INSTITUTO POLITÉCNICO NACIONAL
Escuela Superior de Ingeniería Mecánica y Eléctrica
Sección de Estudios de Posgrado e Investigación
Unidad Zacatenco

Development and Evaluation of Algorithms for Gas Turbine Condition Monitoring and Diagnostics

Tesis para obtener el grado de:
Doctor en Ciencias en Ingeniería Mecánica

Presenta:
M. en C. Juan Luis Pérez Ruiz

Directores de tesis:
Dr. Igor Loboda
Dr. Miguel Toledo Velázquez

Ciudad de México, México, Diciembre 2018

Resumen

Las turbinas de gas son máquinas muy complejas y potencialmente poco fiables. Con el fin de reducir los costos de mantenimiento y mantener una alta confiabilidad, en las últimas décadas se han desarrollado muchos sistemas avanzados de monitoreo de condiciones y diagnóstico. El éxito de estos sistemas depende en gran medida de la perfección del software de diagnóstico, en particular, de la precisión de las decisiones diagnósticas. Esto se puede lograr a través de investigaciones continuas y la validación de diferentes algoritmos de diagnóstico. Siguiendo esta idea, el objetivo de la presente tesis es desarrollar y evaluar diferentes algoritmos aplicados al monitoreo y diagnóstico de turbinas de gas. Se desarrollan cuatro metodologías principales. La primera metodología trata de la creación de un algoritmo de monitoreo en línea basado en la estimación de variables de motor no medidas, como la eficiencia y la potencia del eje. La alta precisión de las estimaciones de cantidades no medidas y sus desviaciones para diversas fallas y condiciones de operación variables confirma que la metodología propuesta puede utilizarse en sistemas de monitoreo en línea reales.

La segunda metodología está destinada a evaluar técnicas de diagnóstico en condiciones de falla variable. Para ese propósito, se propone un principio de clasificación de falla variable para mejorar la representación de escenarios de falla real. Los extensos cálculos de comparación revelan que cualquiera de las técnicas analizadas es una buena alternativa para la identificación de fallas en turbinas de gas. Sin embargo, se encuentra una gran influencia de las clasificaciones de fallas y el límite de severidad de fallas en el nivel de precisión de diagnóstico. La implementación de un nuevo límite para múltiples clases de fallas y errores reales en desviaciones hace que la precisión de diagnóstico del motor se acerque más a lo que ocurre en la práctica.

La tercera metodología trata de una evaluación comparativa de un enfoque de diagnóstico de turbina de gas basado en datos a través de una plataforma especial llamada ProDiMES. La plataforma presenta un enfoque público, en el que diferentes investigadores pueden verificar y comparar sus algoritmos para las etapas de diagnóstico de extracción de características, detección de fallas e identificación de fallas. La comparación de las métricas de ProDiMES de diagnóstico final obtenidas bajo las condiciones óptimas seleccionadas con las métricas de otras soluciones de diagnóstico muestra que el algoritmo propuesto es una herramienta prometedora para los sistemas de diagnóstico de turbinas de gas.

La última metodología compara dos enfoques de diagnóstico de turbinas de gas, basados en datos y basados en física. El diagnóstico para el primer enfoque se realiza en el espacio de desviaciones de las variables monitoreadas, mientras que para el segundo enfoque en el espacio de desviaciones de los parámetros de salud estimados. La metodología se centra en utilizar las condiciones de comparación correctas y considerar los casos de comparación múltiple de opciones de diagnóstico y posibles escenarios de fallas. Los resultados de todos los casos muestran que ambos enfoques reconocen igualmente bien cada una de las fallas del motor y que el rendimiento de diagnóstico promedio también es similar. Por lo tanto, desde el punto de vista de la precisión diagnóstica, la transformación al espacio de desviaciones de los parámetros de salud estimados no tiene ningún efecto y los enfoques son iguales. Para seleccionar un enfoque adecuado para un sistema de monitoreo de turbina de gas real, se deben considerar otros criterios.

Abstract

Gas turbines are very complex and potentially unreliable machines. In order to reduce maintenance costs and maintain high reliability, many advanced condition monitoring and diagnostic systems have been developed in the past decades. The success of these systems strongly depends on the perfection of diagnostic software, particularly, on the accuracy of diagnostic decisions. This can be achieved through continuous investigations and validation of different diagnostic algorithms. Following this idea, the aim of the present thesis is to develop and evaluate different algorithms applied to gas turbine monitoring and diagnostics. Four main methodologies are developed. The first one deals with the creation of an online monitoring algorithm based on the estimation of unmeasured engine variables such as efficiencies and shaft power. The high accuracy of estimations of unmeasured quantities and their deviations for a diverse faults and varying operating conditions confirms that the proposed methodology can be utilized in real on-line monitoring systems.

The second methodology is intended for evaluating diagnostic techniques under variable fault conditions. For that purpose, a principle of a variable fault classification is proposed to enhance the representation of real fault scenarios. Extensive comparison calculations reveal that any of the analyzed techniques is a good alternative for gas turbine fault identification. However, a great influence of fault classifications and fault severity boundary on the level of diagnostic accuracy is found. The implementation of a new boundary for multiple fault classes and real errors in deviations makes the engine diagnostic accuracy closer to what occurs in practice.

The third methodology deals with a benchmarking analysis of a data-driven gas turbine diagnostic approach through a special platform called ProDiMES. The platform presents a public approach, at which different investigators can verify and compare their algorithms for the diagnostic stages of feature extraction, fault detection, and fault identification. The comparison of the final diagnostic ProDiMES metrics obtained under the selected optimal conditions with the metrics of other diagnostic solutions shows that the proposed algorithm is a promising tool for gas turbine diagnostic systems.

The last methodology compares two gas turbine diagnostic approaches, data-driven and physics-based. The diagnosis for the first approach is realized in the space of monitored variable deviations while for the second approach in the space of estimated health parameter deviations. The methodology focuses on using the correct comparison conditions and considering multiple comparison cases of diagnostic options and possible fault scenarios. The results of all the cases show that both approaches equally well recognize each of the engine faults and the average diagnostic performances are also similar. Thus, from the point of view of diagnostic accuracy, the transformation to the space of estimated health parameter deviations has no effect and the approaches are equal. To select a proper approach for a real gas turbine monitoring system, other criteria should be considered.

Preface

The present work is an integrated article-based thesis embracing different peer-reviewed journal articles and conference papers (published or under review) related to the development and evaluation of algorithms in the areas of condition monitoring and diagnostics. The information of the articles was modified to have a consistent thesis structure. Permissions to reproduce copyrighted material can be found in Appendix 1.

Chapter 3 is a modified version of the paper entitled "Estimation and Monitoring of Unmeasured Gas Turbine Variables" published in *Transactions of the Canadian Society for Mechanical Engineering* (in press). (https://doi.org/10.1139/tcsme-2017-0009).

Chapter 4 is a modified version of the paper entitled "Evaluation of Gas Turbine Diagnostic Techniques under Variable Fault Conditions" published in *Advances in Mechanical Engineering,* Vol.9, Issue 10, 2017. (https://doi.org/10.1177/1687814017727471).

Chapter 5 is a modified version of the paper entitled "A Benchmarking Analysis of a Data-Driven Gas Turbine Diagnostic Approach" published in *ASME Turbo Expo* 2018, Oslo, Norway. (https://doi:10.1115/GT2018-76887)

Chapter 6 is a modified version of the paper entitled "Comparative analysis of two gas turbine diagnosis approaches" which has been submitted to *ASME Turbo Expo* 2019 and is under review.

In all of the aforementioned works, the author of this thesis is a main contributor in the areas of methodology development, information analysis, software programming, and article writing and edition.

Agradecimientos

Ante todo, quiero agradecer a Dios por permitirme llegar a hasta este punto de mi formación académica y por todas las bendiciones que me ha regalado en la vida, especialmente por su perdón y misericordia a través del sacrificio de Jesucristo en la cruz, “*Porque de tal manera amó Dios al mundo, que ha dado a su Hijo unigénito, para que todo aquel que en él cree, no se pierda, mas tenga vida eterna”* (Juan 3:16, RVR 1960).

Este trabajo no hubiera sido posible sin la ayuda incondicional, amor, paciencia y consejos de mis padres Juan Luis Pérez Urbina y María Fanny Ruiz Álvarez, y de mis hermanas, Eunice Julieta Pérez Ruiz y Citlally Liliana Pérez Ruiz. Los amo.

Quiero expresar mi gratitud y admiración al Dr. Igor Loboda por el conocimiento y ayuda que me ha brindado a lo largo de los años. No sólo lo considero un mentor y consejero sino también un amigo.

Agradezco profundamente al Dr. Miguel Toledo Velázquez por sus enseñanzas y por darme la confianza y la oportunidad de entrar al programa de doctorado bajo su dirección.

Al Dr. Georgiy Polupan, Dr. Florencio Sánchez Silva y Dr. Ignacio Carvajal Mariscal gracias por supervisar y ayudar a mejorar este trabajo.

Agradezco al Instituto Politécnico Nacional y al Consejo Nacional de Ciencia y Tecnología por brindarme los recursos económicos y materiales para obtener este grado académico.

Gracias a todos mis compañeros por brindarme su amistad durante toda mi estancia en SEPI:

Alexander Vargas Almeida
Daniel Zaldívar Cabrera
Eduardo Ríos Urbán
Elisa Sánchez Cruces
Felipe de Jesús Cisneros Azuara
Gerardo Blancas Hernández
Gonzalo Cauich Lemus
Hugo Solares Cruz
Isaac Juárez Acosta
Iván González Castillo
Jaime García Díaz
Jesús Ramírez Solano
Jonatán Mario Cuéllar Arias
José Iván Algarín Zapién
Karen Anaid Rendón Cortés
Luis Angel Miró Zárate
Omar Magaña Sánchez
Pablo Eduardo Ruiz Ortega
Pablo Javier Ruiz López
Salvador Felipe Altamirano
Santiago Jiménez Sosa
Víctor Manuel Hernández Velázquez
Víctor Manuel Pineda Molina
Willams Leonardo Rogel Ramírez

Content

List of Figures

List of Tables

Chapter 1: State of Art and Research Objectives

1.1 INTRODUCTION

Gas turbines are very sophisticated and costly systems that have been used in the past decades for different industrial applications due to the capacity to produce great amount of energy and because of their high efficiencies [1]. However, due to the elevated complexity, gas turbines are considered potentially unreliable machines. They contain complex systems and subsystems and thousands of components, many of them exposed to high temperatures, pressures and rotations. For that reason, any gas turbine will experience degradation over its useful life. The mechanisms causing deterioration and potential fault events are: dirt deposits, fouling, corrosion and erosion of blade and vanes, foreign and domestic object damage (FOD/DOD), malfunction of inlet guide vanes (IGV) and variable guide vanes (VGV), tip clearance increase, filter clogging, plugged nozzles, cracked rotor disc or blade, cracking of liners, etc. [2,3]. Since gas turbine deterioration has a direct impact on fuel costs, it is essential that all the gas turbine systems and subsystems operate as efficient as possible. For example, in a gas turbine based combined cycle, the percent of life cycle costs for fuel (75%) is by far higher than rest (6% for operation and maintenance and 19% for capital costs). According to study [4], in the past decade the annual fuel costs of a 934 MW combined cycle plant and a 30 aircraft fleet could reach about 200 and 250 million USD, respectively. For the case of the combined cycle plant, this amount could easily reach 6 or 7 billion USD over the entire engine life cycle. Considering this, it is imperative to investigate and control all the forms of gas turbine deterioration.

Over the last years, different studies have addressed the problem of gas turbine deterioration. Performance deterioration in industrial gas turbines is discussed in [5] and different causes of performance degradation in aircraft engines are addressed in [6]. Studies [7] and [8] give a detailed treatment on gas turbine deterioration. Meher-Homji and Bromley [9] provide a comprehensive practical treatment of the causes, effects, and control of compressor fouling as well as the compressor washing approaches and technologies. Burnes and Kurz [10] present a detailed study of degradation and its effects on overall gas turbine performance with an emphasis on recoverable and non-recoverable degradation. An extensive study of gas turbine performance deterioration given in [4] addresses the topics of recoverable and non-recoverable deterioration, effects of site conditions on gas turbine performance, mechanical deterioration, trending performance data and transient analysis for performance deterioration.

According to Meher-Homji et al. [4], gas turbine performance deterioration (see Figure 1.1) can be classified as: 1) recoverable (the deterioration can be eliminated during the engine operation by taking actions such as compressor washing), 2) unrecoverable (the deterioration can be eliminated only if an overhaul is considered but not during engine operation), 3) permanent (the deterioration is still present despite major overhauls). Hanachi et al. [3] classify the causes of gas turbine performance deterioration into two groups: natural causes and physical deterioration (see Figure

1.2). The first type occurs when an engine operating condition deviates from the design point producing degradation no matter if the engine components are in a healthy state. This natural deterioration has a temporary effect that can be reverted when the engine returns to the operation of its design condition. The second type considers health condition degradation of the parts (structural and recoverable deterioration) and it cannot be reverted unless an action is taken. In any case, performance deterioration causes negative effects such as an increase of fuel consumption, elevated exhaust gas temperature, and a decrease of thermal efficiency and power.

Table 1.1 lists common gas path degradations, their symptoms and effects on GT engine performance.

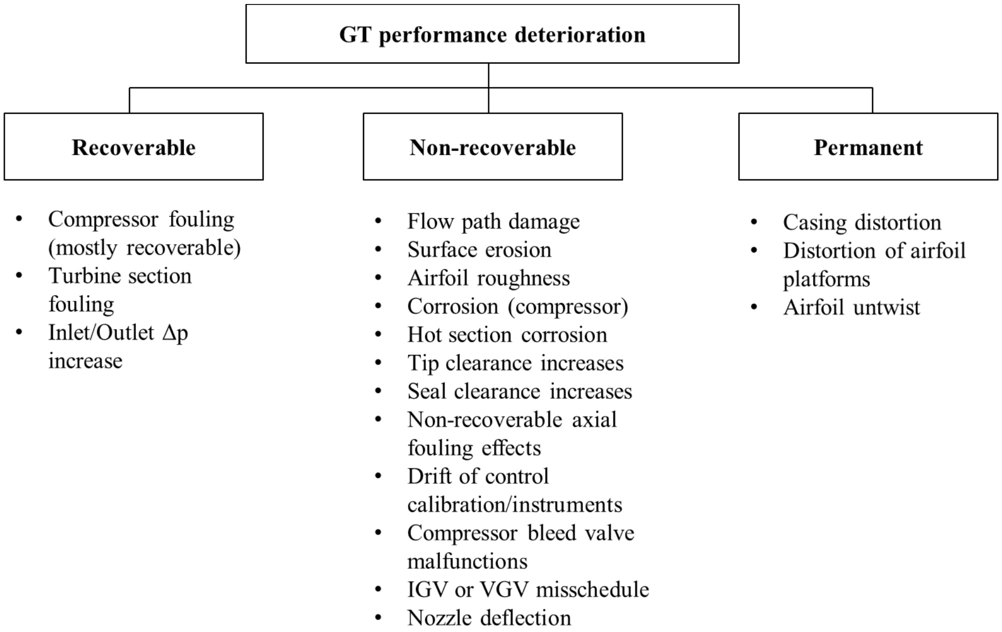

GT performance deterioration
Recoverable
Non-recoverable
Permanent
• Compressor fouling (mostly recoverable)
• Turbine section fouling
• Inlet/Outlet Δp increase
• Flow path damage
• Surface erosion
• Airfoil roughness
• Corrosion (compressor)
• Hot section corrosion
• Tip clearance increases
• Seal clearance increases
• Non-recoverable axial fouling effects
• Drift of control calibration/instruments
• Compressor bleed valve malfunctions
• IGV or VGV misschedule
• Nozzle deflection
• Casing distortion
• Distortion of airfoil platforms
• Airfoil untwist

Figure 1.1 Categories of GT performance deterioration (adapted from [4]).

