldsymbol{x}$. See [Mur12, Tib96] for a more in depth discussion.

Returning to Problem (7.5), for the remainder of this paper we will be concerned with the following *relaxation of the sawtooth minimization problem* (7.5):

$$\min_{\boldsymbol{x}} \quad \left\{ \frac{1}{2} \|A\boldsymbol{x} - \hat{\boldsymbol{y}}\|_2^2 : \quad \|E\boldsymbol{x}\|_1 \leq m \quad \boldsymbol{x} \geq 0 \right\} \tag{7.7}$$

in which A, E, $\boldsymbol{x}$, and $\hat{\boldsymbol{y}}$ are as in (7.4), and the non-convex cardinality constraint has been relaxed into the convex 1-norm constraint. The relaxed problem, (7.7), is a convex program. Note that (7.7) is more constrained than the general LASSO problem, due the the non-negativity constraint on $\boldsymbol{x}$. As in the LASSO problem, the choice of m is non-trivial. Larger values of m provide the minimization room to choose non-zero β_i's which can lead to overfit. As $m \to \infty$, the problem behaves as Problem (7.3) with $s = n$ and the optimal cost approaches 0 (see Remark (3)), similar to results in [SPB08]. Unlike the LASSO problem, the cardinality of the solution to the relaxed Problem (7.7) is a non-decreasing function of the bound m. This result, which is presented in the next Theorem, is the main contribution of the present paper.

Theorem 1. *Let matrices A and E be as in* (7.4) *and consider*

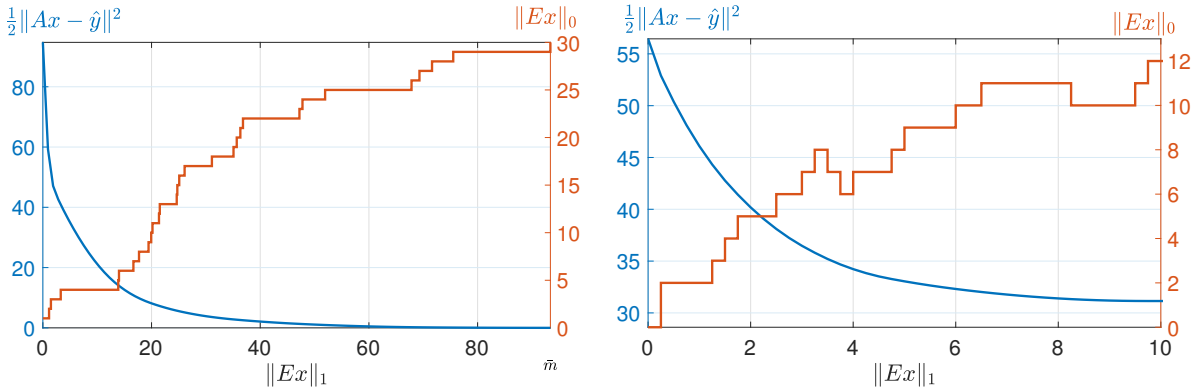
$$\boldsymbol{x}_m^* = \arg\min_{\boldsymbol{x}} \left\{ \frac{1}{2} \|A\boldsymbol{x} - \hat{\boldsymbol{y}}\|_2^2 : \quad \|E\boldsymbol{x}\|_1 \leq m, \quad \boldsymbol{x} \geq 0 \right\}. \tag{7.8}$$

If $\bar{m}$ is the smallest $m > 0$ such that there exists $\boldsymbol{x}_{\bar{m}}^*$ satisfying

$$A\boldsymbol{x}_{\bar{m}}^* = \hat{\boldsymbol{y}}, \quad \|E\boldsymbol{x}_{\bar{m}}^*\|_1 \leq \bar{m}, \quad \boldsymbol{x}_{\bar{m}}^* \geq 0, \tag{7.9}$$

Then $\left\|E\boldsymbol{x}_{m_1}^*\right\|_0 \leq \left\|E\boldsymbol{x}_{m_2}^*\right\|_0$ for any m_1 and m_2 such that $m_1 \leq m_2 < \bar{m}$.

Proof of Theorem 1 will require considerable technical developments. This theorem ensures that one can obtain minimum cardinality solutions by solving convex problems parameterized by a decreasing sequence of scalars. Such a property is rare, arising from special structure inside the particular matrix A from (7.4). In order to illustrate this fact we calculated and plotted



from (7.4) (b) Random A

Figure 7.1: Cost function (left axis) and cardinality (right axis) of the optimal solutions to the relaxation (7.7) for a matrix A with structure as in (7.4) and a randomly generated matrix A. Note the monotonicity of the cardinality present in (a) and absent in (b).

the cost function and the cardinality of the optimal solutions obtained from the relaxation (7.7) for a matrix A with structure as in (7.4) (the one from the numerical example in Section 7.5) and a randomly generated matrix A in Figures 7.1a and 7.1b, respectively.

7.5 Application to Least-Squares Fitting

In this section we will illustrate how Theorem 1 can lead of efficient solutions to the least-squares cardinality constrained Problem 7.3. In order to make the discussion more concrete, assume that the vector of periodic measurements, $\hat{\boldsymbol{y}}$, is such that $\hat{\boldsymbol{y}} = \bar{\boldsymbol{y}} + \boldsymbol{\gamma}$, in which the entries of $\bar{\boldsymbol{y}}$ are such that $y_i = f_{\alpha\beta}(t_0 + iT_s)$, $i = 0, \cdots, n-1$, that is $\bar{\boldsymbol{y}} = A\bar{\boldsymbol{x}}$ for some (unknown) $\bar{\boldsymbol{x}}$, and the entries of the noise vector $\boldsymbol{\gamma}$ are independent random variables γ_i, $i = 0, \cdots, n-1$, with zero mean and known variance σ^2. We seek to obtain a vector close to $\bar{\boldsymbol{x}}$ by solving a series of relaxations in the form of the convex Problem (7.7). Start by calculating the expected value of the function

$$\mathbb{E}\left\{\|A\bar{\boldsymbol{x}} - \hat{\boldsymbol{y}}\|_2^2\right\} = \mathbb{E}\left\{\|\boldsymbol{\gamma}\|_2^2\right\} = n\sigma^2. \tag{7.10}$$

Given a large number of samples, one would therefore expect that the least-squares fitting error would be close to $(1/2)n\sigma^2$. The above discussion suggests the following method for finding the closest least-square fit.

Algorithm 2 (Sawtooth Least-Squares)**.** *Let A, E,* $\boldsymbol{x}$*,* $\hat{\boldsymbol{y}}$ *as in* (7.4)*, and* $\varepsilon > 0$*,* $\sigma > 0$ *be given. Perform the following steps:*

1. *Calculate* $\bar{m} = \min_{\boldsymbol{x}}\{\|E\boldsymbol{x}\|_1 : \quad A\boldsymbol{x} = \hat{\boldsymbol{y}}, \quad \boldsymbol{x} \geq 0\}$.
2. *Calculate* $m_\sigma = \min_{\boldsymbol{x}}\{\|E\boldsymbol{x}\|_1 : \quad \|A\boldsymbol{x} - \hat{\boldsymbol{y}}\|^2 \leq n\sigma^2, \quad \boldsymbol{x} \geq 0\}$.
3. *Determine the largest* $m^* \in [m_\sigma, \bar{m}]$ *such that* $\|E\boldsymbol{x}_{m^*}\|_0 = \|E\boldsymbol{x}_{m_\sigma}\|_0$.
4. *Set* $r = \|E\boldsymbol{x}_{m^*}\|_0$*,* $t = \{i : \boldsymbol{x}_{i+2} > 0, i = 0, \cdots, n\}$*, and solve the least-squares regression Problem 7.2.*

The rationale behind the above algorithm is that Theorem 1 guarantees that m^* and $\boldsymbol{x}_{m^*}$ is the smallest least-squares solution of the relaxation Problem (7.7) with cardinality $\|E\boldsymbol{x}_{m^*}\|_0 = \|E\boldsymbol{x}_{m_\sigma}\|_0$. The optimizations in *Steps 1*, *2* and *4* amount to solving a linear program, a constrained convex quadratic program, and a standard least-squares regression, respectively, all of which can be done efficiently. The search in *Step 3*, thanks to Theorem 1, is also simple because the cardinality of the solution is monotonic in m.

Below we present a numerical example to illustrate the above discussion. MATLAB and Yalmip were used for numerical solutions of all optimization problems [Löf04]. For this example, 31 data points were generated for a sawtooth as in Definition 2 with

$$\alpha = 0.7, \qquad r = 4, \qquad \beta = \{4, 5, 5, 7\} \qquad t = \{0, 8, 14, 23\}$$

and noise was added to each data point drawing from a zero-mean independent Gaussian distribution with variance $\sigma^2 = 0.64$, resulting in the samples already shown in Figure 1.3. Algorithm 2

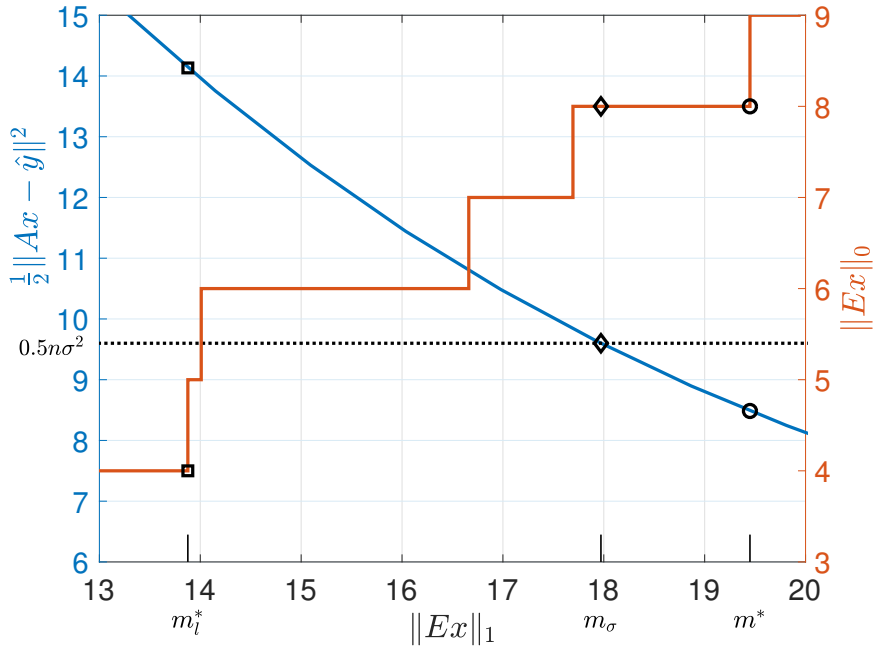


Figure 7.2: Optimal cost (blue) and cardinality (orange) of the relaxation (7.7) as a function of the relaxation parameter $m = \|E\boldsymbol{x}\|_1$. Marks correspond to the points generated by Algorithm 2.

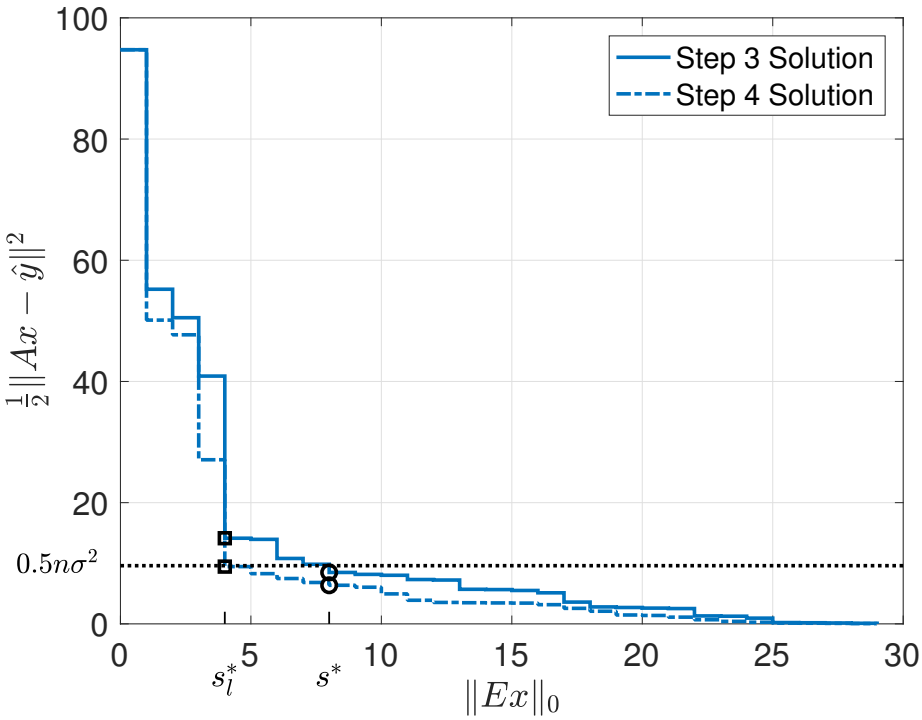


Figure 7.3: Optimal cost and cardinality of the relaxation (7.7) (Algo. 1 step 3) and corresponding RSS (7.2) (Algo. 1 step 4) as a function of the optimal relaxation cardinality $s = \|E\boldsymbol{x}_m\|_0$.