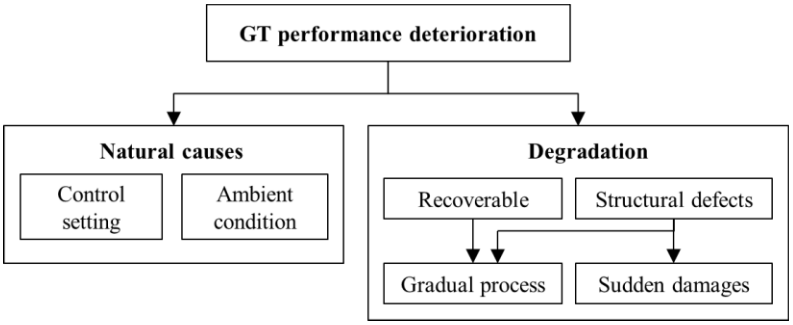


Figure 1.2 Causes of gas turbine performance degradation (adapted from [3]).

GT deterioration can be also produced by mechanical problems and manufacturing variations [4]. Some factors of mechanical degradation include wear in bearings and seals, coupling problems, excessive vibration and noise, problems in the lube oil system, etc. Some sources of manufacturing deviations are related to excessive surface roughness and irregular coating, excessive radial clearance between blade tip and casing and between seal diameter of diaphragms and the disk seal land, reductions in blade chord and thickness, incorrect axial position, etc.

Table 1.1 Gas path degradations and their effects on engine performance [3]

Section	Fault mode	Component level symptoms	Effects on GT performance
Intake	-Filter clogging	-Drop in filter differential pressure	-Reduction of mass flow -Increase of EGT for a given load -Loss of power -Increase of heat rate
Compressor	-Compressor fouling -Blade/Vane erosion -Compressor corrosion	-Decrease in pressure ratio -Loss of isentropic efficiency -Reduction of flow capacity -Elevation of discharge temperature -Reduction of surge margin	-Increase of EGT for a given load -Loss of power -Loss of thrust -Increase of heat rate
	-Tip clearance increase	-Decrease in pressure ratio -Loss of isentropic efficiency -Reduction of flow capacity -Leakage of the compressed air -Reduction of surge margin	
	-IGV/VGV malfunction	-Leakage of the compressed air -Loss of isentropic efficiency -Variation in the flow rate -Compressor stall and surge	
	-FOD	-Drop in pressure ratio -Loss of isentropic efficiency -Drop in the flow rate -Compressor stall and surge	-Rise of EGT -Rotor acceleration -Loss of thrust -Increase of heat rate -Engine blowout
Combustion chamber	-Corrosion and cracking of liners	-Drop in effectiveness of cooling air -Loss of combustor efficiency	-Increase of EGT for a given load -Loss of power -Loss of thrust
	-Clogging of fuel injectors -Malfunction of fuel delivery system	-Variation in fuel flow rate -Delayed or incomplete combustion -Loss of combustion efficiency	-Variation of EGT -Non-uniform EGT profile -Loss of power -Loss of thrust -Increase of heat rate
Turbine	-Nozzle deflection -Turbine erosion	-Drop in turbine pressure ratio -Increase of mass flow capacity -Loss of isentropic efficiency	-Increase of EGT for a given load -Loss of power -Loss of thrust -Increase of heat rate -Increase of mass flow rate
	-Tip clearance increase -Seal clearance increase	-Drop in pressure ratio -Increase in mass flow rate -Leakage of cooling air -Loss of isentropic efficiency	
	-Turbine fouling -Turbine corrosion	-Drop in turbine pressure ratio -Decrease in mass flow capacity -Loss of isentropic efficiency -Reduction of surge margin	-Increase of EGT for a given load -Loss of power -Loss of thrust -Decrease of heat rate
Exhaust	-Surface erosion/corrosion	-Increase of back pressure	-Increase of EGT -Loss of power -Loss of thrust

The effect of manufacturing deviations on performance of axial flow compressor blading is addressed, for example, in [11]. Other sources of deterioration consider excessive drop in inlet filter differential pressure, increased mechanical losses, internal losses, stator nozzle plugging, and overboard leakage.

Throughout the last decades, gas turbine maintenance programs have been created to solve the aforementioned problems. These programs have evolved from simplified to very complex strategies. Figure *1.3* presents these strategies that can be split into four categories: breakdown maintenance (BM), preventive maintenance (PM), condition-based maintenance (CBM) and prognostic health management (PHM) [12–14]. The BM strategy is the simplest one in which no actions are taken until the engine presents a failure and necessary corrections are implemented. This approach is based on empirical knowledge and does not guarantee the engine safety producing high risk, high maintenance costs and overtime. In the PM strategy, planned, coordinated and scheduled inspections and overhauls are carried out helping to prevent faults. The problems with this approach are the excessive overhauls, time consuming and low efficiency due to the neglect of actual engine health condition. Furthermore, considering the differences in operating environment and configuration of each gas turbine, it is difficult to determine the optimal maintenance interval [15,16].

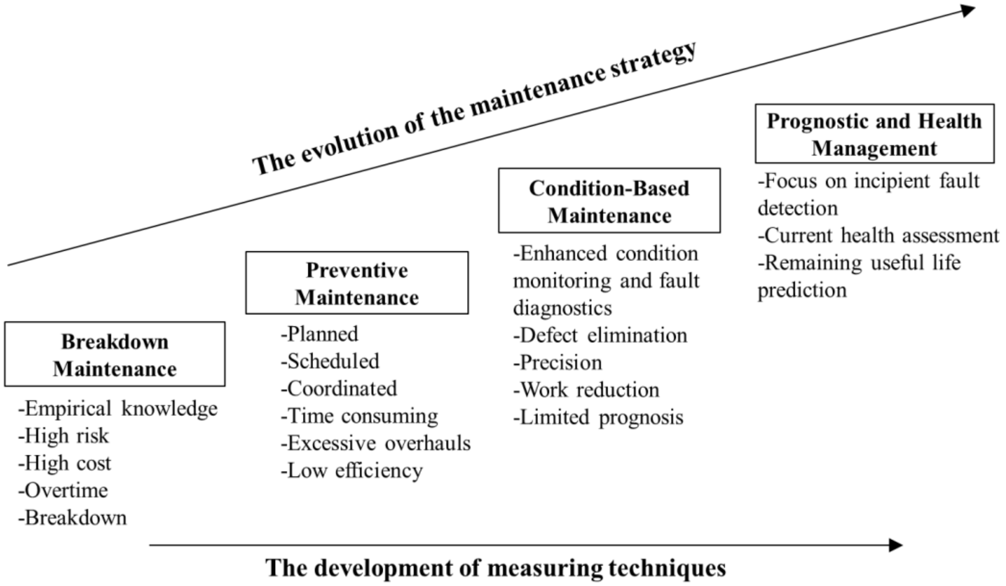

The evolution of the maintenance strategy
Breakdown Maintenance
-Empirical knowledge
-High risk
-High cost
-Overtime
-Breakdown
Preventive Maintenance
-Planned
-Scheduled
-Coordinated
-Time consuming
-Excessive overhauls
-Low efficiency
Condition-Based Maintenance
-Enhanced condition monitoring and fault diagnostics
-Defect elimination
-Precision
-Work reduction
-Limited prognosis
Prognostic and Health Management
-Focus on incipient fault detection
-Current health assessment
-Remaining useful life prediction
The development of measuring techniques

Figure 1.3 The development of GT maintenance strategies (adapted from [12]).

The CBM strategy collects, processes, and intelligently interprets the engine measurements obtained through sensors to assess the actual engine condition without interrupting the normal operation of the engine. Based on the engine condition, maintenance actions can be taken depending on the type and severity of the fault. The advantages of using CBM as a separate strategy or combined with PM are: the reduction of maintenance costs, an increase of system reliability, an enhanced condition monitoring and fault diagnostics, defect elimination, life extension of engine components, precision, work reduction. Different methodologies related to CBM have been developed using data-driven and physic-based modeling to improve gas turbine condition monitoring and diagnostics [17–19]. However, some drawbacks of the CBM are: the high installation costs, limited prognosis because the behavior of the engine cannot be predicted accurately by using limited field monitoring data, the systems need to be properly installed and operated by trained staff, and the implementation of the systems increases the parts that require verification and maintenance [12,14].

The PHM is a new maintenance strategy developed in the recent years intended to address the limitations of CBM. Figure 1.4 displays the CBM/PHM process. PHM involves the prediction of future engine behavior, including remaining useful life, in terms of current operating state and the scheduling of required maintenance actions to maintain the engine health [13]. PHM is viewed as an integrated process since it comprises the stages of anomaly detection, diagnosis and prognosis. An effective PHM system is expected to detect an anomaly condition of the engine in an early stage. Meanwhile, the detected incipient failure should be monitored and predicted to prevent the occurrence of a potential damage.

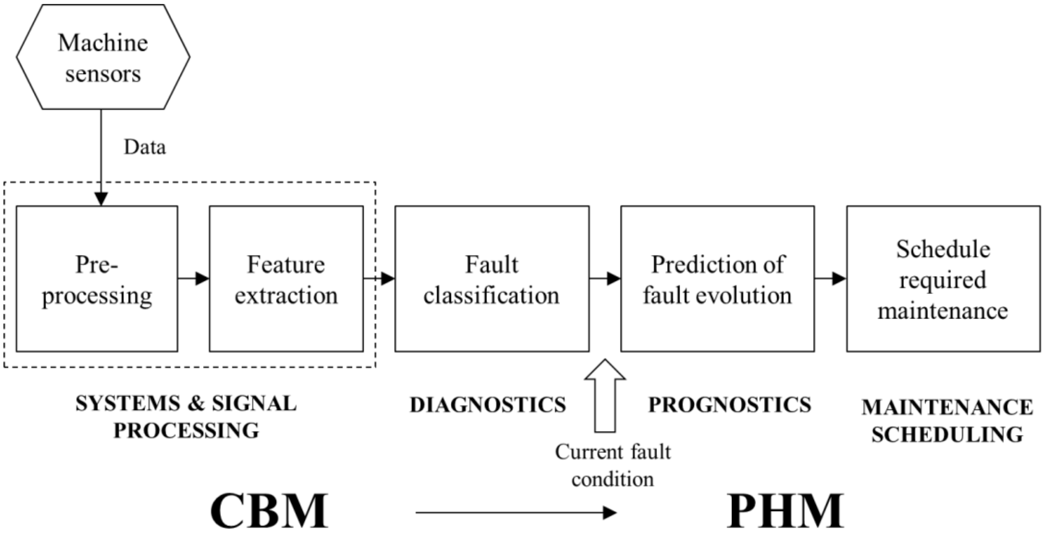


Figure 1.4 The CBM/PHM process (adapted from [13]).

1.2 GAS TURBINE CONDITION MONITORING AND DIAGNOSTICS: AN INSIGHT INTO THE AREAS OF RESEARCH

Important aspects of gas turbines such as reliability, safety, maintenance and operating costs are strongly affected by failures and deterioration. To mitigate these problems, diagnostic systems based on CBM/PHM strategies are widely used [14]. These systems include different approaches such as thermography, boroscopy inspection, vibration and acoustic analysis, diagnostics of fuel and oil systems, wear debris analysis, and gas path analysis (GPA), which has been widely used in the field of gas turbine diagnostics. The systems based on GPA collect, filtrate, and intelligently analyze measured gas path variables (temperatures, pressures, rotation speeds, fuel flow rate, etc.) to monitor the engine health state, identify initial defects and predict future problems. The main advantage of GPA is that there is no need for shutdowns or engine disassembly since they can operate in real time.

The stages of a complete diagnostic process include: data acquisition, data processing, feature extraction, monitoring (fault detection), detailed diagnosis (fault identification) and prognosis [13,20]. Each stage depends on independent, deep and complex algorithms that require years of work. However, the integration of each of them into a complete, efficient and reliable system is one of the most important goals of gas turbine diagnostics. When a diagnostic algorithm is developed, its structure and efficiency depends on the selection and interaction of many factors. An example of these factors can be seen in Figure 1.5. On the basis of accumulated knowledge and experience, researchers choose adequate factors and develop their own diagnostic algorithm. To be optimal, an algorithm should take into account all peculiarities of a given engine, its application, and other diagnostic conditions. Thus, it is not likely that the algorithm be optimal for other engines and applications. As a result, every monitoring system needs the development of appropriate diagnostic algorithms [21].

In many applications, especially in aviation, there are increasingly strict standards to develop reliable monitoring and diagnostic systems. This can be achieved by investigating and improving the algorithms related to such systems. Considering this necessity, the present thesis proposes the development and evaluation of algorithms related to different areas of condition monitoring and diagnostics. Each proposed algorithm is developed based on the selection of specific influence factors as illustrated in Figure 1.5. For example, an algorithm can address the stage of fault identification using physics-based models of a turbofan engine, single faults and multi-point diagnostic analysis. In later chapters, Figure 1.5 is repeated for each algorithm to specify their influence factors.

The following subsections gives an insight into the areas of research to be addressed in this thesis that include diagnostic models, data-driven and physics-based approaches, diagnostics based on measured, and techniques employed for fault diagnostics.