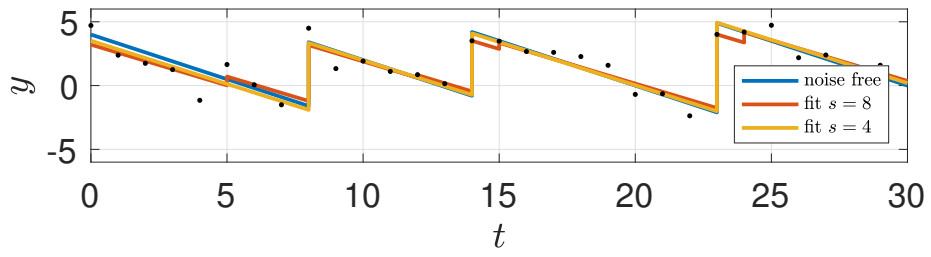


Figure 7.4: Original sawtooth and noisy samples and fits based on the optimal solution of the relaxation (7.7) for $s = 8$ and $s = 4$.

applied to this data set produces

$$\bar{m} \approx 93.4, \qquad m_\sigma \approx 17.97, \qquad m^* = 19.45.$$

The optimal solution to the relaxation (7.7) and the corresponding cardinality is shown in Figure 7.1a. This figure is zoomed in around m_σ in Figure 7.2 to show m_σ (diamond), m^* (circle) and the corresponding cardinality in more detail. The location of the jumps, the cardinality of the solution, and the estimated α, denoted $\hat{\alpha}$, associated with m^* are

$$r = \|E\boldsymbol{x}_{m^*}\|_0 = 9, \qquad t = \{\mathbf{0}, 5, \mathbf{8}, 12, \mathbf{14}, 15, \mathbf{23}, 24, 25\}, \qquad \hat{\alpha} \approx 0.635$$

Note how all jumps present in the original noise-free sawtooth (marked in bold) are still present at the solution. The resulting RSS (7.2) obtained in *Step 4* is about 6.4, which is, as expected, lower than the upper bound guaranteed by the relaxation (7.7), as the algorithm produces

$$\text{RSS} \approx 6.4 \leq \frac{1}{2}\|A\boldsymbol{x}_{m^*} - \hat{y}\| \approx 8.4 \leq \frac{1}{2}\|A\boldsymbol{x}_{m_\sigma} - \hat{y}\| = \frac{1}{2}n\sigma^2 = 9.6.$$

The final fit is shown in Figure 7.4, in which the additional jumps lead, in this case to a small level of *over-fitting*.

One way to reduce this potential over-fitting resulting from the gap between the least-residual and the upper-bound produced by the relaxation is to select m based on the lower bound

produced by solving the regressions (7.2) for $\boldsymbol{x}_m$ for values of $m \leq m_\sigma \leq m^*$ instead of the upper bound given by the relaxation.

Indeed, as suggested by the plot in Figure 7.3 which shows the minimal RSS obtained based on the optimal upper bound solution $\boldsymbol{x}^*_m$ with cardinality s^*, a choice of m that brings the minimal RSS closer to the expected variance in this example is m^*_l with cardinality s^*_l, marked by squares in Figures 7.2 and 7.3. The location of the jumps and the cardinality of the solution associated with m^*_l are

$$r = \|E\boldsymbol{x}_{m^*_l}\|_0 = 4, \qquad t = \{\mathbf{0}, \mathbf{8}, \mathbf{14}, \mathbf{23}\}, \qquad \hat{\alpha} \approx 0.68$$

Note that this adjustment can also be done efficiently because it is based on the optimal solution of the relaxation, $\boldsymbol{x}^*$ and m^*, and not on a full combinatorial search for a given cardinality. This adjusted fit captures exactly the jumps on the original sawtooth.

Finally, note that Figures 7.3 also suggests a simple criterion for selecting the best fit when σ is not known *a priori*. In the spirit of methods like Principal Component Analysis (PCA) [TH09], a selection based on the point at which there is a significant jump on the relaxation fitting cost seems to be a good choice. In this example, such a point also leads to a choice of $m = m^*_l$ and $\|E\boldsymbol{x}_{m^*_l}\|_0 = 4$.

7.6 Proof of Theorem 1

In order to characterize the optimality of the cardinality constrained Problem (7.5), we need to assert certain properties of the relaxed Problem (7.7). First is that the inequality involving m is always binding at the optimal solution.

Lemma 1. *Let $m < \bar{m}$ and $\boldsymbol{x}^*_m$ be as Theorem 1. Then $\|E\boldsymbol{x}^*_m\|_1 = m$.*

Proof. Suppose $\|E\boldsymbol{x}^*_m\|_1 < m < \bar{m}$. In this case, the constraint $\|E\boldsymbol{x}\|_1 \leq m$ in Problem (7.8) is not

active at the optimal solution, therefore

$$\boldsymbol{x}_m^* = \arg\min_{\boldsymbol{x}} \left\{ \frac{1}{2} \|A\boldsymbol{x} - \hat{\boldsymbol{y}}\|_2^2 : \quad \boldsymbol{x} \geq 0 \right\}.$$

However, by the construction in Remark 3, it follows that $A\boldsymbol{x}_m^* = \hat{\boldsymbol{y}}$. But since $\bar{m}$ is the smallest m such that there exists $\boldsymbol{x}$ with $A\boldsymbol{x} = \hat{\boldsymbol{y}}$ and $\boldsymbol{x} \geq 0$, then $m \geq \bar{m} > m$, a contradiction. Therefore, $\|E\boldsymbol{x}_m^*\|_1 = m$. □

With Lemma (1) in mind, whenever $m \leq \bar{m}$ we can rewrite (7.8) as

$$\boldsymbol{x}_m^* = \arg\min_{\boldsymbol{x}} \left\{ \frac{1}{2} \|A\boldsymbol{x} - \boldsymbol{y}\|_2^2 : \quad \|E\boldsymbol{x}\|_1 = m, \quad \boldsymbol{x} \geq 0 \right\} \tag{7.11}$$

in which an equality replaces the inequality $\|E\boldsymbol{x}\|_1 \leq m$. now consider perturbations to the solutions to Problem (7.11) in the form

$$\boldsymbol{x}_\varepsilon^* = \arg\min_{\boldsymbol{x}} \left\{ \frac{1}{2} \|A\boldsymbol{x} - \boldsymbol{y}\|_2^2 : \quad \|E\boldsymbol{x}\|_1 = m - \varepsilon, \quad \boldsymbol{x} \geq 0, \quad \varepsilon > 0 \right\}. \tag{7.12}$$

Define the set of *active positive constraints* at $\boldsymbol{x}_m^*$ for a given m as

$$\mathcal{A}(\boldsymbol{x}_m^*) = \{ i : \quad (\boldsymbol{x}_m^*)_i = 0 \}, \tag{7.13}$$

in which $(\boldsymbol{x}_m^*)_i$ denotes the ith element of vector $\boldsymbol{x}_m^*$. Our next goal is to prove that $\|E\boldsymbol{x}_\varepsilon^*\|_0 \leq \|E\boldsymbol{x}_m^*\|_0$ for all small enough $\varepsilon > 0$. We will do so by proving that for all $0 < \varepsilon \leq \bar{\varepsilon}$ where

$$\bar{\varepsilon} = \max \{ \varepsilon > 0 : \quad \mathcal{A}(\boldsymbol{x}_\varepsilon^*) = \mathcal{A}(\boldsymbol{x}_m^*) \}. \tag{7.14}$$

it is true that

$$\delta \boldsymbol{x}^* = \boldsymbol{x}_\varepsilon^* - \boldsymbol{x}_m^* \leq 0. \tag{7.15}$$

Indeed, (7.15) implies that entries of $\boldsymbol{x}_\varepsilon$ must always be less than or equal to the corresponding entries of $\boldsymbol{x}_m^*$ and it follows that if $(\boldsymbol{x}_m^*)_i = 0$ for some i, then $(\boldsymbol{x}_\varepsilon^*)_i = 0$ as well, proving that $\|E\boldsymbol{x}_\varepsilon\|_0 \leq \|E\boldsymbol{x}_m^*\|_0$, for all $0 < \varepsilon \leq \bar{\varepsilon}$.

We shall now invoke the KKT conditions of optimality [BV04] for Problems (7.11) and (7.12) as an intermediate step in the proof of property (7.15). The next lemma provides for an explicit calculation of $\delta \boldsymbol{x}^*$ through a set of linear equations derived from the conditions of optimality.

Lemma 2. *Let A be as in* (7.4)*, $\boldsymbol{x}_m^*$, $\boldsymbol{x}_\varepsilon^*$ as in* (7.11) *and* (7.12)*, the active set $\mathcal{A}(\boldsymbol{x}_m^*)$ be as in* (7.13)*, and a positive $\bar{\varepsilon} > 0$ as in* (7.14)*. Assemble the matrices*

$$F_m^T = \begin{bmatrix} \boldsymbol{e}_i \end{bmatrix}, \quad i \in \mathcal{A}(\boldsymbol{x}_m^*), \qquad G_m = \begin{bmatrix} \boldsymbol{e}_i \end{bmatrix}, \quad i \notin \mathcal{A}(\boldsymbol{x}_m^*), \qquad F_m G_m = 0,$$

where $\boldsymbol{e}_i$ is the ith column of the $n+2 \times n+2$. All solutions to the equation

$$\begin{bmatrix} G_m^T A^T A G_m & G_m^T \boldsymbol{c} \\ \boldsymbol{c}^T G_m & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{w} \\ v \end{pmatrix} = \begin{pmatrix} \boldsymbol{0} \\ -\varepsilon \end{pmatrix},$$

in which

$$\boldsymbol{c}^T = \begin{pmatrix} 0 & 1 & 1 & \cdots & 1 \end{pmatrix} \in \mathbb{R}^{n+1} \tag{7.16}$$

are such that $\boldsymbol{w} \leq 0$ for all $0 < \varepsilon < \bar{\varepsilon}$ and $\delta \boldsymbol{x}^* = \boldsymbol{x}_\varepsilon^* - \boldsymbol{x}_m^* = G_m \boldsymbol{w} \leq \boldsymbol{0}$.

Proof. Let $\boldsymbol{x}_m^*$, $\lambda_m^* \in \mathbb{R}$, and $\boldsymbol{z}_m^* \in \mathbb{R}^{n+1}$ be the optimal primal and dual solutions satisfying the

KKT conditions associated with Problem (7.11), that is

$$\begin{bmatrix} A^TA & \boldsymbol{c} & -I \\ \boldsymbol{c}^T & 0 & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{x}_m^* \\ \lambda_m^* \\ \boldsymbol{z}_m^* \end{pmatrix} = \begin{pmatrix} A^T\boldsymbol{y} \\ m \end{pmatrix}, \quad \boldsymbol{x}_m^* \geq 0, \quad \boldsymbol{z}_m^* \geq 0, \quad \boldsymbol{z}_m^* \odot \boldsymbol{x}_m^* = 0, \tag{7.17}$$

where $\odot$ denotes entrywise product. Since this is a standard convex quadratic program the KKT conditions are necessary and sufficient for optimality when *Slater's* constraint qualification is satisfied [BV04], which is trivially satisfied by vector $\boldsymbol{x}$ such that $x_i = m/(n-1)$ for $i = 2, ..., n$.

The values of $\boldsymbol{x}_m^*$, λ_m^*, and the nonzero entries of $\boldsymbol{z}_m^*$, denoted $\bar{\boldsymbol{z}}_m^*$, satisfying (7.17) can be equivalently calculated by solving the problem

$$\begin{bmatrix} A^TA & \boldsymbol{c} & -F_m^T \\ \boldsymbol{c}^T & 0 & 0 \\ F_m & 0 & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{x}_m^* \\ \lambda_m^* \\ \bar{\boldsymbol{z}}_m^* \end{pmatrix} = \begin{pmatrix} A^T\boldsymbol{y} \\ m \\ \boldsymbol{0} \end{pmatrix}. \tag{7.18}$$

Because F_m is full row-rank and Lemma 4, the above set of equations is nonsingular, meaning that a solution $(\boldsymbol{x}_m^*, \lambda_m^*, \bar{\boldsymbol{z}}_m^*)$ exists and is unique.

Similarly, optimal solutions to problem (7.12) satisfy

$$\begin{bmatrix} A^TA & \boldsymbol{c} & -I^T \\ \boldsymbol{c}^T & 0 & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{x} \\ \lambda \\ \boldsymbol{z} \end{pmatrix} = \begin{pmatrix} A^T\boldsymbol{y} \\ m - \varepsilon \end{pmatrix}, \quad \boldsymbol{x} \geq 0, \quad \boldsymbol{z} \geq 0, \quad \boldsymbol{z} \odot \boldsymbol{x} = 0, \tag{7.19}$$

and, for all $0 < \varepsilon \leq \bar{\varepsilon}$, also the reduce linear algebra problem

$$\begin{bmatrix} A^TA & \boldsymbol{c} & -F_m^T \\ \boldsymbol{c}^T & 0 & 0 \\ F_m & 0 & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{x}_\varepsilon^* \\ \lambda_\varepsilon^* \\ \bar{\boldsymbol{z}}_\varepsilon^* \end{pmatrix} = \begin{pmatrix} A^T\boldsymbol{y} \\ m-\varepsilon \\ \boldsymbol{0} \end{pmatrix}, \tag{7.20}$$

where $\bar{\boldsymbol{z}}_\varepsilon^*$ denotes a vector with the nonzero entries of $\boldsymbol{z}_\varepsilon^*$, since $\boldsymbol{x}_m^*$ and $\boldsymbol{x}_\varepsilon^*$ share the same associated active set.

From the above discussion, it follows that $\delta\boldsymbol{x}^* = \boldsymbol{x}_\varepsilon^* - \boldsymbol{x}_m^*$, $\delta\lambda^* = \lambda_\varepsilon^* - \lambda_m^*$, and $\delta\boldsymbol{z}^* = \boldsymbol{z}_\varepsilon^* - \boldsymbol{z}_m^*$, must satisfy

$$\begin{bmatrix} A^TA & \boldsymbol{c} & -F_m^T \\ \boldsymbol{c}^T & 0 & 0 \\ F_m & 0 & 0 \end{bmatrix} \begin{pmatrix} \delta\boldsymbol{x}^* \\ \delta\lambda^* \\ \delta\bar{\boldsymbol{z}}^* \end{pmatrix} = \begin{pmatrix} \boldsymbol{0} \\ -\varepsilon \\ \boldsymbol{0} \end{pmatrix}, \tag{7.21}$$

where $\delta\bar{\boldsymbol{z}}^*$ denotes the nonzero entries of $\delta\boldsymbol{z}^*$. Because $F_m\delta\boldsymbol{x}^* = \boldsymbol{0}$, one has that $\delta\boldsymbol{x}^* = G_m\boldsymbol{w}$. Upon pre-multiplication of the first two block-equations of (7.21) by the full-rank matrix $\begin{bmatrix} G_m^T & \boldsymbol{0} \\ \boldsymbol{0} & 1 \end{bmatrix}$, the value of $\boldsymbol{w}$ and $\delta\lambda^*$ can be determined by solving the reduced linear algebra problem [GMW19]

$$\begin{bmatrix} G_m^TA^TAG_m & G_m^T\boldsymbol{c} \\ \boldsymbol{c}^TG_m & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{w} \\ \delta\lambda^* \end{pmatrix} = \begin{pmatrix} \boldsymbol{0} \\ -\varepsilon \end{pmatrix}$$

Then, by Lemma 17, $\boldsymbol{w} \leq 0$, and the fact that G_m is a partition of the identity matrix, it follows that $\delta\boldsymbol{x}^* = G_m\boldsymbol{w} \leq 0$ as claimed. □

Because the constraint set in Problem (7.11) is polyhedral, there is only a finite set of active sets (7.13) for all $m \in [0, \bar{m}]$, each active set spanning a finite sub-interval in $[0, \bar{m}]$. Therefore, one can apply Lemma 2 and inequality (7.15) repeatedly on all active sets from $m_2 < \bar{m}$ towards

$m_1 \leq m_2$ to conclude that $\|E\boldsymbol{x}^*_{m_1}\|_0 \leq \|E\boldsymbol{x}^*_{m_2}\|_0$, thus proving Theorem 1.

7.7 Technical Linear Algebra Results

In this section we develop the additional technical results necessary for the proof of Lemma 17, which is the final step in completing the proof of Theorem 1. The first couple of lemmas are concerned with the structure and rank of some key matrices.

Lemma 3. *Let A be defined as in* (7.4)*, then* $\text{rank}(A) = \text{rank}(A^TA) = n$.

Proof. Since A has n rows, $\text{rank}(A) \leq n$. By inspection, the set of columns 2 through $n+1$ form a linearly independent set and $\text{rank}(A) \geq n$. Therefore $\text{rank}(A) = \text{rank}(A^TA) = n$. □

Lemma 4. *Let A be as in* (7.4) *and $\boldsymbol{c}$ as in* (7.16) *and define*

$$W = \begin{bmatrix} A^TA & \boldsymbol{c} \\ \boldsymbol{c}^T & 0 \end{bmatrix} \in \mathbb{R}^{n+2\times n+2}. \tag{7.22}$$

The rank of W is $n+2$, that is W is full rank.

Proof. Let

$$U = \begin{bmatrix} \mathbf{0} & \boldsymbol{c} \\ \boldsymbol{c}^T & 0 \end{bmatrix}, \qquad V = \begin{bmatrix} A^TA & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix}, \qquad W = U + V.$$

$\text{rank}(U) = 2$ and, by Lemma 3, $\text{rank}(V) = n$. Furthermore, one can show that $R(U) \cap R(V) = \{\mathbf{0}\}$, where $R(X)$ denotes the range space of X. Therefore, by the *rank-sum theorem* [HJ13], $\text{rank}(U+V) = \text{rank}(U) + \text{rank}(V) = n+2$. □

Lemma 5. *Let W be as in* (7.22)*. Define $T_{n+2} \in \mathbb{R}^{n+2\times n+2}$, $\boldsymbol{v} \in \mathbb{R}^{n+1}$, $D \in \mathbb{R}^{n+1\times n+1}$, and*

$V \in \mathbb{R}^{n+2\times n+2}$, *as in*

$$T_{n+2} = \begin{bmatrix} -1 & 0 & 0 & \cdots & 0 & 0 \\ -1 & 1 & 0 & \cdots & 0 & 0 \\ -1 & 1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ -1 & 1 & 1 & \cdots & 1 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix}, \quad \boldsymbol{v} = \begin{pmatrix} -1 \\ 0 \\ 1 \\ \vdots \\ n-2 \\ 1-n \end{pmatrix}, \quad D = \begin{bmatrix} I & \mathbf{0} \\ \mathbf{0} & -1 \end{bmatrix}, \quad V = \begin{bmatrix} \alpha & \boldsymbol{v}^T \\ \boldsymbol{v} & D \end{bmatrix}, \tag{7.23}$$

Then

$$W = T_{n+2}^T V T_{n+2}. \tag{7.24}$$

If $n \geq 3$ *then* $\alpha = n(2n^2 - 4n + 3)/6 \geq 2$.

Proof. One can verify the particular structure of the matrices in (7.23). Regarding the value of α, the first entry of $T_{n+2}^T V T_{n+2}$ is

$$(T_{n+2}^T V T_{n+2})_{1,1} = \alpha + n + 2\sum_{i=2}^{n-2} i = \alpha + n(n-2).$$

Likewise, the first entry of $A^T A$ is

$$(A^T A)_{1,1} = \sum_{i=1}^{n-1} i^2 = \frac{(n-1)n(2(n-1)+1)}{6}.$$

Equating the two expressions one obtains

$$\alpha = \frac{(n-1)n(2(n-1)+1)}{6} - n(n-2) = n(2n^2 - 4n + 3)/6$$

which, for $n \geq 3$, implies that $\alpha \geq 2$. □

Lemma 6. *Let $n \geq 3$. The matrices T_{n+2} and V defined in* (7.23) *are nonsingular and have as inverses*

$$T_{n+2}^{-1} = \begin{bmatrix} -1 & 0 & 0 & \cdots & 0 & 0 \\ -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & -1 & 1 & -1 \\ 0 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix}, \quad V^{-1} = \begin{bmatrix} \beta^{-1} & -\beta^{-1}\boldsymbol{v}^T D^{-1} \\ -\beta^{-1} D^{-1}\boldsymbol{v} & D^{-1} + \beta^{-1} D^{-1}\boldsymbol{v}\boldsymbol{v}^T D^{-1} \end{bmatrix},$$

in which $\beta = \alpha - \boldsymbol{v}^T D^{-1}\boldsymbol{v} = (n-1)^2 \geq 4$.

Proof. The inverse of T_{n+2} can be directly calculated from (7.23). Now, define $\beta = \alpha - \boldsymbol{v}^T D^{-1}\boldsymbol{v}$. Since $D^{-1} = D$, it follows that

$$\boldsymbol{v}^T D^{-1}\boldsymbol{v} = \boldsymbol{v}^T D\,\boldsymbol{v} = \sum_{i=1}^{n} v_i^2 - v_{n+1}^2 = 1 + \sum_{i=1}^{n-2} i^2 - (n-1)^2$$

and

$$\begin{aligned} \beta = \alpha - \boldsymbol{v}^T D^{-1}\boldsymbol{v} &= \sum_{i=1}^{n-1} i^2 - n(n-2) - \left(1 + \sum_{i=1}^{n-2} i^2 - (n-1)^2\right) \\ &= 2(n-1)^2 - n(n-2) - 1 = (n-1)^2 > 0 \end{aligned}$$

which, for $n \geq 3$, implies that $\beta \geq 4 > 0$. Since D is nonsingular and the Schur complement of V is $\beta = \alpha - \boldsymbol{v}^T D^{-1}\boldsymbol{v} > 0$, V is nonsingular. □

Definition 4. *Let $J \subset \{2, \ldots, n+1\}$ and define the matrix*

$$G = \begin{bmatrix} \boldsymbol{e}_i \end{bmatrix}, \quad i \notin J, \qquad\qquad G \in \mathbb{R}^{n+1 \times \kappa_J}, \tag{7.25}$$

where $\kappa_J = n+1-|J|$*, and* $|J|$ *is the cardinality of the set* J*.*

Lemma 7. *Let* $n \geq 3$*,* $G \in \mathbb{R}^{n+1\times\kappa_J}$ *be as in Definition 4, and* $\bar{G} = \mathrm{diag}(G,1)$*. The matrix* $T_G = \bar{G}^T T_{n+2}\bar{G} \in \mathbb{R}^{\kappa_J+1\times\kappa_J+1}$ *is nonsingular and there exists a matrix* $H_G \in \mathbb{R}^{n+2\times\kappa_J+1}$ *such that*

$$T_{n+2}\,\bar{G} = H_G T_G \qquad \textit{with,} \qquad H_G = T_{n+2}\,\bar{G}\,T_G^{-1}. \tag{7.26}$$

Furthermore, when $n+1 \notin J$*, then*

$$T_G = T_{\kappa_J+1}. \tag{7.27}$$

When $n+1 \in J$ *then*

$$T_G = T_{\kappa_J+1} - e_{\kappa_J} e_{\kappa_J+1}^T, \qquad T_G^{-1} = T_{\kappa_J+1}^{-1} + e_{\kappa_J} e_{\kappa_J+1}^T. \tag{7.28}$$

Proof. Existence of H_G and T_G as in (7.26) follows from the fact that G and $\bar{G}$ are full rank. Hence, T_G is nonsingular. Property (7.27) follows from the fact that G is a partition of the identity matrix and the product $\bar{G}^T T \bar{G}$ simultaneously eliminates corresponding columns and rows of T. The inverse property (7.28) in the case $n+1 \in J$ follows from the block-triangular structure of T_{κ_J+1}. □

Using Lemma 7, it will be possible to factor the matrix

$$W_G = \begin{bmatrix} G^T A^T A G & G^T \boldsymbol{c} \\ \boldsymbol{c}^T G & 0 \end{bmatrix} = \bar{G}^T \left(T^T V T\right) \bar{G}$$

in the form

$$W_G = T_G^T V_G T_G, \qquad V_G = H_G^T V H_G. \tag{7.29}$$

Despite the heavy notation required to express the above relationships, the resulting matrices H_G and V_G can be simply calculated by inspection of the index set J. We illustrate such calculations in the next subsections.

7.7.1 The construction of H_G

For simplicity of notation, and when no confusion is possible, we will let $H = H_G$. Recall that the set J, from Definition 4, is the set of indices marking the columns to be removed by multiplication by G. Let j_i be the ith element in J so that $1 < j_1 < j_2 < \cdots < j_{k_J}$.

Algorithm 3. *Let $H^{(0)} = I_{n+2}$, $j_0 = 0$, and $i = 1$, and follow the steps:*

1. *Let $H^{(i)}$ be equal to $H^{(i-1)}$ with the column j_i removed and set column $j_i - 1$ in $H^{(i)}$ to be equal to the sum of columns $j_i - 1$ and j_i from $H^{(i-1)}$.*

2. *Subtract one from each $j_i \in J$, $i \geq i - 1$.*

3. *If $i < k_J$, let $i \leftarrow i + 1$ and go to Step 1. If $i = k_J$ and $j_{k_J} \neq n + 1$, go to Step 4. Otherwise add $\boldsymbol{e}_{n+1}$ to the last column of $H^{(k_J)}$ and go to Step 4.*

4. *Let $H = H^{(k_J)}$ and stop.*

The above algorithm applied to the following example in which $n = 6$ and $J = \{2, 3, 6\}$,

$k_J = 3$, leads to $H^{(0)} = I_8$ and:

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{H^{(1)}} \rightarrow \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{H^{(2)}} \rightarrow \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{H=H^{(3)}}$$

The above algorithm is well defined and works even when $n+1 \in J$. In fact, the reason for a different choice of T_G in (7.28) is precisely so that the corresponding H_G could be computed by the above algorithm in all cases. For example, when $n = 6$ and $J_2 = \{2,3,7\}$, the resulting H is given by

$$H^T = (H^{(3)})^T = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

note the presence of the nonzero element at $n+1 \times \kappa_J + 1$, which occurs due to the additional term in T_G in the case $n+1 \in J$.

Lemma 8. *Let H be constructed as in Algorithm 2 and*

$$\bar{J} = \{1, \ldots, n+3\} - J.$$

Then H has full column rank, $|\bar{J}| = n+3-|J| = \kappa_J + 2$, and

$$\theta_i = \|(H)_i\|_1 = \bar{J}_{i+1} - \bar{J}_i \geq 1, \quad i = 1, \ldots, \kappa_J,$$

with,

$$\theta_{\kappa_J+1} = \begin{cases} \|(H)_{\kappa_J+1}\|_1 = 1 & n+1 \notin J \\ \|(H)_{\kappa_J+1}\|_1 = 2 & n+1 \in J \end{cases}$$

Furthermore

$$(\theta_1 - 1)(\theta_1 - 3) \geq \begin{cases} 0, & 2 \notin J, \\ -1, & 2 \in J \end{cases}$$

Proof. That H has full column rank follows from Algorithm 2. Since both J and $\bar{J}$ are ordered in increasing order, $\bar{J}_{i+1} - \bar{J}_i \geq 1$. Similarly, θ_{κ_J+1} is found from Algorithm 2. By construction $\bar{J}_1 = 1$.