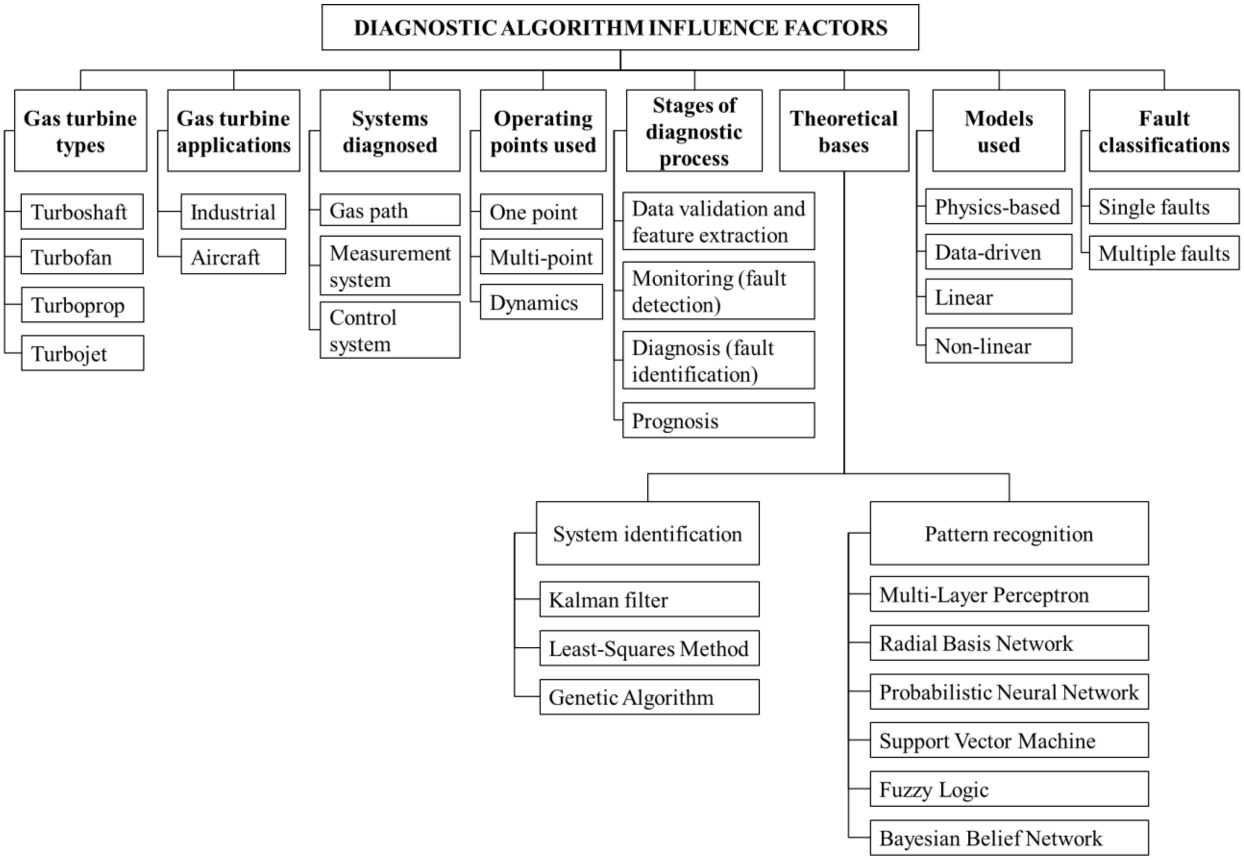


Figure 1.5 Factors that influence the development of diagnostic algorithms (adapted from [21]).

1.2.1 Data-driven and physics-based diagnostic approaches

Gas turbine diagnostic methods can fall into two general approaches: physics-based and data-driven [3,20]. Book [13] by Vachtsevanos et al. covers multiple aspects of both approaches applied to engineering systems. A good theoretical analysis of this approach applied to gas turbine diagnosis can be found, for example, in study conducted by Volponi [22]. In the physics-based approach, system identification techniques employ engine thermodynamic models, which are widely used worldwide to estimate engine faults. Within this approach fault parameters are firstly estimated by system identification methods. These parameters allow us to determine a current engine state [23,24] because great values of the fault parameters point out the engine faulty components. In this way, final diagnostic decisions can be taken without the constraint of a pre-defined fault classification as with the data-driven approach. However, due to the inherent inaccuracy of engine physics-based models, considerable errors of estimated health parameters are observed [25]. Also, a thermodynamic model is not always available in practice.

The data-driven diagnostic approach relies on direct engine measurements as inputs. In a feature extraction stage, the measurements are usually converted into deviations (a.k.a. deltas) and pattern recognition techniques are applied to identify a current engine health state [26–28]. The data-driven approach has the potential advantage of not depending on a physics-based (thermodynamic) model, thus avoiding the problem of model impact on final diagnostic decisions. Excluding the thermodynamic model does not mean a complete rejection of simulation. Nevertheless the models used are of a black-box type and do not need detailed information of engine components that usually is a manufacturer’s property. A disadvantage of this approach is the inability of black-box models to simulate engine faults. Since physical fault simulation on a test-bed is complex, expensive and even dangerous, the possibility to form an extensive real fault classification is questionable. Hence, a thermodynamic model is still used in investigations to generate the data of faulty engines, but in practice it can be easily substituted by real fault data if available.

Multiple diagnostic algorithms belonging to the mentioned approaches have been widely utilized so far and comparative studies of these algorithms are known. Some particular recommendations on the selection of the best algorithm can be found, for example, in [26,29]. However, a side-to-side evaluation between both approaches has not been carried out before by researchers in the community of healthy engine management. A comparative study in the area of fault identification for example, can reveal if the level of fault diagnosis accuracy is similar or completely different in both approaches.

1.2.2 Diagnostics based on measured gas path variables

So far, diagnostic techniques embrace all the primary systems of GT, namely, gas path, transmission, measurement system, control system, fuel system, oil system, starting systems, etc. Among all these techniques, the algorithms that analyze gas path variables can be considered as essential [17,30]. They provide a deep insight into the performance of engine components such as compressors, burner and turbines and reveal different mechanisms of degradation. Some measured gas path variables, called operating conditions, are used to set an engine’s steady state operating point. The remaining measurements are available for engine monitoring and are typically called monitored variables. When the GTE components deteriorate, the monitored variables change accordingly. For this reason, these variables are used to detect and identify GTE faults, e.g., compressor fouling, erosion, foreign object damage, etc. as shown in Figure 1.6. The mechanisms and impact of the faults are described in [4].

The use of measured variables for Gas Path Analysis (GPA) is well known. However, the analysis of unmeasured variables can improve the diagnostic accuracy by providing new input information that is not usually taken into account, thus making the diagnostic process more reliable. These

unmeasured variables can be handful for characterizing overall engine performance and detecting some degradation mechanisms and engine faults that can be hidden for a standard analysis based on the measured variables. Examples of computing and monitoring of these variables include, but are not limited to, engine power [31] or thrust [32,33], turbine inlet temperature [34], compressor and turbine efficiencies [35–37], and compressor air mass flow [7]. Knowledge of these parameters helps in monitoring of engine mission, integrity, and overall efficiency, and also allows a more detailed diagnosis of engine components.
Table 1.2 lists typical measured and unmeasured variables used in different types of engines.

A natural mode to compute unmeasured variables at steady states and transients consists in the application of the well-known GPA, described in detail in [38]. This approach relies on both nonlinear and linear gas path models. One of the latest examples of thermodynamic model-based estimation of unmeasured variables is given in [31]. In addition to model calibration (identification), the authors employ the Bayesian inference approach to separately estimate the power of a gas turbine and a steam turbine joined by a common shaft. A thermodynamic model needs detailed performance maps of all engine components, but these maps are the property of engine manufacturers and are not readily available to engine operators. Furthermore, the studies [31,35,39–41] show that a thermodynamic model and its identification procedure present complex software, which requires significant computer resources. Therefore, the thermodynamic model is not suitable for the real-time computing the unmeasured gas path variables within an online monitoring system.

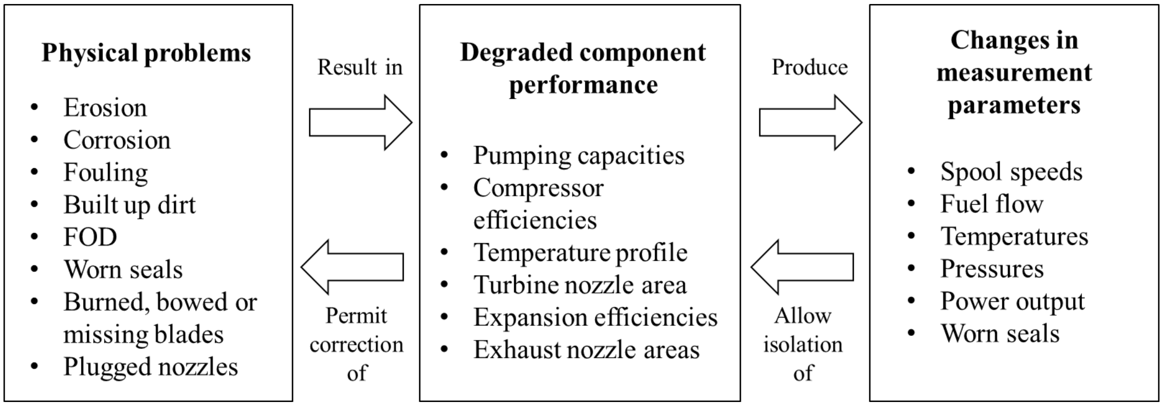


Figure 1.6 Gas turbine fault diagnostics approach (adapted from [42]).

To meet the computational requirements, the GPA uses different linear models originating from the thermodynamic model. To find unknown fault parameters at steady states and transients, Kalman filter (KF) is often applied to linear models [38]. For computing unmeasured variables, two main linear models of the filter (system model and measurement model) are complemented by the model for these variables [43,44]. Studies [32,33] on turbofan engine thrust estimation present good examples of KF applications. Although intended for online application, the KF methodology has some limitations. First, in addition to the inadequacy of an original thermodynamic model, linearization and approximation errors affect the accuracy of the estimated parameters. Second, the implementation of this methodology requires prolonged development time and the efforts of qualified diagnosticians. Apart from a KF-based diagnostic algorithm needing to be developed and verified, a reliable thermodynamic model has to be developed first.

Although many online diagnostic systems have been developed by gas turbine manufacturers, specialized laboratories, and universities, the use of these systems is not a standard practice for production engines. A probable explanation for this is that engine owners and operators do not rely completely upon diagnostic decisions, do not believe in the system’s effectiveness, and, as a result, do not want to invest in a system. However, a wider use of diagnostic systems would allow the acquisition of new data about engine degradation and sensor malfunctions, resulting in improved diagnostic software.

Table 1.2 Typical measured and unmeasured variables used in different types of engines [3].

Type of variable	Performance parameter	Unit	Engine type
Measured	EGT	°C	All types
	Fuel flow	kg/s	All types
	High pressure spool speed	rpm	All types
	Low pressure spool speed	rpm	Two-spool GT
	Intermediate pressure spool speed	rpm	Three-spool GT
	Compressor discharge pressure	kPa	All types
	Compressor discharge temperature	°C	All types
	Turbine inlet temperature	°C	All types
	Torque	Nm	Turboshaft/ Turboprop
Unmeasured	Power	kW	Stationary/ APU
	Thrust	kN	Aero engine
	Specific fuel consumption	kg/kJ	Stationary
	Thrust specific fuel consumption	g/(kN·s)	Aero engine
	Air flow	kg/s	All types
	Exhaust gas flow	kg/s	All types
	Exhaust gas velocity	m/s	Aero engine
	Heat rate	kJ/kWh	Stationary
	Thermal efficiency	%	All types

Given the above reasoning, an easy-to-develop reliable online system appears to be a desirable option. Since the 1970s, simple thermodynamic relationships for computing compressor and turbine efficiencies have attracted the attention of diagnosticians [35,36]. These relationships ideally meet the requirements of an online system, but can be used only for the efficiencies of the specific components. Additionally, the necessary component input and output parameters are not always available. Reference [37] provides another algorithm suitable for an online system. The authors estimate an unmeasured compressor air flow at steady states by means of measured parameters and a small set of nonlinear thermodynamic equations. The study [34] also follows the idea of simplified algorithms for an online system. In an effort to estimate a turbine blade lifetime, unmeasured boundary conditions (heating temperature and heat transfer coefficient) are determined at steady states, using thermodynamic relations and data-driven models. The study proves the accuracy of these conditions and the resulting lifetime estimates. The aforementioned studies only deal with unmeasured gas turbine variables. As they primarily depend on the engine operating conditions, and the degradation influence is smaller, features with a greater diagnostic value are required. The study [45] extends the feature (deviation) extraction on the unmeasured variables. First, to examine the proposal, unmeasured compressor and turbine efficiencies were calculated according to the known thermodynamic relationships like it is done in [35,36], and the turbine blade heating temperature was determined by the model developed in [34]. For these variables, the deviations were calculated then. After the validation of the deviation accuracy on simulated and real data, it was concluded that the new features can be used in a gas turbine monitoring system.

As shown above, the algorithms used in [34,45] for computing the unmeasured variables and their deviations are suitable for use in an online monitoring system because they compute useful diagnostic features and are accurate and simple. Nevertheless, one limitation remains: the variables are calculated by individual algorithms. This is because each algorithm comprises a specific set of thermodynamic (physics-based) equations. As mentioned before, these physics-based algorithms accurately describe the dependence of the unmeasured variables on measured ones at steady states. The question is whether this dependence can be adequately simulated by static data-driven models.

Multiple diagnostic algorithms belonging to the mentioned approaches have been widely utilized so far and comparative studies of these algorithms are known. Some particular recommendations on the selection of the best algorithm can be found, for example, in [26,29]. Nevertheless, studies addressing a general comparison of both approaches are still missing in the literature.

1.2.3 Techniques used for fault diagnostics

As mentioned before, a total diagnostic process can be divided according to three principal and interrelated stages: (a) anomaly detection, (b) fault identification (isolation), and (c) prognosis. These stages are preceded by an additional stage of data validation and feature extraction. The stage of fault identification seems to be the most complex and developed [21]. In this stage, the necessary fault classes are constructed and a diagnostic decision is made in the space of deviations of the monitored variables. For constructing the classes, deviation vectors (patterns) corresponding to real faults can be used. So far available real information rarely suffices to form a representative classification, especially in the beginning of an engine life cycle. Hence, physics-based models are still frequently applied to simulate gas turbine fault scenarios.

Many pattern classification techniques are applied to diagnose gas turbines, including Artificial Neural Networks (ANNs) [24,30,46–50], Genetic Algorithms [46,51], Fuzzy Logic [52–55] and Bayesian Approach [24,56,57]. Among the ANNs employed are Multilayer Perceptron (MLP) [23,50], which is most widespread, Probabilistic Neural Network (PNN) [21,47,58], Radial Basis Network (RBN) [46,50], etc. The MLP consists of a predefined set of input-target pairs and a backpropagation algorithm in the training stage that modifies all weight matrices and bias vectors in the hidden and output layers proportionally to the decreasing gradient of the error function. This update results in the network's ability to learn relationships between the inputs and outputs. When a new input is presented, the outputs of the nearby learning input vectors determine the new output. The PNN utilizes the criterion of a fault probability to classify fault patterns; therefore every diagnosis is accompanied by a confidence probability. The RBN is formed by a layer with radial basis function (RBF) neurons and a layer that generates linear combinations of activations of the radial basis layer. The idea of the RBF neurons is to measure how close the input vector and a weight vector are from each other. In the training process, one neuron is iteratively added at a time to the radial basis layer. This new neuron is created by the input vector that obtains the smallest network error. The neuron addition is stopped when a network error decreases below an error goal or when a maximum neuron number has been reached.

Another type of technique used for fault diagnostics in the past years is Support Vector Machines (SVM) [26,47]. SVM are supervised algorithms applied to classification and regression analysis [59,60]. Given training data as pattern vectors and their corresponding labels, the SVM algorithm maps the original input space into a higher-dimensional feature space through a kernel function to separate the data there with a maximum-margin hyperplane. Since perfect separation is not always possible, the method allows classification errors while a regularization parameter penalizes them. At prediction stage, all classifiers emit votes when a pattern vector is presented. The pattern is assigned to the class with the maximum number of votes. The use of SVMs in machine condition monitoring

[61,62], and especially in gas turbine diagnostics is well known. Qi-hua and Jun [63] apply a multi-class SVM algorithm for aero-engine fault diagnostics. Seo et al. [64] and Lee et al. [65] propose hybrid SVM-ANN methods for defect diagnostics of gas turbines. Zhou et al. [66] develop a gas turbine diagnostic method based on SVM.