Suppose $2 \notin J$. Then, $\bar{J} = \{1, 2, \star \cdots \star\}$ so that $\theta_1 = \bar{J}_2 - \bar{J}_1 \geq 1$

Suppose $2 \in J$, then, $\bar{J} = \{1, q, \star \cdots \star\}$ and $q \geq 3$, then $\theta_1 = \bar{J}_2 - \bar{J}_1 \geq 2$ from which the result follows upon substitution into $(\theta_1 - 1)(\theta_1 - 3)$. □

For example, let $n = 6$, then if $n+1 \notin J$,

$$J = \{2,3,6\}, \qquad \bar{J} = \{1,4,5,7,8,9\}, \qquad \theta = \Big(3,1,2,1,1\Big)$$

otherwise, if $n+1 \in J$

$$J = \{2,3,7\}, \qquad \bar{J} = \{1,4,5,6,8,9\}, \qquad \theta = \Big(3,1,1,2,2\Big).$$

7.7.2 The construction of $V_G = H_G^T V H_G$

A key observation is right multiplication of a matrix by the H_G generated in Algorithm 2 corresponds to the application of Algorithm 2 to the columns of that matrix. Likewise, left multiplication by H_G^T applies Algorithm 2 to the rows of the matrix. Let $V_G = H_G^T V H_G$.

For the examples above, when $n = 6$, $J = \{2,3,6\}$ or $J = \{2,3,7\}$, we have, respectively,

$$V_G = \begin{bmatrix} 31 & 1 & 5 & 4 & -5 \\ 1 & 1 & 0 & 0 & 0 \\ 5 & 0 & 2 & 0 & 0 \\ 4 & 0 & 0 & 1 & 0 \\ -5 & 0 & 0 & 0 & -1 \end{bmatrix} \qquad \text{or,} \qquad V_G = \begin{bmatrix} 31 & 1 & 2 & 7 & -1 \\ 1 & 1 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 \\ 7 & 0 & 0 & 2 & 1 \\ -1 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

It will be at times also convenient to partition H_G as done before with V in (7.23) in the form

$$H_G = \begin{bmatrix} 1 & \mathbf{0} \\ \boldsymbol{h}_1 & U \end{bmatrix}, \tag{7.30}$$

so that

$$V_G = \begin{bmatrix} 1 & \boldsymbol{h}_1^T \\ \mathbf{0} & U^T \end{bmatrix} \begin{bmatrix} \alpha & \boldsymbol{v}^T \\ \boldsymbol{v} & D \end{bmatrix} \begin{bmatrix} 1 & \mathbf{0} \\ \boldsymbol{h}_1 & U \end{bmatrix} = \begin{bmatrix} \alpha + \boldsymbol{h}_1^T(2\boldsymbol{v} + D\boldsymbol{h}_1) & (\boldsymbol{v}^T + \boldsymbol{h}_1^T D)U \\ U^T(\boldsymbol{v} + D\boldsymbol{h}_1) & U^T D U \end{bmatrix}.$$

note that $U^T D U$ is square with dimension $\kappa_J - 1$ and that $\boldsymbol{h}_1 \neq 0$ only if $2 \in J$, in which case

$$\boldsymbol{h}_1 = \begin{bmatrix} 1 & \cdots & 1 & 0 & \cdots & 0 \end{bmatrix},$$

with the first '0' at location $\bar{J}_2$. This means that

$$U^T D\boldsymbol{h}_1 = 0, \qquad D\boldsymbol{h}_1 = \boldsymbol{h}_1, \qquad \boldsymbol{h}_1^T \boldsymbol{h}_1 = \theta_1 - 1, \qquad \boldsymbol{h}_1^T \boldsymbol{v} = \frac{1}{2}(\theta_1 - 4)(\theta_1 - 1)$$

and

$$V_G = \begin{bmatrix} \bar{\alpha} & \bar{\boldsymbol{v}}^T \\ \bar{\boldsymbol{v}} & \bar{D} \end{bmatrix} = \begin{bmatrix} \alpha + (\theta_1 - 1)(\theta_1 - 3) & \boldsymbol{v}^T U \\ U^T \boldsymbol{v} & U^T D U \end{bmatrix}. \tag{7.31}$$

7.7.3 The construction of $\bar{D}$

Lemma 9. *Let $\bar{D}$ be as in* (7.31)*. Then*

$$\bar{D} = Q - 2e_{\kappa_J} e_{\kappa_J}^T. \tag{7.32}$$

with $Q = U^T U \succ 0$ and $Q \in \mathbb{R}^{\kappa_J \times \kappa_J}$. Furthermore, partition $Q = \operatorname{diag}(Q_1, Q_2)$, in which Q_1 is the top diagonal submatrix of dimension $\kappa_J - 2$ in Q and Q_2 is the lower 2×2 matrix in Q. Then

$$Q_1 = \operatorname{diag}(\theta_2, \ldots, \theta_{\kappa_J - 1}) \tag{7.33}$$

and

$$Q_2 = \begin{cases} I_2, & n+1 \notin J, \\ \begin{bmatrix} \theta_{\kappa_J} & 1 \\ 1 & 2 \end{bmatrix}, & n+1 \in J. \end{cases} \tag{7.34}$$

where θ is as in Lemma 8.

Proof. Follows from the particular structure of H and its properties from Lemma 8 and the fact that when $n+1 \notin J$, the operation $U^T D U$ is a sum of select diagonal elements of the identity

matrix inside D, which is what the θ_i's represent. The modification for the case $n+1 \in J$ is necessary because in that case the sum involves the '-1' element at the last entry of D. $\square$

Lemma 10. *Let $\bar{\boldsymbol{v}}$ be as in* (7.31)*, U and $\bar{D}$ as in* (7.30)*. Define $\boldsymbol{u} = \bar{D}^{-1}\bar{\boldsymbol{v}}$. Then*

$$\begin{aligned} \bar{v}_{i-1} &= \frac{1}{2}(\bar{J}_{i+1} - \bar{J}_i)(\bar{J}_{i+1} + \bar{J}_i - 7) \geq -1, && i = 2, \ldots, \kappa_J \\ u_{i-1} &= \frac{\bar{v}_{i-1}}{\theta_i} = \frac{1}{2}(\bar{J}_{i+1} + \bar{J}_i - 7) && i = 2, \ldots, \kappa_J - 1 \end{aligned}$$

with,

$$\begin{aligned} \bar{v}_{\kappa_J} &= \begin{cases} -(n-1), & n+1 \notin J \\ -1, & n+1 \in J \end{cases} \\ u_{\kappa_J - 1} &= \begin{cases} n-2, & n+1 \notin J \\ -1, & n+1 \in J \end{cases} \\ u_{\kappa_J} &= \begin{cases} n-1, & n+1 \notin J \\ \frac{1}{2}(\bar{J}_{\kappa_J+1} - \bar{J}_{\kappa_J})(\bar{J}_{\kappa_J+1} + \bar{J}_{\kappa_J} - 5), & n+1 \in J \end{cases} \end{aligned}$$

Furthermore,

$$u_i \geq 1 + u_{i-1} \qquad\qquad i = 2, \ldots, \nu,$$

where $\nu = \kappa_J$ if $n+1 \notin J$ or $\nu = \kappa_J - 2$ if $n+1 \in J$ and

$$\bar{v}_{\kappa_J - 1} \geq 2n - 5$$

when $n+1 \in J$.

Proof. Start by expanding $\bar{\boldsymbol{v}} = U^T \boldsymbol{v}$ for $i = 2, \ldots, \kappa_J$ to obtain

$$\begin{aligned}
\bar{v}_{i-1} &= \sum_{k=\bar{J}_i}^{\bar{J}_{i+1}-1} v_{k-1} = \sum_{k=\bar{J}_i}^{\bar{J}_{i+1}-1} k-3 = \sum_{k=\bar{J}_i}^{\bar{J}_{i+1}-1} k - 3 \sum_{k=\bar{J}_i}^{\bar{J}_{i+1}-1} 1 \\
&= \frac{\bar{J}_{i+1}^2 - \bar{J}_{i+1} - \bar{J}_i^2 + \bar{J}_i}{2} - 3(\bar{J}_{i+1} - \bar{J}_i) \\
&= \frac{1}{2}(\bar{J}_{i+1} - \bar{J}_i)(\bar{J}_i + \bar{J}_{i+1} - 7)
\end{aligned}$$

By Lemma 8, $\bar{J}_2 \geq 2$ and $\bar{J}_{i+1} - \bar{J}_i \geq 1$ so that $\bar{J}_3 \geq 3$. Therefore, we find that $\bar{J}_i + \bar{J}_{i+1} - 7 \geq -2$ and the result follows. now let $\boldsymbol{u} = \bar{D}_1^{-1} \bar{\boldsymbol{v}}$ such that

$$u_{i-1} = \frac{\bar{v}_{i-1}}{\theta_i}, \qquad\qquad i = 2, \ldots, \kappa_J - 1$$

use Lemma 8 to write

$$u_{i-1} = \frac{\bar{v}_{i-1}}{\bar{J}_{i+1} - \bar{J}_i} = \frac{1}{2}\left(\bar{J}_{i+1} + \bar{J}_i - 7\right).$$

Therefore,

$$u_i - u_{i-1} = \frac{1}{2}\left(\bar{J}_{i+2} + \bar{J}_{i+1} - 7\right) - \frac{1}{2}\left(\bar{J}_{i+1} + \bar{J}_i - 7\right) = \frac{1}{2}\left(\bar{J}_{i+2} - \bar{J}_i\right) \geq 1$$

and by construction of $\bar{J}$, $\bar{J}_{i+1} \geq \bar{J}_i + 1$ and the result follows. note that the special cases identified follow immediately from (7.23) and Lemma 9.

Let $n+1 \in J$, then $\bar{J}_{\kappa_J} \leq n$. Consider $\bar{v}_{\kappa_J - 1}$,

$$\begin{aligned}
\bar{v}_{\kappa_J - 1} &= \frac{1}{2}(\bar{J}_{i+1} - \bar{J}_i)(\bar{J}_{i+1} + \bar{J}_i - 7) \\
&\geq \frac{1}{2}(n+2-n)(n+2+n-7) \\
&= 2n - 5
\end{aligned}$$

completing the proof. □

We need the elementary calculation given in the next lemma before proceeding,

Lemma 11. *Let $p,q \in \mathbb{N}$ with $q \geq 1$. Then,*

$$\frac{1}{q+1}\left(\sum_{i=p}^{p+q} i\right)^2 \leq \sum_{i=p}^{p+q} i^2 - \frac{q}{2}$$

Proof. We have,

$$\begin{aligned}\frac{1}{q+1}\left(\sum_{i=p}^{p+q} i\right)^2 &= \frac{1}{q+1}\left((q+1)p + \sum_{i=0}^{q} i\right)^2 \\ &= (q+1)p^2 + 2p\sum_{i=0}^{q} i + \frac{1}{q+1}\left(\sum_{i=0}^{q} i\right)^2\end{aligned}$$

and,

$$\sum_{i=p}^{p+q} i^2 = (q+1)p^2 + 2p\sum_{i=0}^{q} i + \sum_{i=0}^{q} i^2$$

Therefore, all that needs to be shown is that $0 \leq \sum_{i=0}^{q} i^2 - \frac{q}{2} - \frac{1}{q+1}\left(\sum_{i=0}^{q} i\right)^2$.

$$\begin{aligned}\sum_{i=0}^{q} i^2 - \frac{q}{2} - \frac{1}{q+1}\left(\sum_{i=0}^{q} i\right)^2 &= \frac{q(q+1)(2q+1) - 3q}{6} - \frac{q^2(q+1)}{4} \\ &= \frac{q}{12}(q^2 + 3q - 4)\end{aligned}$$

and has roots $q = -4$ and $q = 1$. For $1 \leq q$, $\frac{q}{12}(q^2+3q-4) \geq 0$. □

Lemma 12. *Let $\boldsymbol{v}$ be as in* (7.23) *and $\bar{\boldsymbol{v}}$ be as in* (7.31)*. For $i \in \mathbb{Z}$ such that $2 \leq i \leq \kappa_J$,*

$$\frac{\bar{v}_{i-1}^2}{\theta_i} \leq \sum_{k=\bar{J}_i}^{\bar{J}_{i+1}} v_k^2$$

Proof. By construction in Lemma 8, $\bar{J}$ is a set of increasing integers, $1 \leq \bar{J}_1$, and $\bar{J}_i < \bar{J}_{i+1}$ for $i = 1, 2, ..., \kappa_J$. Recall, $\bar{v}_{i-1} = \sum_{k=\bar{J}_i}^{\bar{J}_{i+1}-1} v_{k-1} = \sum_{k=\bar{J}_i}^{\bar{J}_{i+1}-1} k-2$. Then, by Lemma 11 with $p = \bar{J}_i - 2$, $q = J_{i+1} - J_i - 1$ and Lemma 8, we have for any $i = 2, ..., \kappa_J$,

$$\begin{aligned}\frac{\bar{v}_{i-1}^2}{\theta_i} &= \frac{1}{\bar{J}_{i+1} - \bar{J}_i}\left(\sum_{k=\bar{J}_i}^{\bar{J}_{i+1}-1} v_{k-1}\right)^2 = \frac{1}{\bar{J}_{i+1} - \bar{J}_i}\left(\sum_{k=\bar{J}_i}^{\bar{J}_{i+1}-1} k-2\right)^2 \\ &\leq \sum_{k=\bar{J}_i}^{\bar{J}_{i+1}-1} (k-2)^2 - \frac{\bar{J}_{i+1} - \bar{J}_i - 1}{2} \leq \sum_{k=\bar{J}_i}^{\bar{J}_{i+1}-1} v_{k-1}^2\end{aligned}$$

Lemma 12 shows the squared components of $\bar{\boldsymbol{v}}$ divided by the number of terms composing the component, is always less than or equal to the sum of the squared components going into it, up to the final two components of $\bar{\boldsymbol{v}}$. □