Different studies demonstrate that ANNs are outperformed by SVMs in many aspects [59,67]. Some of them are: ANNs suffer from multiple local minima while the solution of SVMs is global and unique; ANNs are much more prone to overfitting than SVMs; SVMs have better generalization than ANNs for small number of samples; the geometric interpretation of SVMs is simpler and give sparse solutions; ANNs use empirical risk minimization whilst SVMs use structural risk minimization; and unlike of SVMs, the computational complexity of ANNs directly depends on the input space dimensionality.

Since gas turbines are very complex machines and need to be monitored, exhaustive comparative studies and benchmarking analysis about diagnostic techniques can give solid recommendations on how to construct an effective monitoring system [23,26,68–70]. However, the theoretical accuracy results provided by different studies are still not sufficient to give a clearer idea to designers, diagnosticians and engineers on how accurate the diagnostic decisions will be for a wide range of gas turbine diagnostic conditions and how much these conditions affect the techniques employed. Besides, the necessity of considering engine fault representations closer to reality is essential in order to produce more truthful and reliable diagnostic assessments.

Despite the advances in the research and development in the areas of engine health management (EHM), there are still inconsistency in defining and representing EHM problems [71]. Over the past years, different EHM solutions have been published but they are applied to different platforms, problems with diverse complexity and evaluations of performance with different metrics. This has created difficulties to correctly compare methodologies and to effectively develop algorithms [72]. To solve the above problems, a recent benchmarking platform called ProDiMES (Propulsion Diagnostic Method Evaluation Strategy) has been created by the NASA Glenn Research Center. The software simulates a fleet of commercial aircraft engines using a nonlinear physics-based steady state model. The simulated data present steady state gas path variables collected every flight (snapshots) for healthy and faulty engines under varying flight conditions. A user diagnostic solution treats the collected data producing diagnostic decisions, and then for this solution ProDiMES determines a set of accuracy performance metrics, which are reviewed in depth in [73]. To illustrate the benchmarking methodology, ProDiMES also includes an example diagnostic solution and its metrics. Thus, using the same input data and performance metrics, different users can validate and compare their developments. Some of the comparative studies based on the ProDiMES platform can be found in [26,29,68,70].

1.3 RESEARCH CENTERS/UNIVERSITIES WORKING IN THE FIELD OF GAS TURBINE DIAGNOSTICS

Over the last decades, several research centers and universities worldwide have been involved in the areas of gas turbine condition monitoring and diagnostics. Table 1.3 presents some active institutions as well as their scientific publications. Like other centers, they play an important role in the research and development of advanced gas turbine diagnostic algorithms and systems. In Mexico, the area of monitoring and diagnosis of gas turbines is not well developed; however, research institutes such as Instituto Politecico Nacional (IPN), Centro de Investigación en Ingeniería y Ciencias Aplicadas (CIICAp) and Instituto Nacional de Electricidad y Energías Limpias (INEEL) have made some contributions to gas turbine diagnostics.

Table 1.3 Research centers and universities working in the field of gas turbine diagnostics.

Research center/University	**Country**
NASA Glenn Research Center [29,68,72,74]	USA
Cranfield University [17,18,30,46,51,55,75]	United Kingdom
National Technical University of Athens [24,25,28,38,41,58,70,76]	Greece
Carleton University [3,77–80]	Canada
University of Liège [27,29,76,81–85]	Belgium
National Research Council of Canada [68,73,86–88]	Canada
Concordia University [89–93]	Canada
Qatar University [94–98]	Qatar
National Aerospace University (KhAI) [34,99–102]	Ukraine
Harbin Engineering University [103–106]	China
Nanjing University of Aeronautics and Astronautics [107–113]	China
Instituto Politécnico Nacional [19,21,45,114,115]	Mexico

1.4 RESEARCH OBJECTIVES

Considering the analysis performed above and a general conclusion that effective diagnostic systems need the development and evaluation of different algorithms, the research objectives of the present thesis are:

1. To present the foundations of nonlinear gas turbine performance simulation to gain a better understanding of diagnostic models as well as the software structure of a thermodynamic model.

2. To develop a methodology for computing any necessary gas turbine unmeasured variables and their deviations by “light” data-driven algorithms that are not exigent for computational resources and can be used in online monitoring systems.

3. To develop a procedure to assess different gas turbine diagnostic techniques and to perform the technique comparison under variable fault conditions.

4. To develop and benchmark a data-driven gas turbine diagnostic algorithm through a platform that allows engine data generation, computation of performance metrics and comparison with other diagnostic solutions.

5. To perform a comparative study between physics-based and data driven gas turbine diagnostic approaches in order to know whether they achieve the same level of diagnosis accuracy.

1.5 THESIS OUTLINE

The organization of this thesis is as follows:

Chapter 2 firstly gives an overview of gas turbine diagnostic models. Next, the test case engines used throughout the thesis are briefly presented. The chapter also contains an example of a free turbine engine performance simulation and the software of its thermodynamic model. A qualitative analysis of model accuracy is finally addressed based on the comparison of two test case engines.

Chapter 3 presents the development of an online monitoring algorithm based on unmeasured gas turbine variable estimations. Some principles of GTE diagnostics based on measured and unmeasured variables are firstly introduced. Next, the chapter proposes three different variations of the algorithm for estimating unmeasured quantities under the conditions of simple engine faults. The algorithm with the best performance is finally selected to analyze the estimation of unmeasured quantities for more complex fault scenarios that validate the algorithm.

Chapter 4 performs a comparative study of gas path diagnostic techniques using a principle of variable structure classification applied to cover possible fault scenarios in gas turbine maintenance. A foundation for each technique under analysis is firstly given in the chapter as well as the procedure for diagnostic technique evaluation. Then, different case studies are performed using the diagnostic procedure to compare the techniques and draw solid conclusions about the influence of variable fault conditions on the techniques.

Chapter 5 introduces and validates a data-driven gas turbine diagnostic algorithm using the ProDiMES benchmarking software. In an initial section, the ProDiMES structure that includes a benchmarking process, engine fleet simulation and diagnostic performance metrics is described. Then, the proposed algorithm testing procedure is covered that includes the steps of fleet-average baseline modeling, pattern-based fault classification, validation process, tuning and comparison of the techniques. Finally, the best configuration of the algorithm is compared with other diagnostic solutions.

Chapter 6 presents an algorithm for comparing physics-based and data-driven gas turbine diagnostic approaches. A general description of the methodology is given along with the foundation of each approach. Two test case engines with their comparison conditions and case studies are employed to evaluate the approaches.

All the algorithms developed in the present thesis are implemented in Matlab that offers convenient toolboxes for different areas such as machine learning and pattern recognition assisting in effective algorithm development.

Chapter 7 summarizes the proposed methodologies, their contributions to the areas of research and future works.

Chapter 2: Non-Linear Gas Turbine Performance Simulation

2.1 AN OVERVIEW OF GAS TURBINE DIAGNOSTIC MODELS

2.1.1 Thermodynamic non-linear model

In the past, there have been unsuccessful attempts to implant mechanically damaged components in gas turbines to measure the resulting deterioration. This is clearly an extremely expensive method and it is not possible to implant faults due to the possibility of causing more serious damages. The only feasible method of systematically investigating the deterioration of gas turbines is by mathematical models (also called physics-based models) based on established non-linear thermodynamic relationships to describe engine stationary or dynamic states. These models must be validated with available field data. Since the studies carried out by Saravanamuttoo et al., especially [35,39], the application of the these mathematical models for stationary states has become a common practice by many researchers worldwide [23,24,72,77,116–119]. Reviews on gas turbine modeling techniques can be found, for example, in [86,120]. The most important feature of the models is the ability to implement controlled faults, where the efficiency and flow characteristics of the components can be modified to represent different levels of degradation. The model gives a complete and successive description of all principal gas path components (compressor, combustor, turbine, etc.). The model computes a $(m\times1)$-vector $\vec{Y}$ of gas path monitored variables as a function of a vector $\vec{U}$ of steady state operating conditions (control variables and ambient conditions) as well as a $(r\times1)$-vector $\vec{\theta}$ of health parameters, which shift component operating maps in different directions simulating gradual deterioration mechanisms and faults of varying severity. When a fault in a determined component occurs, all the engine parameters change as well due to a non-linear dependence $Y(\vec{\theta})$ between the measured and the health parameters. Consequently, the thermodynamic model can be presented by the following structured expression:

$$\vec{Y} = f(\vec{U},\vec{\theta}) \tag{2.1}$$

There exist two manners to improve the accuracy of gas path mathematical models: 1) by specifying and deepening the description of the physical processes in the gas turbine, 2) by identifying the model with experimental data. The first way is very complex and slow. The second one, the identification of the non-linear model, means the search for such values of internal parameters of the model (for example, health parameters $\vec{\theta}$) which bring the model parameters closer to the measurement variables $\vec{Y}^*$. Therefore, we can present the operation of the identification as the minimization of an objective function [17,19,121]:

$$\hat{\vec{\theta}} = \arg\min\left\|\vec{Y}^* - \vec{Y}(\vec{\theta},\vec{U})\right\| \tag{2.2}$$

where $\| \|$ is the norm of the vector. The procedure of identifying the nonlinear model forms the foundation of the most perfect and complex diagnostic algorithms. Paper [40] describes one of the first thermodynamic model identification procedures. Reference [41] provides the results of the procedure implementation in an integrated condition monitoring and diagnostic system. The minimization of the objective function presented in Eq. (2.2) is performed iteratively until obtaining the best estimated health parameter $\hat{\vec{\theta}}$ for real $\vec{\theta}$. For example, the minimization of the objective function using genetic algorithms (GA) is addressed in [17]. The GA have the advantage that any function (even non-smooth functions) can be optimized, a global search is employed to avoid local minimums, and probabilistic instead of deterministic transition rules are used to create the next generation of strings from the current one [122].

2.1.2 Linear model based methods

The relationship between the dependent parameters in the gas path (pressures and temperatures, thrust, mass flow, etc.) and the independent parameters (pressure ratio, flow capacity and efficiencies) is non-linear [17]. However, the engine model presented in Eq. (2.1) can be simplified by linearizing the non-linear dependence $\vec{Y}(\vec{\Theta})$ between measured and fault parameters at a certain operating point (such as maximum power or cruise) producing a static linear model given by:

$$\delta\vec{Y} = \mathbf{H} \cdot \delta\vec{\theta} \qquad (2.3)$$

It relates a vector $\delta\vec{\theta}$ of small relative changes of the fault parameters to a vector $\delta\vec{Y}$ of the corresponding relative deviations of the monitored variables by an influence coefficient matrix $\mathbf{H}$. Since linearization errors are not too great, the linear model can be successfully applied for fault simulation at any fixed operating point. The matrix $\mathbf{H}$ reflects the influence of each fault parameter on the measured variables and can be computed by means of the thermodynamic model. The gas path variables $\vec{Y}$ are firstly computed through the model for nominal (healthy state) fault parameters $\vec{\theta}_0$. Then, small variations are introduced by turns in fault parameters and the calculation of the variables $\vec{Y}$ is repeated for each corrected parameter. Finally, for each pair Y_i and θ_j the corresponding influence coefficient is obtained using the following expression:

$$H_{ij} = \frac{\delta Y_i}{\delta\theta_j} = \frac{Y_i(\vec{\theta}_j) - Y_i(\vec{\theta}_0)}{Y_i(\vec{\theta}_0)} \bigg/ \frac{\theta_j - \theta_{0j}}{\theta_{0j}} \qquad (2.4)$$

With the assumption of Eq. (2.3), the first works using GPA were introduced by Urban in the late 70's [42,123] applied to gas turbine condition monitoring and diagnostics of single and simultaneous multiple engine faults. Since then, different studies have applied this GPA method. Studies carried out by Volponi et al. [23] and Ganguli [52,53,118] use a linear model for diagnosing a twin-spool turbofan engine. Studies [124,125] utilize a linear model of a two-spool turboshaft engine to create a variable and complex fault classification for diagnostics. Yu et al. [126] compare different linear models for gas turbine performance. Kamboukos and Mathioudakis [127] make a comparison of linear and non-linear for the purpose of condition monitoring and diagnostics.

When health parameter deviations $\delta\hat{\vec{\theta}}$ are required, they can be estimated with Eq. (2.5) as direct solution of the linear system in Eq. (2.3) if the number of measurements equals the number of health parameters, or with Eq. (2.6) as the solution for the least-squares method (LSM) if the number of measurements is greater than the number of health parameters [128].

$$\delta\vec{\hat{\theta}} = \mathbf{H}^{-1}\delta\vec{Y} \tag{2.5}$$

$$\delta\vec{\hat{\theta}} = \left(\mathbf{H}^{\mathrm{T}}\mathbf{H}\right)^{-1} \cdot \mathbf{H}^{\mathrm{T}} \cdot \delta\vec{Y} \tag{2.6}$$

2.1.3 Non-linear dynamic model

It is common that the use of models for diagnostics at steady states is more frequent than those for transients. However, current studies demonstrate that there is a growing interest in GT diagnostics during dynamic operation [49,129]. A thermodynamic gas path model (dynamic model) is therefore in increasing demand. As distinct from the static model of Eq. (2.1), in the dynamic model a time variable t is added to the argument set of the function $\vec{Y}$ and the vector $\vec{U}$ is given as a time function. Thus, a dynamic model has the following structure [19]:

$$\vec{Y} = f\left(\vec{U}(t),\vec{\theta},t\right) \tag{2.7}$$

A separate influence of time variable t is explained by inertia nature of GT dynamic processes, in particular, by inertia moments of gas turbine rotors. The gas path parameters $\vec{Y}$ of the model (2.4) are computed numerically as a solution of the system of differential equations in which the right parts are calculated from a system of algebraic equations reflecting the conditions of the components combined work at transients. These algebraic equations differ a little from the static model equations. For that reason, the numeric procedure of the algebraic equation system solution is conserved in the dynamic model. Therefore, the nonlinear dynamic model includes the most of static model subprograms. Thus, the nonlinear static and dynamic models tend to be united in a common complex program.

2.2 TEST CASE ENGINES

To develop the proposed gas turbine diagnostic algorithms, three test case engines are employed. Each of them is presented by a non-linear thermodynamic model with the same structure of Eq. (2.1) and they are briefly presented below. Engine variables such as operating conditions, monitored variables, and fault parameters are specified in more detail for each diagnostic algorithm in the subsequent chapters.

2.2.1 Test case engine 1

The first test case engine (GT1) is a turbo-shaft stationary power plant for natural gas-pumping. Its model was validated against the manufacturer data and identified with real engine data [100,121]. Figure 2.1 displays a general scheme of this engine. For diagnostic purposes, the main components under study correspond to inlet device (ID), compressor (C), combustion chamber (CC), high pressure turbine (HPT), and power turbine (PT).

2.2.2 Test case engine 2

The second test case engine (GT2) is a two-spool turboshaft engine for propulsion and power generation applications. Figure 2.2 shows the generalized engine scheme. Its model belongs to a complete gas turbine performance simulation software called GasTurb created by the company GasTurb GmbH [130]. GasTurb allows simulating the performance of a wide variety of gas turbines applied to aviation, power generation or propulsion. Studies using this software can be found, for example in [131,132].

2.2.3 Test case engine 3

The third engine under study (GT3) is a high-bypass turbofan engine presented by a steady state thermodynamic model called C-MAPSS (commercial modular aero-propulsion system simulation) created for diagnostics research [72,117]. C-MAPSS is part of the ProDiMES software and works with two spool speeds, fan and core speed. Figure 2.3 shows the station numbers, the modules and the simulated sensor variables of the engine that are described in more detail in Chapter 5.

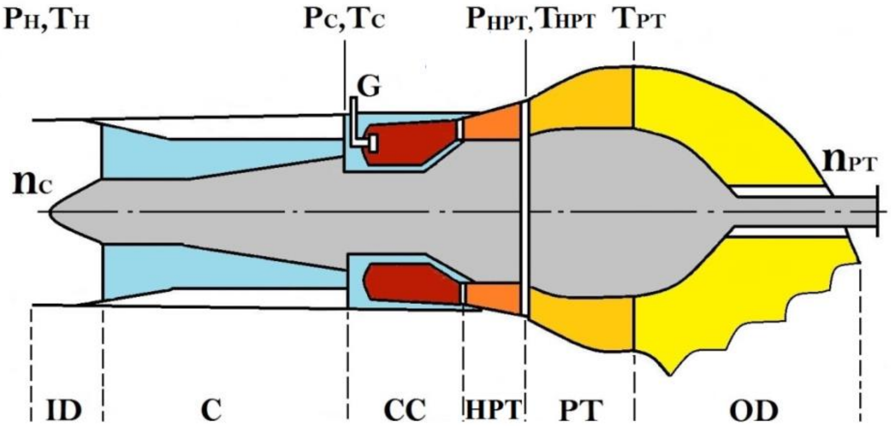


Figure 2.1 General scheme of GT1.

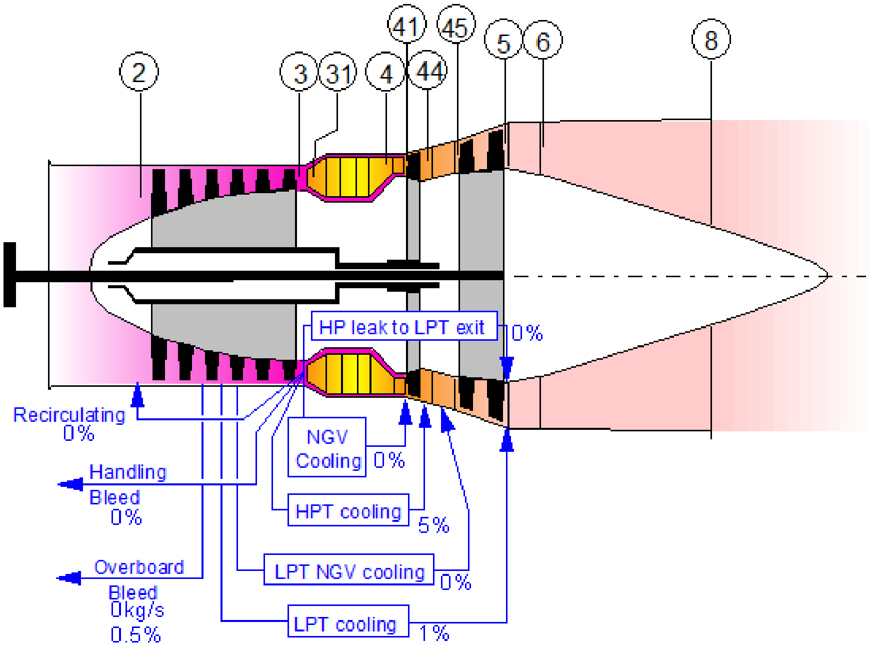


Figure 2.2 General scheme of GT2.

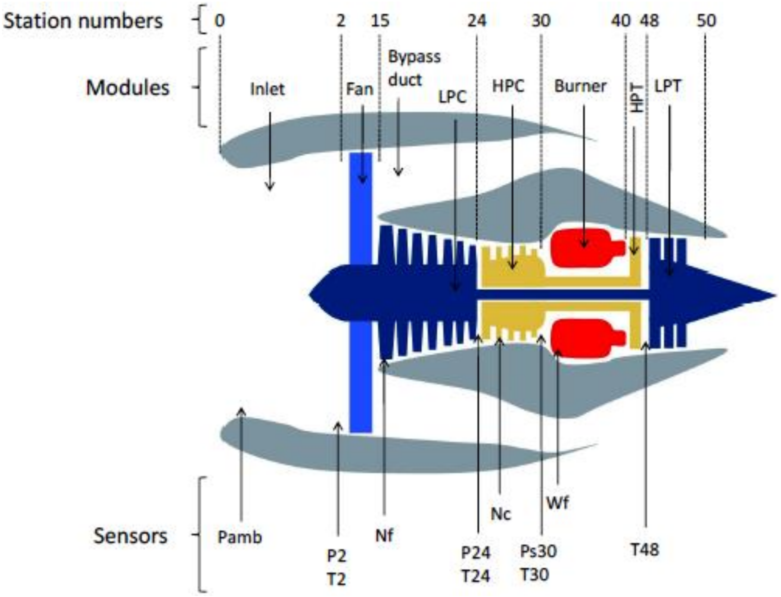


Figure 2.3 General scheme of GT3.

2.3 STRUCTURE OF A THEMODYNAMIC MODEL

2.3.1 Performance simulation of a free turbine engine

Any gas turbine is a collection of discrete components, each of which must be described in thermodynamic terms. Thus, a complete thermodynamic gas turbine model can be constructed by linking all component models [133]. Let us briefly present an example of gas turbine performance simulation for a free turbine engine. The below description was extracted from Saravanamuttoo and MacIsaac [39] (Eqs. 2.8-2.12) and from Kurz and Brun [7] (Eqs. 2.13-2.18). Figure 2.4 displays the main sections of the engine, which has the same configuration as GT1.

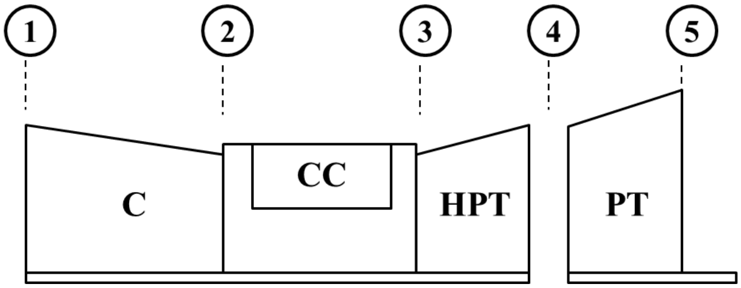


Figure 2.4 Free turbine engine (adapted from [39]).

Performance prediction methods depend on the application of physical laws and performance maps of each component. To achieve the steady-state engine operation, it is necessary to maintain the flow balance between each main component and a power balance between the compressor and its high pressure turbine, and also between the free power turbine and its load. Since the flow through each component is generally described by a set of empirically derived maps that, in turn, are expressed in a dimensionless way, it is convenient to define the equations that describe the steady state behavior of the gas turbine in a dimensionless manner. For example, if fuel and bleed flows are not available a flow balance between the compressor and the gas generator turbine would require that:

$$W_3 = W_1 \tag{2.8}$$

where W_1 and W_3 are the compressor and high pressure turbine mass flows, respectively. In a dimensionless form, Eq. (2.8) is transformed to:

$$\frac{W_3\sqrt{T_3}}{P_3} = \frac{W_1\sqrt{T_1}}{P_1} \times \frac{P_1}{P_2} \times \frac{P_2}{P_3} \times \sqrt{\frac{T_3}{T_1}} \tag{2.9}$$

where:

$\frac{W_3\sqrt{T_3}}{P_3}$ is the dimensionless high pressure turbine flow

$\frac{W_1\sqrt{T_1}}{P_1}$ is the dimensionless compressor flow

$\frac{P_2}{P_1}$ is the compressor pressure ratio

$\frac{P_3}{P_2}$ is the combustor pressure ratio

$\frac{T_3}{T_1}$ is the dimensionless turbine temperature ratio

For work compatibility between the compressor and its driving turbine:

$$W_1 C_{pa} \Delta T_{12} = W_3 C_{pg} \Delta T_{34} \eta_m \tag{2.10}$$

where η_m is the mechanical efficiency; C_{pa} and C_{pg} are specific heat capacities for air and exhaust gas, respectively. Hence,

$$\Delta T_{12} = \Delta T_{34} \frac{W_3 C_{pg} \eta_m}{W_1 C_{pa}} \tag{2.11}$$

In a dimensionless form, since $W_3 = W_1$:

$$\frac{\Delta T_{12}}{T_1} = \frac{\Delta T_{34}}{T_3} \frac{T_3}{T_1} \frac{C_{pg}\eta_m}{C_{pa}} \tag{2.12}$$

In this way, Eq. (2.9) and Eq. (2.12) for $\frac{T_3}{T_1}$ are obtained and the engine operating point that satisfies both must be found by trial and error, using data obtained from the non-linear component characteristics. The procedure by which this is performed is commonly called component matching.

For the free power turbine of Figure 2.1, the flow compatibility with the power turbine and the gas generator must also be established and it the following structure:

$$\frac{W_4\sqrt{T_4}}{P_4}=\frac{W_3\sqrt{T_3}}{P_3}\times\frac{P_3}{P_4}\sqrt{\frac{T_4}{T_3}} \tag{2.13}$$

Another condition is imposed by the flow capacity of the gas generator and power turbine. From a pressure balance, we get the identity

$$\frac{P_3}{P_4}=\frac{P_3}{P_2}\frac{P_2}{P_1}\frac{P_1}{P_4} \tag{2.14}$$

where P_3/P_2 is determined by the pressure drop in the combustor. This means that for any compressor pressure ratio, the gas generator pressure ratio is fixed by the power turbine pressure ratio. In particular, if the power turbine is in choke, then the gas generator pressure ratio is a unique fixed value. Furthermore, the fuel flow is determined from:

$$W_f=\frac{C_{pg}(T_3-T_2)W_1}{LHV} \tag{2.15}$$

As mentioned before, any component of the engine can be represented by a performance map. In order to allow for small deviations from the design point, the dependency of operating parameters can be linearized, i.e. we do not necessarily need to model the entire performance map (Table 2.1). Even with these relatively simple relationships, the typical behavior of a two shaft engine can be simulated: the limitation by maximum speed at inlet temperatures below the match temperature, while being limited by maximum TIT at temperatures above the match temperature. The optimum speed of the power turbine will depend on the gas generator operating point. With

$$\left(\frac{u}{c_{ax}}\right)_{opt}\propto\left(\frac{N_{pt}}{Q_4}\right)_{opt} \tag{2.16}$$

we derive the optimum power turbine speed from:

$$N_{pt,opt}=\left(\frac{N_{pt}}{Q_4}\right)_{opt}Q_4=\left(\frac{N_{pt}}{Q_4}\right)_{opt}W\frac{RT_4}{P_4} \tag{2.17}$$

where N_{pt} is the power turbine speed and, Q is the volumetric flow, c_{ax} is the axial velocity and u is the tip speed. The reduction in power turbine efficiency at off-optimum speeds can be approximated by

$$P=P_{opt}\left[2.0\frac{N_{pt}}{N_{pt,opt}}-\left(\frac{N_{pt}}{N_{pt,opt}}\right)^2\right] \tag{2.18}$$

These conditions are sufficient to explain the behavior of gas turbines at off-design conditions, for example, at ambient temperatures different from the ‘‘match’’ temperature: At higher ambient temperatures, the pressure ratio P_2/P_1 is reduced because the same head produces less pressure ratio, and the mass flow is reduced because the choked power turbine nozzle limits the flow (temperature topping). At lower ambient temperatures, gas generator speed limits override a limit in firing temperatures (speed topping).

Increasing the power turbine nozzle flow area reduces the power turbine resistance (i.e., increases the flow capacity), allowing a higher portion of the overall pressure ratio across the gas generator turbine. The TIT needed for the gas generator turbine to drive the compressor is reduced. When an engine matched to give rated TIT at rated gas generator speed at, for example, 15°C (59°F) is operated at 49°C (120°F), the rated TIT occurs before the rated gas generator speed can be obtained. To increase the gas generator speed, the power turbine area would have to be increased.

Table 2.1 Linearized operating parameters [7]

Linearized parameter	**Depending on**	**Equation**
Compressor pressure ratio	Corrected gas generator speed	$k_{prc} = \Delta TT_c / \Delta N_{gpcorr}$
Compressor flow function	Corrected gas generator speed	$K_{Wc} = \Delta W_c / \Delta N_{gpcorr}$
Compressor efficiency	Corrected gas generator speed	$k_{\eta c} = \Delta \eta_c / \Delta N_{gpcorr}$
Gas generator pressure ratio	Compressor pressure ratio	$k_{gpt} = \Delta TT_{gpt} / \Delta TT_c$
Gas generator turbine efficiency	Gas generator turbine speed	$k_{\eta gpt} = \Delta \eta_{gpt} / \Delta\left(Q_3 / N_{gp}\right)$

2.3.2 Software of the model

Mathematically, a thermodynamic model is a system of non-linear algebraic equations that reflect the balance of mass, heat and energy for all components that operate under stationary conditions. The number of algebraic equations in the thermodynamic model can be more than fifteen and it presents complex software that includes dozens of subroutines as listed in Table 2.2 for GT1. Figure 2.5 and Figure 2.6 show the hierarchy level of each program and the principal subroutine "Dr" for computing the throttling, respectively. The rest of the subroutines employed in the software can be found in Appendix 2. Most of these subprograms are independent universal modules of a simulated gas turbine, thus simplifying the creation of a new engine model.