Lemma 13. *Let v be as in* (7.23)*,* $\bar{\boldsymbol{v}}$ *be as in* (7.31)*. Then,*

$$(\theta_1 - 1)(\theta_1 - 3) + \boldsymbol{v}^T \boldsymbol{v} - \bar{\boldsymbol{v}}^T Q^{-1} \bar{\boldsymbol{v}} \geq 0 \tag{7.35}$$

Proof. Recall $\theta_1 \in \mathbb{N}$ and from Lemma 9 that $Q = \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix}$ where $Q_1 = \begin{pmatrix} \theta_2 & \cdots & \theta_{\kappa_J - 1} \end{pmatrix}$ and Q_2 depends on whether $n+1 \in J$. Apply Lemma 12 to each term of $\bar{\boldsymbol{v}}$ with corresponding terms of $\boldsymbol{v}$, then

$$\bar{\boldsymbol{v}}^T \begin{bmatrix} Q_1^{-1} & 0 \\ 0 & 0 \end{bmatrix} \bar{\boldsymbol{v}} \leq \boldsymbol{v}^T \begin{bmatrix} I_{n-\theta_{\kappa_J}} & 0 \\ 0 & 0 \end{bmatrix} \boldsymbol{v}. \tag{7.36}$$

Partition $U = \begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix}$ so that $U_2 \in \mathbb{R}^{2\times 2}$. If $n+1 \notin J$, by Lemma 8, $Q_2^{-1} = I_2 = U_2$. Then,

$$\bar{v}_{\kappa_J-1}^2 + \bar{v}_{\kappa_J}^2 = \bar{\boldsymbol{v}}^T \begin{bmatrix} 0 & 0 \\ 0 & Q_2^{-1} \end{bmatrix} \bar{\boldsymbol{v}} = \boldsymbol{v}^T \begin{bmatrix} 0 & 0 \\ 0 & U_2 \end{bmatrix} \boldsymbol{v} = v_n^2 + v_{n+1}^2. \tag{7.37}$$

If $n+1 \in J$, we have $Q_2 = \begin{bmatrix} \theta_{\kappa_J} & 1 \\ 1 & 2 \end{bmatrix}$ with $Q_2^{-1} = \frac{1}{2\theta_{\kappa_J}-1} \begin{bmatrix} 2 & -1 \\ -1 & \theta_{\kappa_J} \end{bmatrix}$ and,

$$\boldsymbol{v}^T \begin{bmatrix} 0 & 0 \\ 0 & I_{\theta_{\kappa_J}+1} \end{bmatrix} \boldsymbol{v} = \sum_{j=1}^{\theta_{\kappa_J}+1} v_{n+2-i}^2 = \sum_{j=1}^{\theta_{\kappa_J}+1} (n-i)^2$$

and,

$$\bar{\boldsymbol{v}}^T \begin{bmatrix} 0 & 0 \\ 0 & Q_2^{-1} \end{bmatrix} \bar{\boldsymbol{v}} = \frac{2\bar{v}_{\kappa_J-1}^2 + 2\bar{v}_{\kappa_J-1} + \theta_{\kappa_J}}{2\theta_{\kappa_J}-1}$$

Direct calculation shows that $\bar{v}_{\kappa_J-1} = \theta_{\kappa_J}(n-\theta_{\kappa_J}-1) + \theta_{\kappa_J}(\theta_{\kappa_J}-1)/2$. Consider,

$$\begin{aligned}
&\boldsymbol{v}^T \begin{bmatrix} 0 & 0 \\ 0 & I_{\theta_{\kappa_J}+1} \end{bmatrix} \boldsymbol{v} - \bar{\boldsymbol{v}}^T \begin{bmatrix} 0 & 0 \\ 0 & Q_2^{-1} \end{bmatrix} \bar{\boldsymbol{v}} \\
&= \sum_{j=1}^{\theta_{\kappa_J}+1} (n-i)^2 - \frac{2\bar{v}_{\kappa_J-1}^2 + 2\bar{v}_{\kappa_J-1} + \theta_{\kappa_J}}{2\theta_{\kappa_J}-1} \\
&= \frac{(\theta_{\kappa_J}-1)(6-12n+6n^2+(6n-5)\theta_{\kappa_J} - \theta_{\kappa_J}^2 + \theta_{\kappa_J}^3)}{6(2\theta_{\kappa_J}-1)} \\
&= \frac{(\theta_{\kappa_J}-1) f(n,\theta_{\kappa_J})}{6(2\theta_{\kappa_J}-1)}
\end{aligned}$$

Since $n+1 \in J$, we have that $n \geq \theta_{\kappa_J} \geq 2$. Then, $2\theta_{\kappa_J}-1 > \theta_{\kappa_J}-1 > 0$. Since $6n^2-12n+6 \geq 0$

as well as $6n-5 \geq 1$, we have

$$f(n, \theta_{\kappa_J}) \geq \theta_{\kappa_J}^3 - \theta_{\kappa_J}^2 + \theta_{\kappa_J} \geq 0$$

where $\theta_{\kappa_J}^3 - \theta_{\kappa_J}^2 + \theta_{\kappa_J} = \theta_{\kappa_J}(\theta_{\kappa_J}^2 - \theta_{\kappa_J} + 1) \geq 0 \; \forall \theta_{\kappa_J} \in \mathbb{R}$, and we conclude

$$\bar{\boldsymbol{v}}^T \begin{bmatrix} 0 & 0 \\ 0 & Q_2^{-1} \end{bmatrix} \bar{\boldsymbol{v}} \leq \boldsymbol{v}^T \begin{bmatrix} 0 & 0 \\ 0 & I_{\theta_{\kappa_J}+1} \end{bmatrix} \boldsymbol{v} \tag{7.38}$$

Therefore, in either case, we have

$$\begin{aligned} \bar{\boldsymbol{v}}^T Q^{-1} \bar{\boldsymbol{v}} &= \bar{\boldsymbol{v}}^T \begin{bmatrix} Q_1^{-1} & 0 \\ 0 & 0 \end{bmatrix} \bar{\boldsymbol{v}} + \bar{\boldsymbol{v}}^T \begin{bmatrix} 0 & 0 \\ 0 & Q_2^{-1} \end{bmatrix} \bar{\boldsymbol{v}} \\ &\leq \boldsymbol{v}^T \begin{bmatrix} I_{n-\theta_{\kappa_J}} & 0 \\ 0 & 0 \end{bmatrix} \boldsymbol{v} + \boldsymbol{v}^T \begin{bmatrix} 0 & 0 \\ 0 & I_{\theta_{\kappa_J}+1} \end{bmatrix} \boldsymbol{v} = \boldsymbol{v}^T \boldsymbol{v} \end{aligned} \tag{7.39}$$

Recall, $(\theta_1 - 1)(\theta_1 - 3) \geq 0$ for $\theta_1 \neq 2$, which can only occur if $2 \in J$ by Lemma 8. Therefore, if $2 \notin J$, the result follows.

If $2 \in J$, then $\theta_1 \geq 2$ and correspondingly, the first column of U^T is $\mathbf{0}$. Therefore, $\bar{v}_1 = (U^T)_1 \boldsymbol{v}$ does not contain the first term of $\boldsymbol{v}$, $v_1 = -1$, and since $\bar{\boldsymbol{v}} = U^T \boldsymbol{v}$, it follows $\bar{\boldsymbol{v}}$ does not contain this term either. Therefore, by the inequality above, we may remove this term from $\boldsymbol{v}^T \boldsymbol{v}$ so that

$$(\theta_1 - 1)(\theta_1 - 3) + \boldsymbol{v}^T \boldsymbol{v} - \bar{\boldsymbol{v}}^T Q^{-1} \bar{\boldsymbol{v}} \geq -1 + \boldsymbol{v}^T \boldsymbol{v} - \bar{\boldsymbol{v}}^T Q^{-1} \bar{\boldsymbol{v}} \geq 0.$$

yielding the result. □

Lemma 14. *Let the index set J be as in Definition 4,* $\beta > 0$ *as in Lemma 5, and* V_G *and its*

partition $\bar{\alpha}$, $\boldsymbol{v}$, *and* $\bar{D}$ *be as in* (7.31). *Then*

$$\bar{\beta} = \bar{\alpha} - \bar{\boldsymbol{v}}^T \bar{D}^{-1} \bar{\boldsymbol{v}} \geq \beta > 0.$$

Furthermore, V_G *is nonsingular and*

$$V_G^{-1} = \begin{bmatrix} \bar{\beta}^{-1} & -\bar{\beta}^{-1} \bar{\boldsymbol{v}}^T \bar{D}^{-1} \\ -\bar{\beta}^{-1} \bar{D}^{-1} \bar{\boldsymbol{v}} & \bar{D}^{-1} + \bar{\beta}^{-1} \bar{D}^{-1} \bar{\boldsymbol{v}} \bar{\boldsymbol{v}}^T \bar{D}^{-1} \end{bmatrix}.$$

Proof. From Lemma 9, we have $\bar{D} = Q - 2e_{\kappa_J} e_{\kappa_J}^T$. From (7.23), we have $\boldsymbol{v}^T D^{-1} \boldsymbol{v} = \boldsymbol{v}^T \boldsymbol{v} - 2(n-1)^2$. If $n+1 \notin J$, both $\bar{D}$ and Q are diagonal matrices, therefore,

$$\bar{\boldsymbol{v}}^T (\bar{D}^{-1} - Q^{-1}) \bar{\boldsymbol{v}} = -2 \bar{\boldsymbol{v}}^T \boldsymbol{e}_{\kappa_J} \boldsymbol{e}_{\kappa_J}^T \bar{\boldsymbol{v}} = -2(n-1)^2$$

from which it follows,

$$\begin{aligned} \bar{\beta} - \beta &= \bar{\alpha} - \bar{\boldsymbol{v}}^T \bar{D}^{-1} \bar{\boldsymbol{v}} - (\alpha - \boldsymbol{v}^T D^{-1} \boldsymbol{v}) \\ &= (\theta_1 - 1)(\theta_1 - 3) - (\bar{\boldsymbol{v}}^T Q^{-1} \bar{\boldsymbol{v}} - 2(n-1)^2) + \boldsymbol{v}^T \boldsymbol{v} - 2(n-1)^2 \\ &= (\theta_1 - 1)(\theta_1 - 3) + \boldsymbol{v}^T \boldsymbol{v} - \bar{\boldsymbol{v}}^T Q^{-1} \bar{\boldsymbol{v}} \end{aligned}$$

By Lemma 13, $(\theta_1 - 1)(\theta_1 - 3) + \boldsymbol{v}^T \boldsymbol{v} - \bar{\boldsymbol{v}}^T Q^{-1} \bar{\boldsymbol{v}} \geq 0$, and the result follows.

now assume $n+1 \in J$, then $n \geq \theta_{\kappa_J} \geq 2$ and $\bar{v}_{\kappa_J} = -1$. Recalling $\bar{v}_{\kappa_J - 1} = \theta_{\kappa_J}(n - \theta_{\kappa_J} - 1) + \theta_{\kappa_J}(\theta_{\kappa_J} - 1)/2$, we find,

$$\begin{aligned} \bar{\boldsymbol{v}}^T (\bar{D}^{-1} - Q^{-1}) \bar{\boldsymbol{v}} &= \frac{-2}{2\theta_{\kappa_J} - 1} \begin{pmatrix} \bar{v}_{\kappa_J - 1} & -1 \end{pmatrix} \begin{bmatrix} 1 & -\theta_{\kappa_J} \\ -\theta_{\kappa_J} & \theta_{\kappa_J}^2 \end{bmatrix} \begin{pmatrix} \bar{v}_{\kappa_J - 1} \\ -1 \end{pmatrix} \\ &= -\frac{\theta_{\kappa_J}^2 (1 - 2n + \theta_{\kappa_J})^2}{2(2\theta_{\kappa_J} - 1)} \end{aligned}$$

Therefore,

$$\begin{aligned}
\bar{\beta} - \beta &= \bar{\alpha} - \bar{\boldsymbol{v}}^T \bar{D}^{-1} \bar{\boldsymbol{v}} - (\alpha - \boldsymbol{v}^T D^{-1} \boldsymbol{v}) \\
&= (\theta_1 - 1)(\theta_1 - 3) + \boldsymbol{v}^T \boldsymbol{v} - \bar{\boldsymbol{v}}^T Q^{-1} \bar{\boldsymbol{v}} - 2(n-1)^2 + \frac{\theta_{\kappa_J}^2 (1 - 2n + \theta_{\kappa_J})^2}{2(2\theta_{\kappa_J} - 1)} \\
&\geq \frac{\theta_{\kappa_J}^2 (1 - 2n + \theta_{\kappa_J})^2}{2(2\theta_{\kappa_J} - 1)} - 2(n-1)^2 \\
&= f_{\theta_{\kappa_J}}(n)
\end{aligned}$$

where the inequality follows from Lemma 13. After some algebra we have,

$$\begin{aligned}
f_{\theta_{\kappa_J}}(n) = 4 - 2\theta_{\kappa_J} + \theta_{\kappa_J}^2 + 2\theta_{\kappa_J}^3 + \theta_{\kappa_J}^4 \\
+ (-8 + 16\theta_{\kappa_J} - 4\theta_{\kappa_J}^2 - 4\theta_{\kappa_J}^3)n + (4 - 8\theta_{\kappa_J} + 4\theta_{\kappa_J}^2)n^2
\end{aligned}$$

note that the n^2 term, $4 - 8\theta_{\kappa_J} + 4\theta_{\kappa_J}^2 = 4(\theta_{\kappa_J} - 1)^2$ is always positive, and that $f_{\theta_{\kappa_J}}(n)$ is a strictly convex function of n since $\theta_{\kappa_J} \geq 2$. The global minimum occurs at $n = \theta_{\kappa_J}^2 - 3\theta_{\kappa_J}^3$ and is always negative. Hence, when $\theta_{\kappa_J} \geq 2$ we have, $f_{\theta_{\kappa_J}}(n) \geq f_{\theta_{\kappa_J}}(0) = 4 - 8\theta_{\kappa_J} + \theta_{\kappa_J}^2 + 2\theta_{\kappa_J}^3 + \theta_{\kappa_J}^4 \geq 0$ and the result follows. note that V_G^{-1} comes from the matrix inversion Lemma [HJ13]. Since $\bar{D}$ is nonsingular and its Schur complement $\bar{\beta} > 0$, V_G is nonsingular. □

We now have the machinery necessary to address problem (7.48), that is

$$T_G^T V_G T_G \begin{pmatrix} \boldsymbol{x} \\ \lambda \end{pmatrix} = \begin{bmatrix} G^T A^T A G & G^T \boldsymbol{c} \\ \boldsymbol{c}^T G & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{x} \\ \lambda \end{pmatrix} = \begin{pmatrix} \boldsymbol{0} \\ -\varepsilon \end{pmatrix} = -\varepsilon \boldsymbol{e}_{\kappa_J + 1}.$$

From Lemma 6 the last column of T_G^{-T} is always $\boldsymbol{e}_{\kappa_J+1}$, so that

$$T_G \begin{pmatrix} \boldsymbol{x} \\ \lambda \end{pmatrix} = -\varepsilon V_G^{-1} T_G^{-T} \boldsymbol{e}_{\kappa_J+1} = -\varepsilon V_G^{-1} \boldsymbol{e}_{\kappa_J+1} \tag{7.40}$$

The terms in the right-side equation above can be explicitly calculated.