Table 2.2 Main program and subroutines for GT1 thermodynamic model

No.	**Identifier**	**Description**
1	ATMOS.FOR	Standard atmosphere parameters
2	Atmosx.for	Choosing the mode of input parameters definition
3	BCN.FOR	Rotation speed correction
4	BG.FOR	Air flow correction
5	CLS.FOR	Screen cleaning
6	CPFT.FOR	Thermodynamic properties calculation using temperature
7	CRIT.FOR	Critical parameters computation through temperature
8	Dr.for	Computation of the throttling
9	DROSSL.FOR	Throttle computation (supercharger in the circuit)
10	DSIMQ.FOR	Standard program of linear equations system solution by means of exclusion
11	DSUM.FOR	Sum calculation
12	DSUM2.FOR	Calculation of a sum of squares
13	ETAG.FOR	Combustion efficiency
14	Gg1.for	Gas generator computation

Table 2.2 Main program and subroutines for GT1 thermodynamic model (Continuation)

15	Holder.for	Computation holder
16	INI90G.FOR	Component performance initialization
17	Inoutd.for	Input from DATA.DAT, output to DATA.DAT
18	INTCN.FOR	Standard program of the interpolation of the function of two arguments
19	KAMSG.FOR	Combustion chamber computation
20	KOMPR.FOR	Compressor exit parameters computation
21	LAM.FOR	Velocity coefficient computation
22	LINTX.FOR	Interpolation of the function of one argument
23	LINTXY.FOR	Interpolation of the function of two arguments given by a table
24	LINXYT.FOR	Turbine performance interpolation
25	NACHBL.FOR	Initial block common for different calculations
26	NAGNET.FOR	Supercharger calculations
27	Parreg.for	Power set parameter calculation in dynamics
28	PARVTS.FOR	Air starter parameters
29	Pdst1.for	Independent variables and discrepancies for a free turbine engine with one spool gas generator
30	POINT.FOR	Determination a gas path parameter through its number in a common array
31	Prntf.for	Output of the parameters table to a file
32	RDPAR.FOR	Additional parameters calculation
33	RDST1.FOR	Calculation of a free turbine engine with a one spool gas generator
34	RESET.FOR	Screen reset
35	RSUMN.FOR	Standard program to solve a nonlinear system by the Newton method
36	SEAR.FOR	Searching the table interval where the given value is situated
37	SEARCH.FOR	Linear interpolation of a compressor performance
38	SIGKS.FOR	Combustion chamber pressure loss parameter
39	SIGP4.FOR	Pressure loss parameter in the channel between HPT and PT
40	SIGS.FOR	Nozzle pressure loss parameter
41	SIGS1.FOR	Internal gas path nozzle pressure loss parameter
42	SIGST.FOR	Pressure loss parameter in the volume after PT
43	SIGVH.FOR	Input device pressure loss parameter
44	Sithar.for	Program to expose not expected situations in component performance calculations
45	Sitter.for	Program to expose not expected situations in common thermodynamic calculations
46	SOPLO.FOR	Nozzle parameter calculations
47	Stoil.for	Standard hydro-carbonaceous fuel performances
48	Table.for	Forming the table for SITHAR messages
49	TAUVD.FOR	Air heating coefficient for the 2nd air bleeding point of the compressor
50	TAUVD1.FOR	Air heating coefficient for the 1st air bleeding point of the compressor
51	TAUVD2.FOR	Air heating coefficient for the air bypass point in the compressor
52	TEXTIN.FOR	Text line initialization
53	TFI.FOR	Thermodynamic properties calculation using the given enthalpy
54	TFS.FOR	Thermodynamic properties calculation using the given entropy
55	TURBIN.FOR	Turbine discharge parameters calculation
56	WHIBLU.FOR	White symbols, blue background
57	WHIRED.FOR	White symbols, red background
58	XKVD.FOR	Compressor parameters computation
59	XNAGN.FOR	Supercharger parameters computation
60	XTND.FOR	PT parameters computation
61	XTVD.FOR	HPT parameters computation

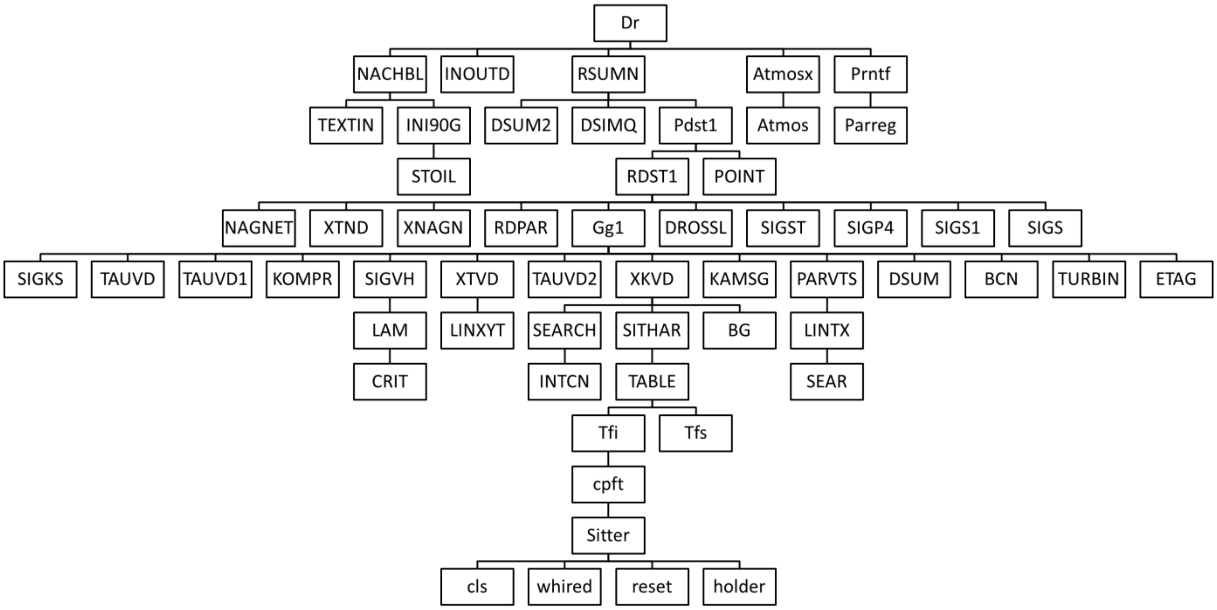


Figure 2.5 Main program and subroutines for GT1 thermodynamic model

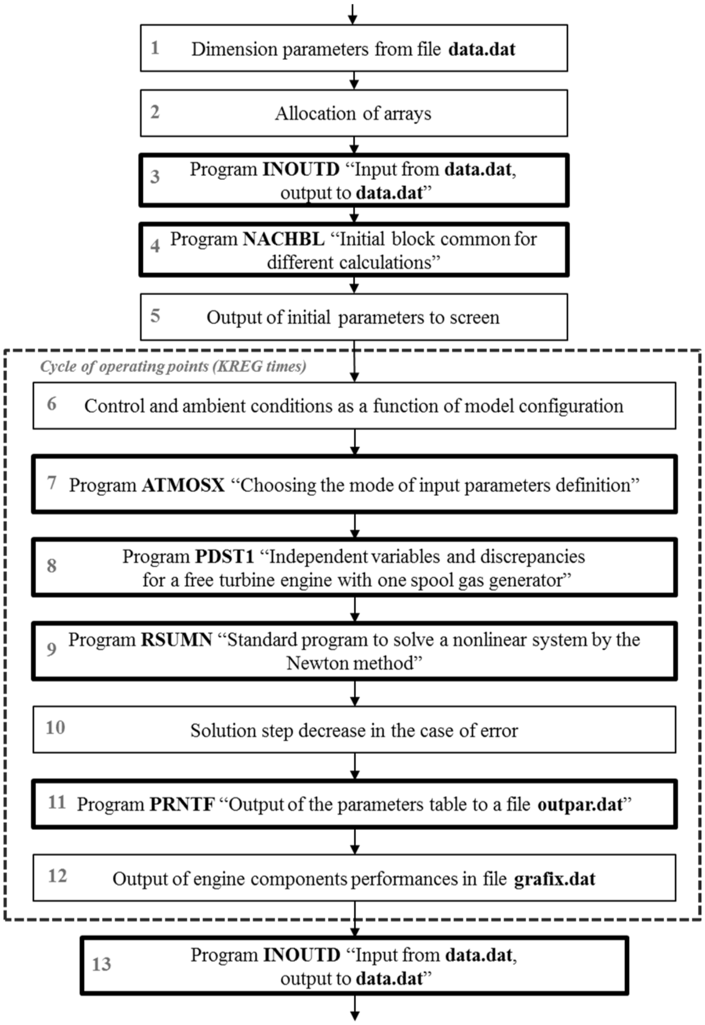

1 Dimension parameters from file data.dat
2 Allocation of arrays
3 Program INOUTD "Input from data.dat, output to data.dat"
4 Program NACHBL "Initial block common for different calculations"
5 Output of initial parameters to screen
Cycle of operating points (KREG times)
6 Control and ambient conditions as a function of model configuration
7 Program ATMOSX "Choosing the mode of input parameters definition"
8 Program PDST1 "Independent variables and discrepancies for a free turbine engine with one spool gas generator"
9 Program RSUMN "Standard program to solve a nonlinear system by the Newton method"
10 Solution step decrease in the case of error
11 Program PRNTF "Output of the parameters table to a file outpar.dat"
12 Output of engine components performances in file grafix.dat
13 Program INOUTD "Input from data.dat, output to data.dat"

Figure 2.6 Main program "Dr" (computation of the throttling).

The software of the non-linear model is developed on the basis of the universal model and the generalized gas turbine scheme with necessary blocks (engine components) and sections as illustrated in Figure 2.7. The model allows executing several types of calculations. For example, the computation of the throttling allows determining the engine variables for the series of stationary regimes under the conditions of fixed environmental variables. Only the regime variable changes from one point to the other in the calculation. The calculation of a stationary regime includes as a basic part the successive computation of the gas path (in other words, the calculation of the components from the beginning: input, compressor, combustion chamber and so on). This calculation has a specialty: the arguments of the characteristics of components (compressors, turbines) are unknown at the time of use. Therefore, we form the following procedure. We include the arguments in the vector $\vec{X}$ of unknown variables. Before the computation, we give the variables initial values and calculate the gas path with the determined vector $\vec{X}$. However, the calculated variables in the gas path are not correct because of the use of inaccurate values of the vector $\vec{X}$. Therefore, we form the system of algebraic equations as follows.

$$\vec{H}\left(\vec{X},\vec{U},\vec{\theta}\right)=0 \tag{2.19}$$

Each equation of this system reflects a joint working condition of the gas turbine components. For example, the following equivalence is the equation corresponding to the power balance of a rotor:

$$H_N=\frac{N_C}{N_T\eta_{mec}-\Delta N_T}-1 \tag{2.20}$$

where N_C and N_T correspond to the power of compressor and turbine. The system is solved by Newton's method that presents a cyclic procedure. At every new iteration i, the current solution $\vec{X}^{i+1}$ is given by the sum of the previous solution and the current correction:

$$\vec{X}^{i+1}=\vec{X}^{i}+\Delta\vec{X}^{i} \tag{2.21}$$

The correction $\Delta\vec{X}^{i}$ is determined as the solution of the following linear system:

$$\vec{N}^{i}=B^{i}\Delta\vec{X}^{i} \tag{2.22}$$

where the elements of matrix B (influence of $\vec{X}$ on H) are computed according to the formula:

$$\left(B_{ql}\right)^{i}=\frac{\delta H_q^i}{\delta X_l^i} \tag{2.23}$$

In the model, each component characteristic $v=f\left(\pi_1,\pi_2\right)$ is written in the following parametric form:

$$v=v_0\left(1+a\frac{\pi_1-\pi_{10}}{\Delta\pi_1}+b\frac{\pi_2-\pi_{20}}{\Delta\pi_2}+c\right) \tag{2.24}$$

where v_0 is the initial value of the characteristic; π_1,π_2 and π_{10},π_{20} are current and basic quantities of the arguments; $\Delta\pi_1,\Delta\pi_2$ are the probable changes of the arguments π_1 and π_2; a ,b

, c are constant coefficients, the parameters of the characteristics of components (form the vector $\vec{\theta}$ in the model). Such presentation of the characteristics allows automating the process of computation of the influence matrix H .

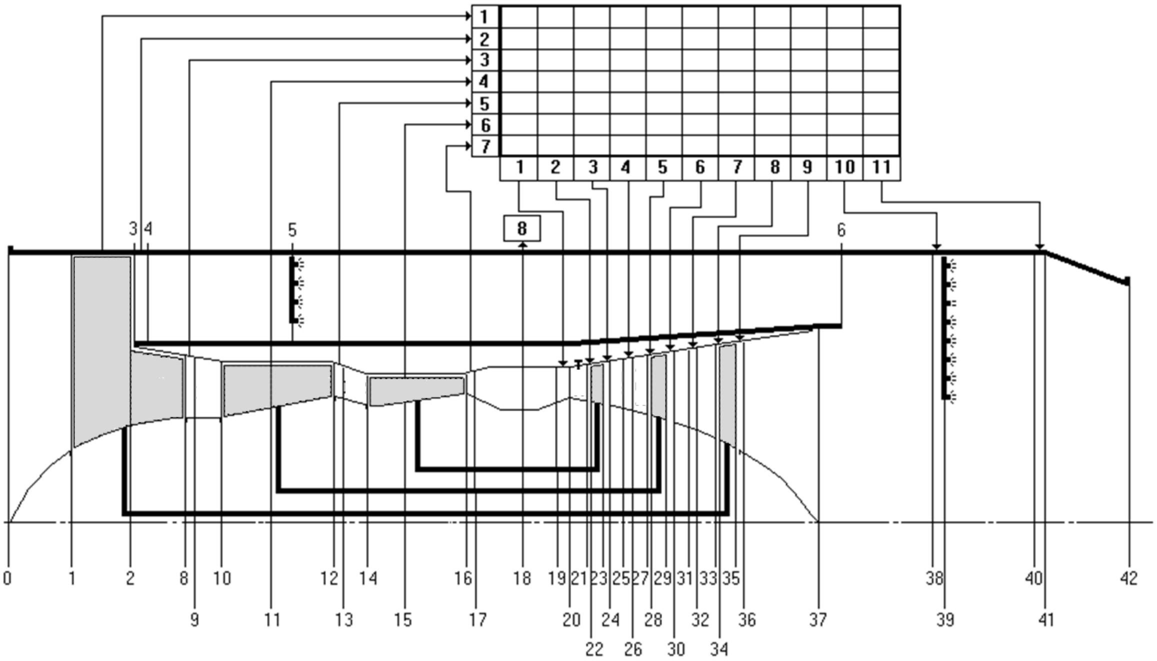


Figure 2.7 Generalized gas turbine scheme with components and stations.

2.4 QUALITATIVE ANALYSIS OF MODEL ACCURACY

In order to have reliable monitoring and diagnostic systems based on GPA, it is necessary that non-linear models describe the performance of gas turbines with a good level of accuracy. A simplified and easy form to verify the thermodynamic model accuracy is through a qualitative analysis of the curves of influence coefficients H against operating conditions U. When the non-linear thermodynamic relationships are well described in a model, the behavior of these curves is smooth and without perturbations.

To exemplify an accuracy model analysis, let us consider two engine models with the same structure, GT1 and GT2 to obtain plots of H versus U. Since coefficients H can present high variations (produced by the inexactitude of linear models) in certain fault severity intervals, an analysis of H versus $\delta\theta$ needs to be firstly performed to find an interval where the values of influence coefficients remain as invariable as possible. For that purpose, healthy and faulty engine data for various fault parameters and fault severity intervals are simulated. Table 2.3 lists six fault parameters with three intervals for each of them simulated by the GasTurb software (GT2). Figure 2.8 displays the compressor map where each square point represents a shift in engine performance due to $\delta\eta_C$ for a fixed relative gas generator spool speed. With the simulated data for different gas path monitored variables, deviations $\delta\vec{Y}$ and influence matrices are computed using Eq. (2.4). As can be seen in Figure 2.9 for δG_{PT}, intervals 1 and 3 present considerable variations of influence coefficients (Δ between points A and B $\neq 0$) for some monitored variables while Interval 2 shows almost invariable influence coefficients (Δ between points A and B $\cong 0$).

Table 2.3 Fault parameters and severity intervals to analyze the behavior of influence coefficients for GT2.