Lemma 15. *The product*

$$V_G^{-1}\boldsymbol{e}_{\kappa_J+1} = \bar{\beta}^{-1}u_{\kappa_J}\begin{pmatrix} -1 \\ \boldsymbol{u} - \bar{\beta}\boldsymbol{w}/u_{\kappa_J} \end{pmatrix} \tag{7.41}$$

where, $u_{\kappa_J} = n-1 > 0$, $\boldsymbol{w} = \boldsymbol{e}_{\kappa_J}$ *if* $n+1 \notin J$, *and* $u_{\kappa_J} = \bar{v}_{\kappa_J-1} + \theta_{\kappa_J} > 0$, $\boldsymbol{w} = \theta_{\kappa_J}\boldsymbol{e}_{\kappa_J} - \boldsymbol{e}_{\kappa_J-1}$ *if* $n+1 \in J$.

Proof. Using V_G^{-1} from Lemma 14

$$V_G^{-1}\boldsymbol{e}_{\kappa_J+1} = \bar{\beta}^{-1}\begin{pmatrix} -\bar{\boldsymbol{v}}^T\bar{D}^{-1}\boldsymbol{e}_{\kappa_J} \\ (\bar{\beta}\bar{D}^{-1} + \bar{D}^{-1}\bar{\boldsymbol{v}}\bar{\boldsymbol{v}}^T\bar{D}^{-1})\boldsymbol{e}_{\kappa_J} \end{pmatrix}.$$

We have $\boldsymbol{u} = \bar{D}^{-1}\bar{\boldsymbol{v}} = \bar{D}^{-1}U^T\boldsymbol{v}$ from which it follows

$$\boldsymbol{u}\boldsymbol{u}^T = \begin{bmatrix} u_1\boldsymbol{u} & u_2\boldsymbol{u} & \cdots & u_{\kappa_J}\boldsymbol{u} \end{bmatrix} = \bar{D}^{-1}\bar{\boldsymbol{v}}\bar{\boldsymbol{v}}^T\bar{D}^{-1}$$

Therefore, letting $\boldsymbol{w} = \bar{\beta}\bar{D}^{-1}\boldsymbol{e}_{\kappa_J}$,

$$(\bar{\beta}\bar{D}^{-1} + \bar{D}^{-1}\bar{\boldsymbol{v}}\bar{\boldsymbol{v}}^T\bar{D}^{-1})\boldsymbol{e}_{\kappa_J} = \boldsymbol{w} + u_{\kappa_J}\boldsymbol{u}$$

and $\bar{\boldsymbol{v}}^T\bar{D}^{-1}\boldsymbol{e}_{\kappa_J} = \boldsymbol{u}^T\boldsymbol{e}_{\kappa_J} = u_{\kappa_J}$ which leads to (7.41).

Let $n+1 \notin J$. From Lemma 9, it follows that $\bar{D}^{-1}\boldsymbol{e}_{\kappa_J} = -\boldsymbol{e}_{\kappa_J}$, then

$$u_{\kappa_J} = \boldsymbol{u}^T\boldsymbol{e}_{\kappa_J} = \bar{\boldsymbol{v}}^T\bar{D}^{-1}\boldsymbol{e}_{\kappa_J} = -\bar{\boldsymbol{v}}^T\boldsymbol{e}_{\kappa_J} = -\bar{v}_{\kappa_J} = -v_{n+1} = n-1$$

and

$$\boldsymbol{w} = \bar{\beta}\bar{D}^{-1}\boldsymbol{e}_{\kappa_J} = -\bar{\beta}\boldsymbol{e}_{\kappa_J} = \begin{pmatrix} 0 & \cdots & 0 & -\bar{\beta} \end{pmatrix}^T.$$

now let $n+1 \in J$. From Lemma 9, $\bar{D}^{-1}\boldsymbol{e}_{\kappa_J} = \boldsymbol{e}_{\kappa_J - 1} - \theta_{\kappa_J}\boldsymbol{e}_{\kappa_J}$, so that

$$u_{\kappa_J} = \boldsymbol{u}^T\boldsymbol{e}_{\kappa_J} = \bar{\boldsymbol{v}}^T\bar{D}^{-1}\boldsymbol{e}_{\kappa_J} = \bar{v}_{\kappa_J - 1} - \bar{v}_{\kappa_J}\theta_{\kappa_J} = \bar{v}_{\kappa_J - 1} + \theta_{\kappa_J}$$

since $\bar{v}_{\kappa_J} = -1$. Furthermore,

$$\boldsymbol{w} = \bar{\beta}\bar{D}^{-1}\boldsymbol{e}_{\kappa_J} = \bar{\beta}(\boldsymbol{e}_{\kappa_J - 1} - \theta_{\kappa_J}\boldsymbol{e}_{\kappa_J}) = \bar{\beta}\begin{pmatrix} 0 & \cdots & 0 & 1 & -\theta_{\kappa_J} \end{pmatrix}^T,$$

concluding the proof. □

Corollary 1. *Let $\boldsymbol{u}$ be as defined in Lemma 10 and $\boldsymbol{v}$ as in* (7.23)*, then $u_1 \geq -1$.*

Proof. From Lemma 8, $\min(\bar{J}_2) = 2$ and $\min(\bar{J}_3) = 3$, therefore by Lemma 10, we have $u_1 = \frac{\bar{v}_1}{\theta_2} = \frac{1}{2}(\bar{J}_3 + \bar{J}_2 - 7) \geq -1$ □

We are now ready to prove the main result of this section.

Lemma 16. *Let T_G be as in Lemma 7 and V_G be as in* (7.31) *and $\varepsilon > 0$. Then*

$$\begin{pmatrix} \boldsymbol{x} \\ \lambda \end{pmatrix} = -\varepsilon T_G^{-1} V_G^{-1} \boldsymbol{e}_{\kappa_J}.$$

is such that $\boldsymbol{x} \leq \boldsymbol{0}$.

Proof. From Lemma 7,

$$T_G^{-1} = T_{\kappa_J + 1}^{-1} + \zeta \boldsymbol{e}_{\kappa_J}\boldsymbol{e}_{\kappa_J + 1}^T,$$

where $\zeta = 0$ if $n+1 \notin J$ and $\zeta = 1$ if $n+1 \in J$. One can then show using Lemma 15 that

$$\boldsymbol{z} = \bar{\beta}\, u_{\kappa_J}^{-1}\, T_G^{-1} V_G^{-1} \boldsymbol{e}_{\kappa_J}$$

in which the entries of $\boldsymbol{z}$ are given by

$$\begin{aligned}
z_1 &= 1, \\
z_i &= u_{i-1} - u_{i-2} \geq 0, \qquad i = 2, \ldots \kappa_{J-1} \\
z_{\kappa_J} &= u_{\kappa_J - 1} - u_{\kappa_J - 2} - \bar{\beta} w_{\kappa_J - 1}/u_{\kappa_J} + (\zeta - 1)(u_{\kappa_J} - \bar{\beta} w_{\kappa_J}/u_{\kappa_J}) \\
z_{\kappa_J + 1} &= u_{\kappa_J} - \bar{\beta} w_{\kappa_J}/u_{\kappa_J},
\end{aligned}$$

with $u_0 = -1$. That $z_i \geq 0$, $i = 1, \ldots \kappa_{J-1}$, follows from Corollary 1, for $i = 2$, and Lemma 10, for $i = 2, ..., \kappa_J - 1$. Since

$$\begin{pmatrix} \boldsymbol{x} \\ \lambda \end{pmatrix} = -\varepsilon \bar{\beta}^{-1} u_{\kappa_J} \boldsymbol{z}$$

and $\varepsilon > 0$, $\bar{\beta} > 0$, and $u_{\kappa_J} > 0$, it remains to show that

$$z_{\kappa_J} = u_{\kappa_J - 1} - u_{\kappa_J - 2} - \bar{\beta} w_{\kappa_J - 1}/u_{\kappa_J} + (\zeta - 1)(u_{\kappa_J} - \bar{\beta} w_{\kappa_J}/u_{\kappa_J}) \geq 0.$$

When $n+1 \notin J$, then by Lemmas 10 and 15

$$u_{\kappa_J - 1} - u_{\kappa_J - 2} \geq 1 \qquad u_{\kappa_J} = n - 1, \qquad w_{\kappa_J - 1} = 0, \qquad w_{\kappa_J} = 1, \qquad \zeta = 0$$

so that

$$z_{\kappa_J} \geq 1-(n-1)+\frac{\bar{\beta}}{n-1} \geq 1-(n-1)+\frac{(n-1)^2}{n-1} = 1 \geq 0,$$

where the penultimate inequality follows from the fact that $\bar{\beta} \geq \beta = (n-1)^2$, as shown in Lemmas 5 and 14.

The case when $n+1 \in J$ is more involved. From Lemmas 10 and 15

$$u_{\kappa_J-1} = -1, \qquad w_{\kappa_J-1} = -1 \qquad w_{\kappa_J} = \theta_{\kappa_J} \qquad \zeta = 1$$

and it follows,

$$z_{\kappa_J} = \frac{\bar{\beta}}{u_{\kappa_J}} - 1 - u_{\kappa_J-2}$$

Substituting u_{κ_J} and u_{κ_J-2} from Lemma 10

$$z_{\kappa_J} = \frac{\bar{\beta}}{\bar{v}_{\kappa_J-1}+\theta_{\kappa_J}} - \frac{\bar{v}_{\kappa_J-2}}{\theta_{\kappa_J-1}} - 1 = \frac{\bar{\beta} - (\bar{v}_{\kappa_J-1}+\theta_{\kappa_J})\left(\frac{\bar{v}_{\kappa_J-2}}{\theta_{\kappa_J-1}}+1\right)}{\bar{v}_{\kappa_J-1}+\theta_{\kappa_J}}$$

From Lemma 14

$$\begin{aligned}
\bar{\beta} &= \bar{\alpha} - \bar{\boldsymbol{v}}^T \bar{D}^{-1} \bar{\boldsymbol{v}} \\
&= \alpha + (\theta_1 - 1)(\theta_3 - 1) - \left(\frac{\bar{v}_1^2}{\theta_2} + \frac{\bar{v}_2^2}{\theta_3} + \cdots + \frac{\bar{v}_{\kappa_J-2}^2}{\theta_{\kappa_J-1}} - 2\bar{v}_{\kappa_J-1} - \theta_{\kappa_J} \right) \\
&\geq \alpha - 1 + 2\bar{v}_{\kappa_J-1} + \theta_{\kappa_J} - \left(\frac{\bar{v}_1^2}{\theta_2} + \frac{\bar{v}_2^2}{\theta_3} + \cdots + \frac{\bar{v}_{\kappa_J-2}^2}{\theta_{\kappa_J-1}} \right)
\end{aligned}$$

where the last inequality comes from Lemma 8. Therefore

$$z_{\kappa_J} \geq \frac{y_{\kappa_J}}{\bar{v}_{\kappa_J-1} + \theta_{\kappa_J}}$$

in which

$$y_{\kappa_J} = \alpha - 1 + \bar{v}_{\kappa_J-1} - \left(\frac{\bar{v}_1^2}{\theta_2} + \frac{\bar{v}_2^2}{\theta_3} + \cdots + \frac{\bar{v}_{\kappa_J-2}^2}{\theta_{\kappa_J-1}} \right) - \frac{\bar{v}_{\kappa_J-2}}{\theta_{\kappa_J-1}} (\bar{v}_{\kappa_J-1} + \theta_{\kappa_J}). \tag{7.42}$$