Fault parameter	Interval	Severity change
-Delta compressor capacity δG_C -Delta compressor efficiency $\delta\eta_C$ -Delta HPT capacity δG_{HPT} -Delta HPT efficiency $\delta\eta_{HPT}$ -Delta PT capacity δG_{PT} -Delta PT efficiency $\delta\eta_{PT}$	1	Start value = 0 (healthy engine) Step size = −0.001% Final value = −0.097%
	2	Start value = −0.01% Step size = −0.01% Final value = −0.98%
	3	Start value = −1% Step size = −0.1% Final value = −10%

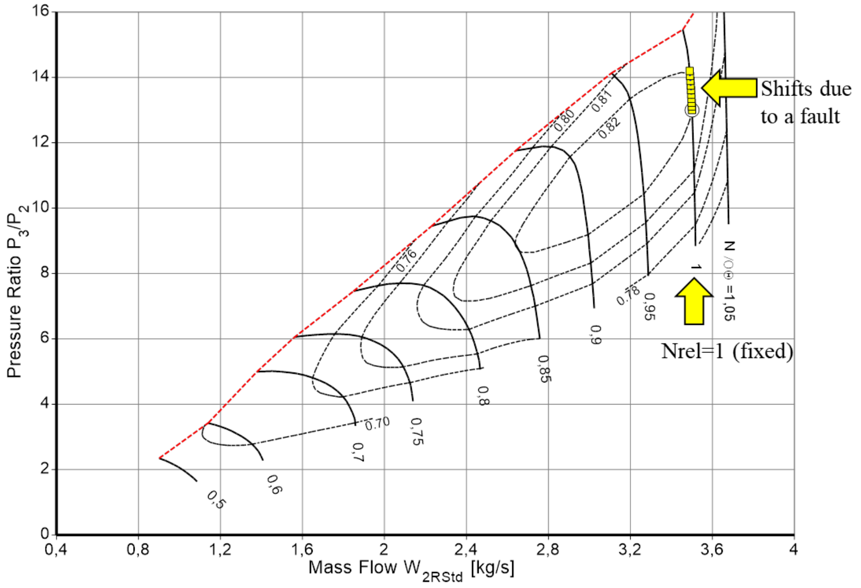


Figure 2.8 Engine performance shifts in the compressor map due to $\delta\eta_C$ (GT2).

A similar analysis of H vs $\delta\theta$ for GT1 is also performed. After analyzing different intervals for both engines and considering the zones where there is not a considerable variation of coefficients H , a value of −1% of severity for all fault parameters is selected. Using this fixed fault severity value, the behavior of influence coefficients against operating conditions is plotted. Figure 2.10 shows influence coefficients ($\delta\eta_C$ =−1%) computed for different operating points set by the increase of gas generator spool speed and for both engines. It is visible that there is little impact of operating condition fluctuation on coefficients influence coefficients. However, the curves of GT2 are smoother than GT1. This indicates that the model from GasTurb captures the engine non-linear thermodynamic relationships with a better level of accuracy than GT1. This can be explained by the fact that the model of GT1 is based on traditional modeling. In a traditional gas turbine performance modeling process, the common method is to discretize component characteristic maps, stored them into the model in the form of arrays, and use a interpolation method to gain component off-design condition performance in the thermodynamic computation [134]. In this way, discretization and interpolation are key procedures of this traditional method. Due to the sparsity of gas turbine component characteristic maps, and discretization which will further reduce characteristic data, component characteristic tables introduced into the thermodynamic model may lose too much information to represent the shape of characteristic map exactly.

Therefore, the above simplified analysis gives place to a possible GT1 model enhancement by considering more optimal methods to describe the thermodynamic relationships without losing information in the modeling process.

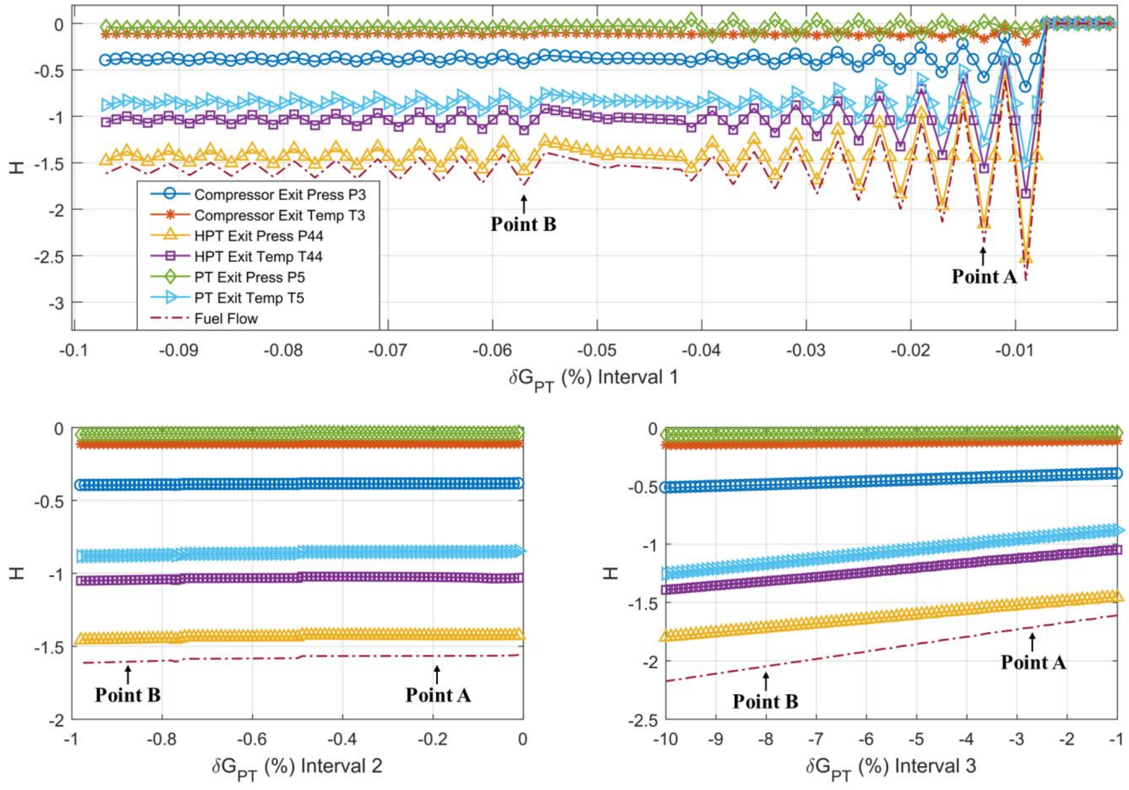


Figure 2.9 Variations of influence coefficients in three intervals of δG_{PT} (GT2).

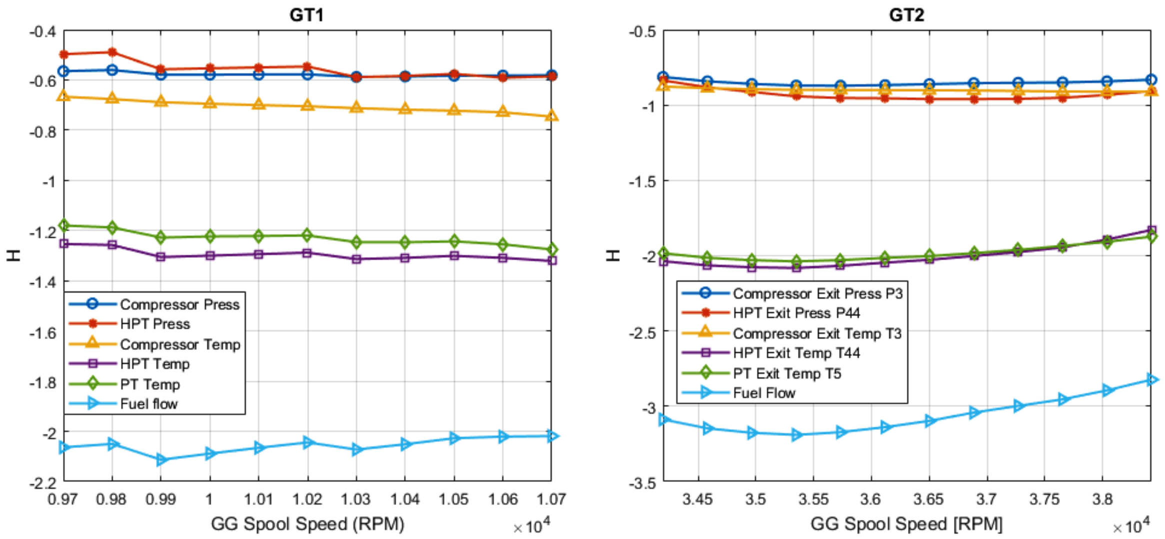


Figure 2.10 Influence coefficients versus GG spool speed ($\delta \eta_C = -1\%$) for GT1 and GT2.

Chapter 3: Development of an Online Monitoring Algorithm for Estimating Unmeasured Gas Turbine Variables

3.1 OVERVIEW

In this chapter we propose to compute any necessary unmeasured variables and their deviations by "light" data-driven algorithms that are not exigent for computational resources and can be used in on-line monitoring systems. The algorithms are examined on the data of GT1 for natural gas pumping units. Although the data are generated by a thermodynamic model, this model is not directly used in the algorithms. Instead, polynomial models that will operate on real data only are built and used. Within the implemented methodology, a baseline model for the unmeasured variables is firstly determined using the thermodynamic model data. Then, with the help of the thermodynamic model, faulty engine data are simulated and a model for the unmeasured variables is determined. Finally, the deviations between the values affected by faults and the baseline values are calculated and analyzed. In addition to this algorithm, the chapter analyzes two others that allow us to choose the best way to estimate unmeasured variables for more complex fault scenarios.

Figure 3.1 displays in highlighted blocks all the influence factors that determine the main characteristics of the proposed diagnostic algorithm and its development in the chapter. The selection includes: a turboshaft engine for industrial application; gas path diagnostic systems; one-point diagnostic analysis; the stage of data validation and feature extraction; the use of physics-based, data-driven and non-linear models; a fault classification based on single and multiple faults; and the LSM as a method of system identification.

3.2 DIAGNOSTICS BASED ON MEASURED VARIABLES

To draw useful diagnostic information from raw recorded data, a total gas turbine diagnostic process usually includes a preliminary feature extraction procedure, in which deviations are computed [68,135]. A deviation is defined for a monitored variable Y as a relative discrepancy between a gas path measured value Y^* and a baseline value Y_0. These discrepancies are analyzed instead of monitored variables themselves in order to avoid the influence engine operating conditions.

As the baseline depends on the operating conditions and a vector $\vec{\theta}_0$ corresponds to a healthy engine, a baseline model can be presented by:

$$\vec{Y}_0 = f\left(\vec{U}, \vec{\theta}_0\right) = f\left(\vec{U}\right) \tag{3.1}$$

Therefore, deviations for variables $Y_{i,i=1,m}$ are computed by the following equation:

$$\delta Y_i = \frac{Y_i^* - Y_{0i}\left(\vec{U}\right)}{Y_{0i}\left(\vec{U}\right)} \tag{3.2}$$

Deviations consist of a systematic component induced by engine degradation or faults and a noise component that results from sensor errors and a baseline model inadequacy, which in turn depends on the mathematical techniques employed to create it. Loboda et al. [135] shows that complete second order polynomials are sufficiently accurate for baseline models. These polynomials adequately describe engine behavior, and the maximum error relatively to the thermodynamic model does not exceed 0.3% for all possible operating conditions

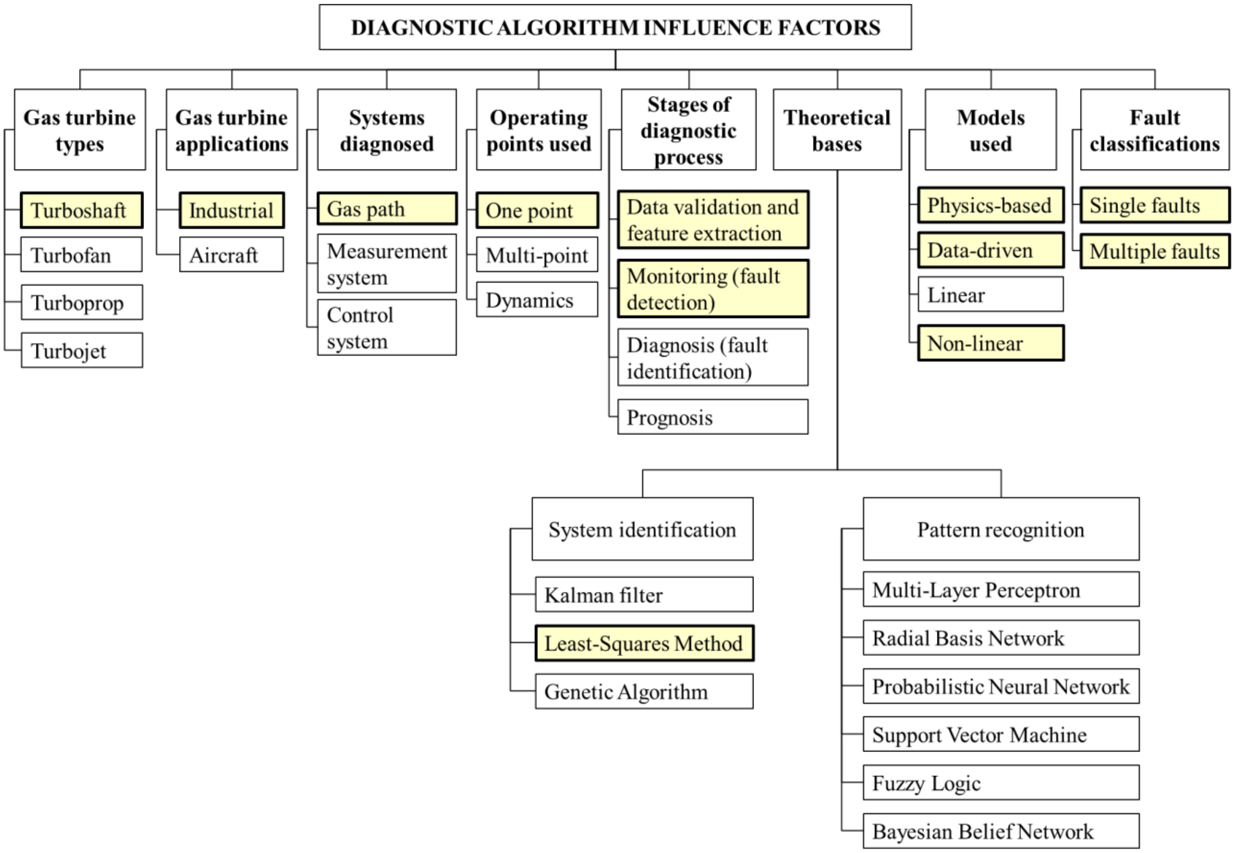


Figure 3.1 Main characteristics of the proposed diagnostic algorithm in Chapter 3.

3.3 DIAGNOSTICS BASED ON UNMEASURED VARIABLES

As mentioned before, one of the monitoring system functions is to compute and monitor important unmeasured engine variables Z, for example, temperature after combustion chamber, compressor and turbine efficiencies, and engine power output. In addition to a thermodynamic model, a Kalman Filter approach can be applied to compute unmeasured variables [23]. Another option is to apply thermodynamic relations between them and measured quantities [31,99,101]. In this case, extensive investigations are needed to develop the best model for each unmeasured variable. We propose to compute all necessary unmeasured variables and their deviations through a universal data-driven model. As in the case of measured quantities, baseline values of unmeasured variables only depend on operating conditions $\vec{U}$. Consequently, a baseline function can be presented by:

$$\vec{Z}_0 = f\left(\vec{U},\vec{\theta}_0\right) = f\left(\vec{U}\right) \tag{3.3}$$

Under faulty conditions, unmeasured variables also depend on fault parameters $\vec{\theta}$. Since the latter are not measurable, the use of measured quantities $Y\left(\vec{\theta},\vec{U}\right)$ is proposed instead. Thus, the function to compute unmeasured variables is given by:

$$\vec{Z} = f\left(\vec{Y},\vec{U}\right) \tag{3.4}$$

Since these variables are determined through measurements $\vec{Y}$ and $\vec{U}$, they can be called indirectly measured variables. Based on our previous experience, full second order polynomials are chosen to build the necessary models $\vec{Z}_0\left(\vec{U}\right)$ and $\vec{Z}\left(\vec{Y},\vec{U}\right)$. Unknown polynomial coefficients are determined by the Least-Squares Method (LSM). Similar to the variables $\vec{Y}$, the variables $\vec{Z}$ are strongly influenced by operating conditions and cannot be used as diagnostic features. To draw diagnostic information from indirectly measured variables, deviations are computed in the same way as with the measured variables resulting in:

$$\delta Z = \frac{Z\left(\vec{Y},\vec{U}\right) - Z_0\left(\vec{U}\right)}{Z_0\left(\vec{U}\right)} \tag{3.5}$$

Because variables $\vec{Z}$ determine principal engine performances and the integrity of engine components, their deviations δZ will be valuable diagnostic features.