From Lemmas 8 and 10, $\theta_{\kappa_J} \geq 1$ and $\bar{v}_{\kappa_J-1} \geq 0$. Therefore, z_{κ_J} will be non negative if $y_{\kappa_J} \geq 0$. If we are able to prove that

$$\left(\frac{\bar{v}_1^2}{\theta_2} + \cdots + \frac{\bar{v}_{\kappa_J-2}^2}{\theta_{\kappa_J-1}} \right) + \frac{\bar{v}_{\kappa_J-2}}{\theta_{\kappa_J-1}} (\bar{v}_{\kappa_J-1} + \theta_{\kappa_J}) \leq \frac{1}{6} \left(2n^3 - 9n^2 + 7n - 6 \right) \tag{7.43}$$

then substituting α from Lemma 5, $\bar{v}_{\kappa_J-1} \geq 2n-5$ from Lemma 10, and (7.43) into (7.42), one obtains for $n \geq 3$,

$$y_{\kappa_J} \geq \frac{n}{6}(2n^2 - 9n + 13) + 2n - 6 - \frac{1}{6} \left(2n^3 - 9n^2 + 7n - 6 \right) = 3n - 5 \geq 0,$$

which will complete the proof. All that remains to prove is (7.43). Let J be such that, for $p \in \mathbb{Z}$ and $0 \leq p \leq n-2$, it holds that

$$\{n+1-i\}_{i=0}^{i=p} \subseteq J \qquad \text{and} \qquad J \cap \{n-p\} = \emptyset$$

so that $n+1 \in J$ and $J \subset \{2,3,...,n+1\}$. With J parameterized in this way, we can offer

alternative calculations of $\bar{v}_{\kappa_J-1}$ and θ_{κ_J} based on $\bar{\boldsymbol{v}} = U^T \boldsymbol{v}$ from (7.31) and Lemma 8,

$$\theta_{\kappa_J} = p+2 \tag{7.44}$$

$$\begin{aligned}\bar{v}_{\kappa_J-1} &= v_n + \sum_{i=0}^{p} v_{n-1-i} = n-2+\sum_{i=0}^{p} n-3-i \\ &= (n-3)(p+2) - p(p+1)/2+1\end{aligned} \tag{7.45}$$

It is also clear from Lemma 10,

$$\frac{\bar{v}_{\kappa_J-2}}{\theta_{\kappa_J-1}} \leq V_{n-2-p} = n-4-p. \tag{7.46}$$

Furthermore, repeated application of Lemma 12 along with (7.46) yields

$$\left(\frac{\bar{v}_1^2}{\theta_2}+\cdots+\frac{\bar{v}_{\kappa_J-2}^2}{\theta_{\kappa_J-1}}\right) \leq 1+\frac{(n-4-p)(n-3-p)(2n-7-2p)}{6} \tag{7.47}$$

Therefore, consider the left-hand side of (7.43),

$$\begin{aligned}&\left(\frac{\bar{v}_1^2}{\theta_2}+\cdots+\frac{\bar{v}_{\kappa_J-2}^2}{\theta_{\kappa_J-1}}\right)+\frac{\bar{v}_{\kappa_J-2}}{\theta_{\kappa_J-1}}(\bar{v}_{\kappa_J-1}+\theta_{\kappa_J}) \\ &\quad\leq \left(\frac{\bar{v}_1^2}{\theta_2}+\cdots+\frac{\bar{v}_{\kappa_J-2}^2}{\theta_{\kappa_J-1}}\right) \\ &\qquad\qquad +(n-4-p)\left((n-3)(p+2)-p(p+1)/2+1+p+2\right) \\ &\quad\leq 1+\frac{(n-4-p)(n-3-p)(2n-7-2p)}{6} \\ &\qquad\qquad +(n-4-p)\left((n-3)(p+2)-p(p+1)/2+p+3\right) \\ &\quad= f_n(p)\end{aligned}$$

where the first inequality follows from (7.44), (7.45), and (7.46), and the second inequality

follows from (7.47). The resulting upper bound $f_n(p)$ is the following polynomial on p

$$f_n(p) = \frac{p^3}{6} - \left(\frac{n}{2} - 1\right) p^2 + \left(\frac{5}{6} - \frac{3n}{2}\right) p + \left(\frac{n^3}{3} - \frac{3n^2}{2} + \frac{7n}{6} - 1\right).$$

note that $f_n(p)$ is cubic in p and the leading coefficient is positive. Taking the derivative of f_n yields a convex quadratic, and setting it equal to zero, we identify the local minima and maxima of $f_n(p)$,

$$p_1 = n - 2 - \sqrt{7/3 - n + n^2} \qquad\qquad p_2 = n - 2 + \sqrt{7/3 - n + n^2}.$$

When $n \geq 3$, it is straightforward to show that

$$p_1 < 0, \quad \text{and} \quad p_2 > n - 2 \geq 1$$

and $df/dp < 0$ for $p \in [0, n-2]$. Therefore, the maximum of $f_n(p)$ occurs at $p = 0$, and we have,

$$f_n(p) \leq \max_{p \in [0, n-2]} f_n(p) = f_n(0) = \frac{1}{6}(2n^3 - 9n^2 + 7n - 6)$$

which proves (7.43). □

7.7.4 The Final Connection

Recall, we delayed proof of Lemma 17 in Section 7.6. Section 7.7 has developed all of the technical results necessary to prove the Lemma. We conclude this section with a proof of Lemma 17.

Lemma 17. *Let G be of the form in Definition 4. For any* $\varepsilon > 0$*,*

$$\begin{bmatrix} G^T A^T A G & G^T \boldsymbol{c} \\ \boldsymbol{c}^T G & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{x}_G \\ \lambda_G \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ -\varepsilon \end{pmatrix} \tag{7.48}$$

has a unique solution $(\boldsymbol{x}_G, \lambda_G)$ *and* $\boldsymbol{x}_G \leq \mathbf{0}$*.*

Proof. We have,

$$W_G \begin{pmatrix} \boldsymbol{x}_G \\ \lambda_G \end{pmatrix} = T_G^T V_G T_G \begin{pmatrix} \boldsymbol{x}_G \\ \lambda_G \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ -\varepsilon \end{pmatrix}$$

Since T_G and V_G are both invertible by Lemmas 6 and 14, we have,

$$\begin{pmatrix} \boldsymbol{x}_G \\ \lambda_G \end{pmatrix} = T_G^{-1} V_G^{-1} T_G^{-T} \begin{pmatrix} \mathbf{0} \\ -\varepsilon \end{pmatrix} = -\varepsilon T_G^{-1} V_G^{-1} \boldsymbol{e}_{\kappa_J}$$

and by Lemma (16), it follows that the solution $\boldsymbol{x}_G \leq 0$ and is unique. □

7.8 Conclusions

In this paper we have devised a method for estimating the parameters of a sawtooth function from a noisy data set by solving a series of convex relaxations. Our main result is a proof that the cardinality of the set of jumps in the estimated sawtooth function is a non-decreasing function of the parameter of a corresponding relaxed convex constraint. This result is significantly stronger than comparable relaxation strategies, such as the popular LASSO regression, in which minimal cardinality solutions are sought but not guaranteed. This problem's solution is of general interest in engineering, physics and applied mathematics and minimal cardinality solutions have gained more attention with the rise in applications of Machine Learning. Indeed, any PWL signal

with a constant negative slope of any magnitude is well defined by our problem structure.

The main result is dependent on the structure of the problem setup, in particular the matrix A, which parameterizes the flow of time for given PWL signals. In future work, the structure of the A matrix can be further investigated, and more generalized requirements may be found that continue to enforce the minimal cardinality result, which is of prime interest in optimization problems and the rapidly growing field of data science. We conjecture that the key property enabling proof of the cardinality result comes from the unique, "stepped" structure of A^TA.

Chapter 7, in full, is a reprint of the material as it was submitted to Journal of Optimization Theory and Application. Allen, Cody; de Oliveira, Maurcio, Springer 2020. The dissertation author was the primary investigator and author on this paper.

Chapter 8

Conclusion

This dissertation has attempted to add some useful modeling techniques, interesting results and thought provoking discourse to the body of literature of gas turbines, particularly, industrial gas turbines. In the so-called age of machine learning and data science, our goal has been to marry together the physics of the 20th century with the new algorithms and computational resources of the 21st century. In our attempt we have taken the hybrid approach in that we have utilized known relationships and equations all the while trying to use data to fit model parameters, both physics based and non-physics based. We believe we have shown enough evidence for the reader to conclude that this hybrid approach to modeling physical systems is superior to just one or the other by themselves.

This dissertation has left many closely related problems for future research. Indeed, as most research goes, we have only peeled back some of the layers of the onion, but there are yet others. One example is in the use of implementing the reduced rank models and the Mahalonobis distance. The method provided was heuristic in nature and further investigations may lead to better rules for classifying healthy or unhealthy machines. Another example is in the optimization of maintenance schedules and the requirement of linear slopes of degradation. Further research

could extend the method developed here to include nonlinear degradation functions. A final example is the extension of the minimum cardinality theorem. Our theorem depends on the special structure of the matrix A, and the parameterization of the sawtooth function. It is our belief that this matrix structure can be exploited further to more generalized functions, while retaining the cardinality theorem.

Lastly, we leave the reader with our thoughts on applying machine learning generally to physics. There is indeed a blurry line between where machine learning starts and physics ends. After all, most of physics is mathematical models in some form, where optimization may be used to solve for coefficients. In the time it has taken to perform all of the research contained within, there has been an exponential increase in the number of research papers published that utilize machine learning algorithms like neural networks. While this is leading to better modeling of physical systems, a certain clarity of thought and simplicity in understanding is lost with these pieces of work. The smallest of useful neural networks are quite complex, and typically have very little describable properties relating directly to the physical systems they model. In my opinion, the use of these algorithms and machine learning in general is leading to lazy thinking. In science it is not sufficient to model a system to a given degree of accuracy; this is not where the scientist stops. Science was developed to uncover the truths of nature, and to transcribe them into physical laws. Black box modeling does not achieve this goal. It is up to us, the new generation of scientists, to remember this credo, and to not stop searching for simple, elegant truths and laws of natures. I finish with a quote, “The important thing is to not stop questioning, curiosity has its own reason for existing,” - some really smart guy named Albert Einstein.

Bibliography

[AHdO19] Cody W Allen, Chad M Holcomb, and Maurício de Oliveira. Estimating recoverable performance degradation rates and optimizing maintenance scheduling. *Journal of Engineering for Gas Turbines and Power*, 141(1):011032, 2019.

[AJ08] Peter Atkins and Loretta Jones. *Chemical Principles*. W.H. Freeman and Company, 4th edition, 2008.

[AMM13] Hamed Ahmadi, José R Martí, and Ali Moshref. Piecewise linear approximation of generators cost functions using max-affine functions. In *2013 IEEE Power & Energy Society General Meeting*, pages 1–5. IEEE, 2013.

[And85] T.W. Anderson. *Introduction to Multivariate Statistical Analysis*. Wiley, 1985.

[Apo74] Tom Apostol. *Mathematical Analysis*. Addison-Wesley, Boston, MA/US, 2nd edition, 1974.

[ARDM12] N Aretakis, I Roumeliotis, G Doumouras, and K Mathioudakis. Compressor washing economic analysis and optimization for power generation. *Applied energy*, 95:77–86, 2012.

[AW05] George B Arfken and Hans J Weber. *Mathematical methods for physicists*. Elsevier Academic Press, Waltham, MA/US, 6th edition, 2005.

[BBPB18] A. Bemporad, V. Breschi, D. Piga, and S. Boyd. Fitting jump models. *Automatica*, 96:11–21, 2018.

[Bej88] Adrian Bejan. *Advanced Engineering Thermodynamics*. John Wiley and Sons, 1st edition, 1988.

[BG07] Meherwan P Boyce and Francisco Gonzalez. A study of on-line and off-line turbine washing to optimize the operation of a gas turbine. *Journal of engineering for gas turbines and power*, 129(1):114–122, 2007.

[BGFK15] Klaus Brun, Terrence A Grimley, William C Foiles, and Rainer Kurz. Experimental evaluation of the effectiveness of online water-washing in gas turbine compressors. *Journal of Engineering for Gas Turbines and Power*, 137(4):042605, 2015.

[BK18] D. Burnes and R. Kurz. Performance degradation effects in modern industrial gas turbines. In *Proceedings of Zurich 2018 Global Power and Propulsion Forum*, 2018.

[Boy12] Meherwan P Boyce. *Gas turbine engineering handbook*. Elsevier, Waltham, MA/US, 4th edition, 2012.

[BV04] Stepehn Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, United Kingdom, 2004.

[C.A16] M.de Oliveira C.Allen, C. Holcomb. Fault detection using reduced rank linear engine models. *Proceedings of the ASME Turbo Expo: Turbine Technical Conference*, 2016.

[Cha12] Chris Chatfield. *The Analysis of Time Series*. Chapman Hall, 2012.

[CLM05] Paolo Chiesa, Giovanni Lozza, and Luigi Mazzocchi. Using hydrogen as gas turbine fuel. *Transactions of the ASME-A-Engineering for Gas Turbines and Power*, 127(1):73–80, 2005.

[CMRMR01] T. Cordero, F. Marquez, J Rodriguez-Mirasol, and J. Rodriguez. Predicting heating values of lignocellulosics and carbonaceous materials from proximate analysis. *Fuel*, 80(11), 2001.

[DHS01] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. John Wiley and Sons, Inc., New York, NY, 2001.

[DS18] G.H. Davies and M. Smith. Method and arrangement for controlling fuel supply for a gas turbine, April 2018. US Patent 9,938,905.

[DSR+13] S. Das, S. Sarkar, A. Ray, A. Srivastava, and D.L. Simon. Anomaly detection in flight recorder data: A dynamic data-driven approach. In *American Control Conference (ACC), 2013*, pages 2668–2673, 2013.

[EIA17] EIA. Annual engergy outlook 2017 with projections to 2050. Technical report, U.S. Energy Information Administration, Washington DC, US, 1 2017.

[ELITG15] James Diwa Enyia, Yiguang Li, Dodeye Ina Igbong, and Isaiah Thank-God. Industrial gas turbine on-line compressor washing for power generation. *International Journal of Engineering Research & Technology*, 4, 2015.

[FFdC+17] Nina Enaasen Flø, Leila Faramarzi, Thomas de Cazenove, Odd Arne Hvidsten, Anne Kolstad Morken, Espen Steinseth Hamborg, Kai Vernstad, Guillaume Watson, Steinar Pedersen, Toine Cents, et al. Results from mea degradation and reclaiming processes at the co2 technology centre mongstad. *Energy Procedia*, 114:1307–1324, 2017.

[GAF+18] R Green, C. Allen, B. Fox, C. Holcomb, and M. Leon. Overview of digital asset management for industrial gas turbine applications. In *The Future of Gas Turbine Technology 9th International Gas Turbine Conference*. European Turbine Network, 2018.

[GE15] Digital GE. Apm-product benefits get connected brochure. *GE Digital Marketing Brochure*, 2015.

[GG19] D.A. Gutz and R.E. Goeller. System and method for controlling a fuel flow to a gas turbine engine, May 2019. US Patent 10,302,020.

[GMW19] Philip E Gill, Walter Murray, and Margaret H Wright. *Practical optimization*. SIAM, 2019.

[HA13] Raazesh Sainudiin Hamid Asgari, XiaoQi Chen. Modelling and simulation of gas turbines. *International Journal of Modelling and Control*, 2013.

[HJ13] Roger Horn and Charles Johnson. *Matrix Analysis*. Cambridge, 2013.

[HLBC16] Houman Hanachi, Jie Liu, Avisekh Banerjee, and Ying Chen. Prediction of compressor fouling rate under time varying operating conditions. *Proceedings of the ASME Turbo Expo: Turbine Technical Conference*, 2016.

[HLC14] C. Holcomb, S. Lacoursiere, and R. Collins. Low flow correction for gas turbine engine fuel valve characteristics, December 2014. US Patent 8,919,129.

[HLD+18] H. Hanachi, J. Liu, P. Ding, II Y. Kim, and C. Mechefske. Predictive compressor wash optimization for economic operation of gas turbine. *Journal of Engineering for Gas Turbines and Power*, 2018.

[HTSS11] N. Hepperle, D. Therkorn, E. Schneider, and S. Staudacher. Assessment of gas turbine and combined cycle power plant performance degradation. In *Proceedings of ASME Turbo Expo 2011*, 2011.

[IBSV15] Atul Ingle, James Bucklew, William Sethares, and Tomy Varghese. Slope estimation in noisy piecewise linear functions. *Signal processing*, 108:576–588, 2015.

[Igi17] Uyioghosa Igie. Gas turbine compressor fouling and washing in power and aerospace propulsion. *Journal of Engineering for Gas Turbines and Power*, 139, 2017.

[IPF+14] Uyioghosa Igie, Pericles Pilidis, Dimitrios Fouflias, Kenneth Ramsden, and Panagiotis Laskaridis. Industrial gas turbine performance: Compressor fouling and on-line washing. *Journal of Turbomachinery*, 136, 2014.

[IR16] A.Alexiou I. Roumeliotis, N.Aretakis. Industrial gas turbine health and performance assessment with field data. *Proceedings of the ASME Turbo Expo: Turbine Technical Conference*, 2016.

[JKVN16] Dirk C Jordan, Sarah R Kurtz, Kaitlyn VanSant, and Jeff Newmiller. Compendium of photovoltaic degradation rates. *Progress in Photovoltaics: Research and Applications*, 24(7):978–989, 2016.

[JN06] Stephen J. Wright Jorge Nocedal. *Numerical Optimization*. Springer Series in Operations Research. Springer, 2006.

[JPSM18] G. Jombo, J. Pecinka, S. Sampath, and D. Mba. Influence of fouling on compressor dynamics: Experimental and modeling approach. *Journal of Engineering for Gas Turbines and Power*, 140, 2018.

[JRS+16] J.Hall, R.Thatcher, S.Koshevets, L.Thomas, and R.Jones. Development and field validation of a large-frame gas turbine power train for steel mill gases. *Proceedings of ASME Turbo Expo 2011: Power for Land, Sea and Air*, 2016.

[JWHT13] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning*, volume 103 of *Springer Texts in Statistics*. Springer New York, New York, NY, 2013.

[K+18] Tong Seop Kim et al. Model-based performance diagnostics of heavy-duty gas turbines using compressor map adaptation. *Applied Energy*, 212:1345–1359, 2018.

[KB12] R. Kurz and K. Brun. Fouling mechanisms in axial compressors. *Journal of Engineering for Gas Turbines and Power*, 134, 2012.

[KBW09] R. Kurz, K. Brun, and M. Wollie. Degradation effects on industrial gas turbines. *Journal of Engineering for Gas Turbines and Power*, 131, 2009.

[Kia02] Philip Kiameh. *Power Generation Handbook*. Wiley, 1st edition, 2002.

[KNG92] L. J. Kerr, T. S. Nemec, and G. W. Gallops. Real-time estimation of gas turbine engine damage using a control-based kalman filter algorithm. *Journal of Engineering for Gas Turbines and Power*, 114(2):187–195, 1992.

[KSH00] Thomas Kailath, Ali H. Sayed, and Babak Hassibi. *Linear Estimation*. Prentice Hall, 2000.

[KT04] Gennady G. Kulikov and Haydn A. Thompson. *Dynamic Modelling of Gas Turbines: Identification, Simulation, Condition Monitoring and Optimal Control*. Springer, April 2004.

[Lju99] Lennart Ljung. *System Identification Theory for the User*. Prentice Hall Information and System Science Series. Prentice Hall, 1999.

[Löf04] J. Löfberg. Yalmip : A toolbox for modeling and optimization in matlab. In *In Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[Log11] Kevin Logan. Using a ships propeller for hull condition monitoring. In *ASNE Intelligent Ships Symposium IX*, 2011.

[LS86] AN Lakshminarasimha and HIH Saravanamuttoo. Prediction of fouled compressor performance using stage stacking techniques. In *ASME Symposium on Turbomachinery Performance Deterioration*, 1986.

[LWZ+13] Jay Lee, Fangji Wu, Wenyu Zhao, Masoud Ghaffari, Lnxia Liao, and David Siegel. Prognostics and health management design for rotary machinery systems - reviews, methodology and applications. *Mechanical Systems and Signal Processing*, 2013.

[Mat96] Jack D. Mattingly. *Elements of Gas Turbine Propulsion*. McGraw-Hill, 1996.

[MF10] R Misener and CA Floudas. Piecewise-linear approximations of multidimensional functions. *Journal of optimization theory and applications*, 145(1):120–147, 2010.

[MHBS13] Cyrus Meher-Homji, Andrew F. Bromley, and Jean-Pierre Stalder. Gas turbine performance deterioration and compressor washing. In *Proceedings of the 2nd Middle East Turbomachinery Symposium*, 2013.

[MHFW89] C. Meher-Homji, A. Focke, and M.B. Wooldridge. Fouling of axial flow compressors - causes, effects, detection and control. *18th Turbomachinery Symposium*, pages 55–76, 1989.

[MHT83] J Mitsui, Y Hori, and M Tanaka. Thermohydrodynamic analysis of cooling effect of supply oil in circular journal bearing. *Journal of lubrication Technology*, 105(3):414–420, 1983.

[MKGC91] G. Merrington, Oh-Kyu Kwon, G. Goodwin, and B. Carlsson. Fault detection and diagnosis in gas turbines. *Journal of Engineering for Gas Turbines and Power*, 113(2):276–282, 1991.

[MMG14] E. Mohammadi and M. Montazeri-Gh. Simulation of full and part-load performance deterioration of industrial two-shaft gas turbine. *Journal of Engineering for Gas Turbines and Power*, 136, 2014.

[MPSV10] M. Morini, M. Pinelli, P.R. Spina, and M. Venturini. Computational fluid dynamics simulation of fouling on axial compressor stages. *Journal of Engineering for Gas Turbines and Power*, 132, 2010.

[MSBB10] Michael Moran, Howard Shapiro, Daisie Boettner, and Margaret Bailey. *Fundamentals of Engineering Thermodynamics*. Wiley, 7th edition, 2010.

[M.T09] W.Gawlik M.Tavakoli, B.Vahidi. An educational guide to extract the parameters of heavy-duty gas turbines model in dynamic studies based on operational data. *IEEE Transactions on Power Systems*, 2009.

[Mur12] K.P. Murphy. *Machine Learning A Probabilistic Perspective*. MIT Press, 2012.

[MW16] Merriam-Webster. *Merriam-Webster Online Dictionary*. Merriam-Webster, 2016.

[Pic08] Clifford Pickover. *Archimedes to Hawking: laws of science and the great minds behind them*. Oxford University Press, 2008.

[R.K01] K.Brun R.Kurz. Degradation in gas turbine systems. *Transactions of the ASME*, 2001.

[RK15] Steffen Rebennack and Josef Kallrath. Continuous piecewise linear delta-approximations for bivariate and multivariate functions. *Journal of Optimization Theory and Applications*, 167(1):102–117, 2015.

[Rob12] Michael J Roberts. *Signals and systems: analysis using transform methods and MATLAB*. McGraw-Hill Higher Education, New York, NY/US, 2nd edition, 2012.

[Rue99] Michel Ruel. A simple method to determine control valve performance and its impacts on control loop performance. *CONTROL Magazine*, 1999.

[RXK12] Saeid Mokhatab Rainer X. Kurz. Important properties for industrial gas turbine fuels. *Pipeline and Gas Journal*, 239(6), 2012.

[SA05] Changdong Sheng and JLT Azevedo. Estimating the higher heating value of biomass fuels from basic analysis data. *Biomass and bioenergy*, 28(5):499–507, 2005.

[SBW08] Rajat Sekhon, Hany Bassily, and John Wagner. A comparison of two trending strategies for gas turbine performance prediction. *Journal of Engineering for Gas Turbines and Power*, 130(4):041601–041601, 2008.

[SC02] P.P. Parikh S.A. Channiwala. A unified correlation for estimating hhv of solid, liquid and gaseous fuels. *Fuel*, 81(8), 2002.

[SCBS09] D Sánchez, R Chacartegui, J A Becerra, and T Sánchez. Determining compressor wash programmes for fouled gas turbines. *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy*, 223(4):467–476, 2009.

[Sch14] Dorin Scheianu. Methods and results in remote monitoring and diagnosing a fleet of industrial gas turbines. In *ASME Turbo Expo 2014: Turbine Technical Conference and Exposition*, volume 6, page V006T06A014, Dusseldorf, GE, 2014.

[Sko09] Sigurd Skogestad. *Chemical and Energy Process Engineering*. CRC Press, 1st edition, 2009.

[SMP89] A Stamatis, K Mathioudakis, and KD Papailiou. Adaptive simulation of gas turbine performance. In *ASME 1989 International Gas Turbine and Aeroengine Congress and Exposition*. American Society of Mechanical Engineers, 1989.

[SPB08] Alessandro Magnani Stephen P. Boyd. Convex piecewise-linear fitting. *Optimization and Engineering*, 2008.

[SR15] Donald L Simon and Aidan W Rinehart. A model-based anomaly detection approach for analyzing streaming aircraft engine measurement data. In *ASME Turbo Expo 2014: Turbine Technical Conference and Exposition*. American Society of Mechanical Engineers Digital Collection, 2015.

[SRC01] H.I.H Saravanamutto, G.F.C. Rogers, and H. Cohen. *Gas Turbine Theory*. Pearson Prentice Hall, 5th edition, 2001.

[Str88] Gilbert Strang. *Linear Algebra and its Applications*. Sauders HBJ, 3rd edition, 1988.

[TH09] Jerome Friedman Trevor Hastie, Robert Tibshirani. *Elements of Statistical Learning*. Springer, 2009.

[Tib96] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[TMBK14] E. Tsoutsanis, N. Meskin, M. Benammar, and K. Khorasani. A component map tuning method for performance prediction and diagnostics of gas turbine compressors. *Applied energy*, 135:572–585, 2014.

[TS94] Koji Tsunokawa and Joseph L Schofer. Trend curve optimal control model for highway pavement maintenance: Case study and evaluation. *Transportation Research Part A: Policy and Practice*, 28(2):151–166, 1994.

[TV12] Alejandro Toriello and Juan Pablo Vielma. Fitting piecewise linear continuous functions. *European Journal of Operational Research*, 219(1):86–95, 2012.

[VCZ+04] A Valero, L. Correas, A. Zaleta, A. Lazzaretto, V. Verda, M. Reini, and V. Rangel. On the thermoeconomic approach to the diagnosis of energy system malfunctions part 2. malfunction definitions and assessment. *Journal of Energy*, 2004.

[Vol14] Allan J. Volponi. Gas turbine engine health management: Past, present and future trends. *Journal of Engineering for Gas Turbines and Power*, 2014.

[VP12] Mauro Venturini and Nicola Puggina. Prediction reliability of a statistical methodology for gas turbine prognostics. *Journal of Engineering for Gas Turbines and Power*, 134(10):101601–101601, 2012.

[WAL03] Michael Wall, Rechtsteiner Andreas, and Rocha Luis. *A Practical Approach to Microarray Data Analysis*, volume 103. Los Alamos National Laboratory, 2003.

[WHTT17] Fei-Long Wang, Ya-Ling He, Song-Zhen Tang, and Zi-Xiang Tong. Parameter study on the fouling characteristics of the h-type finned tube heat exchangers. *International Journal of Heat and Mass Transfer*, 112:367–378, 2017.

[W.R83] W.Rowen. Simplified mathematical representations of heavy-duty gas turbines. *Journal of Engineering for Power Transactions of the ASME*, 1983.

[Zab80] T. Zaba. Losses in gas-turbines due to deposits on the blading. *Brown Boveri Review*, 67:715–722, 1980.