3.4 SELECTED ENGINE VARIABLES

To evaluate the proposed methodologies, GT1 is employed. The main components under analysis are the compressor (C), high pressure turbine (HPT), combustion chamber (CC), and power turbine (PT).

Table 3.1 presents the measured variables used in the study, the operating conditions (ambient and control variables) used as baseline model input parameters as well as monitored gas path variables typically employed in engine diagnostics. The unmeasured variables chosen are listed in Table *3.2*. All pressures and temperatures given in these tables correspond to a discharge section of the corresponding components.

The fault parameters used to describe faulty scenarios in each component correspond to the performances of air or gas consumption (G) and isentropic efficiency (η). Thirteen samples with healthy and different faulty conditions are generated based on these parameters as shown in Table 3.3. The first sample (Sample 0) represents a healthy engine or no-fault scenario. Samples 1-6 correspond to single faults of a specific severity. Samples 7-9 consider multiple faults based on the shift of two parameters in the same engine component. Samples 10-12 take into account the changes of four parameters in two engine components. Each sample contains the variables $\vec{U}$, $\vec{Y}$ and $\vec{Z}$ at the same 270 operating points.

To determine the influence of measurement errors on the results, three different noise schemes were studied and applied by turn to all samples under analysis:

- “No noise”: original samples of simulated data with no noise added;
- “Filtered noise”: the noise filtered by averaging 20 successive measurements is added;
- “Full noise”: measurement noise without filtering is added.

An individual level of measurement noise is chosen for each variable according to the recommendations made in previous studies [45]. Using the data described above, unmeasured variables and their deviations are determined as addressed in the following subsections.

Table 3.1 Measured variables of GT1.

No.	Description	Unit	Symbol
	Operating conditions $\vec{U}$		
1	Ambient pressure	Pa	P_H
2	Inlet temperature	K	T_H
3	Power turbine rotation speed	rpm	n_{PT}
4	Fuel gas mass flow	kg/h	G
	Measured variables $\vec{Y}$		
1	Compressor pressure	Pa	P_C
2	Gas generator rotation speed	1/min	n_C
3	High pressure turbine temperature (Exhaust Gas Temperature-EGT)	K	T_{HPT}
4	Compressor temperature	K	T_C
5	High pressure turbine pressure (Exhaust Gas Pressure-EGP)	Pa	P_{HPT}
6	Power turbine temperature	K	T_{PT}

Table 3.2 Unmeasured variables (vector $\vec{Z}$) of GT1

No.	Description	Unit	Symbol
1	Combustion chamber discharge pressure	Pa	P_{CC}
2	Combustion chamber discharge temperature	K	T_{CC}
3	Compressor efficiency	-	η_C
4	High pressure turbine efficiency	-	η_{HPT}
5	Power turbine efficiency	-	η_{PT}
6	High pressure turbine power output (GTE shaft power)	W	W_{HPT}

3.5 ALGORITHMS FOR ESTIMATING UNMEASURED QUANTITIES

To ensure the most reliable estimation of unmeasured quantities, three algorithms are proposed. To make the analysis simpler, these algorithms are determined and tested on the data of the samples generated for healthy and single fault conditions (Samples 0-6). For each algorithm, the three noise schemes described above are considered. Algorithm 1 is based on the description given in Subsections 3.1 and 3.2. In Algorithm 2, the baseline models $\hat{\vec{Y}}_0 = f(\vec{U})$ and $\hat{\vec{Z}}_0 = f(\vec{U})$ are firstly determined, and deviations δY and δZ are computed for all faults. On these data, function $\delta\hat{\vec{Z}} = f(\delta\vec{Y})$ is then determined. Finally, unmeasured variables $\hat{Z}$ is recovered using known deviation $\delta\hat{Z}$. Algorithm 3 follows the same steps as Algorithm 2 with the difference that only the compressor faults (Samples 1 and 2) are considered when we determine the function $\delta\vec{\hat{Z}} = f(\delta\vec{Y})$. To evaluate the effectiveness of the procedures, different estimation errors ε are computed as differences between estimated quantities and true values that can be computed or directly obtained from the original samples. The schemes of the algorithms are presented in Figure 3.2 and the following subsections describe the algorithms in more detail.

Table 3.3 Simulated healthy and faulty conditions of GT1.

Sample	Fault parameters and severity	Description
0	None	No-fault
Single faults		
1	$G_c = -0.03$	Compressor air consumption decrease
2	$\eta_c = -0.03$	Compressor efficiency decrease
3	$G_{hpt} = -0.03$	High pressure turbine gas consumption decrease
4	$\eta_{hpt} = -0.03$	High pressure turbine efficiency decrease
5	$G_{pt} = -0.03$	Power turbine gas consumption decrease
6	$\eta_{pt} = -0.03$	Power turbine efficiency decrease
Multiple faults (two parameters)		
7	$G_c = -0.025;\ \eta_c = -0.015$	Compressor fault
8	$G_{hpt} = -0.025;\ \eta_{hpt} = -0.015$	High pressure turbine fault
9	$G_{pt} = -0.025;\ \eta_{pt} = -0.015$	Power turbine fault
Multiple faults (four parameters)		
10	$G_c = -0.015;\ \eta_c = -0.01$ $G_{hpt} = -0.015;\ \eta_{hpt} = -0.01$	Compressor and high pressure turbine fault
11	$G_{hpt} = -0.015;\ \eta_{hpt} = -0.01$ $G_{pt} = -0.015;\ \eta_{pt} = -0.01$	High pressure turbine and power turbine fault
12	$G_c = -0.015;\ \eta_c = -0.01$ $G_{pt} = -0.015;\ \eta_{pt} = -0.01$	Compressor and power turbine fault

3.5.1 Algorithm 1

Baseline model $\hat{Z}_0 = f(\vec{U})$

Despite faulty and non-faulty engine states can be generated through the thermodynamic model of Engine 1, the direct use of such a model in the algorithms produces more complexity and high computational costs. In order to have “light” algorithms and overcome these problems, the thermodynamic model is only used as a source of the data for creating simple surrogate data-driven models for the algorithm.

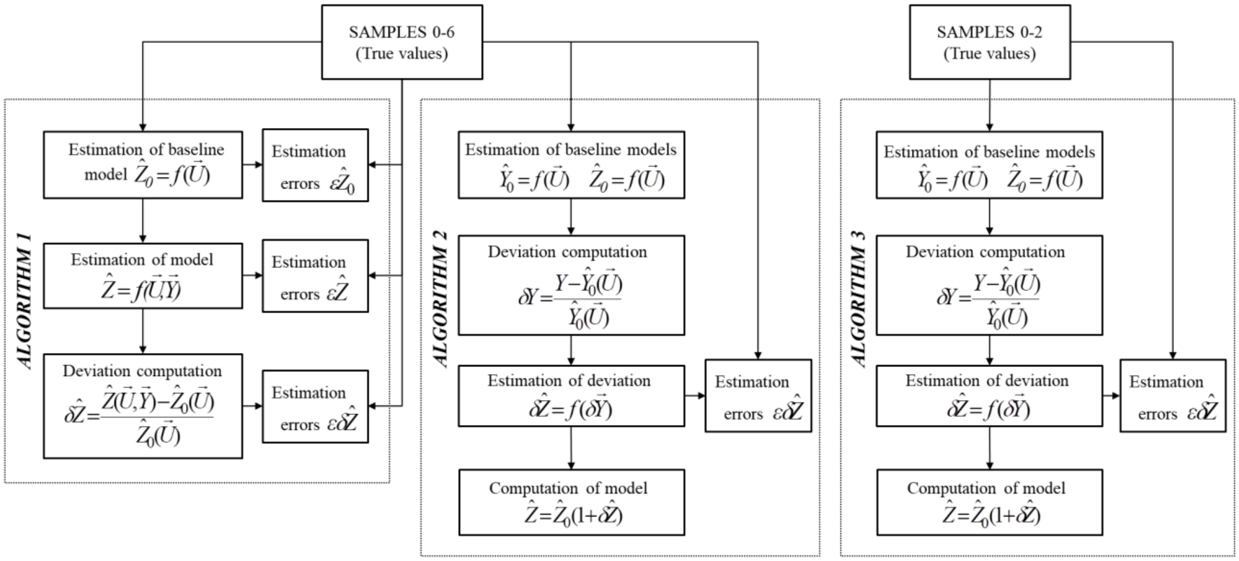

SAMPLES 0-6
(True values)
ALGORITHM 1
Estimation of baseline model $\hat{Z}_0=f(\vec{U})$
Estimation errors $\varepsilon\hat{Z}_0$
Estimation of model $\hat{Z}=f(\vec{U},\vec{Y})$
Estimation errors $\varepsilon\hat{Z}$
Deviation computation $\delta\hat{Z}=\frac{\hat{Z}(\vec{U},\vec{Y})-\hat{Z}_0(\vec{U})}{\hat{Z}_0(\vec{U})}$
Estimation errors $\varepsilon\delta\hat{Z}$
ALGORITHM 2
Estimation of baseline models $\hat{Y}_0=f(\vec{U})$ $\hat{Z}_0=f(\vec{U})$
Deviation computation $\delta Y=\frac{Y-\hat{Y}_0(\vec{U})}{\hat{Y}_0(\vec{U})}$
Estimation of deviation $\delta\hat{Z}=f(\delta\vec{Y})$
Estimation errors $\varepsilon\delta\hat{Z}$
Computation of model $\hat{Z}=\hat{Z}_0(1+\delta\hat{Z})$
SAMPLES 0-2
(True values)
ALGORITHM 3
Estimation of baseline models $\hat{Y}_0=f(\vec{U})$ $\hat{Z}_0=f(\vec{U})$
Deviation computation $\delta Y=\frac{Y-\hat{Y}_0(\vec{U})}{\hat{Y}_0(\vec{U})}$
Estimation of deviation $\delta\hat{Z}=f(\delta\vec{Y})$
Estimation errors $\varepsilon\delta\hat{Z}$
Computation of model $\hat{Z}=\hat{Z}_0(1+\delta\hat{Z})$

Figure 3.2 Algorithms for estimating unmeasured quantities.

Considering the above information, the initial step determines a baseline model $\hat{Z}_0 = f\left(\vec{U}\right)$ using healthy engine data from Sample 0. For one unmeasured variable and four operating conditions $\vec{U}$, the complete second-order polynomial that simulates a healthy engine behavior has the following structure:

$$\hat{Z}_0\left(\vec{U}\right) = a_0 + a_1u_1 + a_2u_2 + a_3u_3 + a_4u_4 + a_5u_1u_2 + a_6u_1u_3 + a_7u_1u_4 + a_8u_2u_3 + a_9u_2u_4 + a_{10}u_3u_4 + a_{11}u_1^2 + a_{12}u_2^2 + a_{13}u_3^2 + a_{14}u_4^2 \tag{3.6}$$

Considering all unmeasured variables at n operating points, baseline values are described by a linear system $\mathbf{Z_0} = \mathbf{VA}$ and the solution for unknown coefficients $\hat{a}_{ij}$ stored in a $(k \times r)$-matrix $\hat{\mathbf{A}}$ is given by the LSM and expressed as:

$$\hat{\mathbf{A}} = \left(\mathbf{V}^{\mathrm{T}}\mathbf{V}\right)^{-1}\mathbf{V}^{\mathrm{T}}\mathbf{Z_0} \tag{3.7}$$

where $\mathbf{Z_0}$ is $(n \times r)$-matrix of true values for r unmeasured variables and $\mathbf{V}$ is a $(n \times k)$-matrix containing k components ($1, u_1, u_2, ..., u_3^2, u_4^2$) from operating conditions. After finding the model coefficients $\hat{a}_{ij}$, a baseline function estimation $\vec{\hat{Z}}_0\left(\vec{U}\right)$ for all unmeasured variables can be now determined using Eq. (3.6). Also, a comparison between the estimates $\hat{Z}_0$ and true values Z_0 is available. Figure 3.3 shows baseline estimates $\hat{Z}_0$ of an important unmeasured variable Z_2 (combustion chamber temperature T_{CC}). The estimates were computed using the "Filtered noise" scheme.

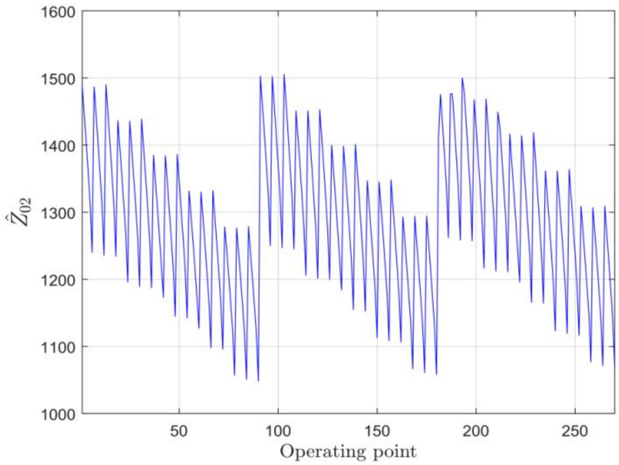


Figure 3.3 Baseline estimates $\hat{Z}_{02}$ (combustion chamber discharge temperature T_{CC}) for filtered noise

A cyclic behavior of the plotted variables is explained by a mode to change operating conditions during generating the data samples. As the estimates and true values differ a little and will not be seen in Figure 3.3, the difference between them is plotted in Figure 3.4 in the form of a relative estimation error $\varepsilon\hat{Z}_{02} = \frac{\hat{Z}_{02} - Z_{02}}{Z_{02}}$. It is visible that most of errors are smaller than 0.1% and the maximum error 0.6% takes place only in one point. Table 3.4 shows the Root-Mean Square Errors (RMSE) of baseline estimates $\hat{Z}_0$ for all the unmeasured variables and the three noise schemes. For the "No noise" scheme, the input data is ideally accurate, and $\varepsilon\hat{Z}_0$ is an approximation error. It varies for different variables, but its level is comparable to the errors of the baseline for measured variables [69]. The error $\varepsilon\hat{Z}_{04}$ is very small (corresponds to a computer error) because the variable Z_4 (high pressure turbine efficiency) is constant at baseline conditions. Comparing the results for the three schemes, the filtered noise influences a little. However, if the measurements are not averaged ("Full noise" scheme), the baseline errors are doubled (on average, from 0.0014 to 0.0027).

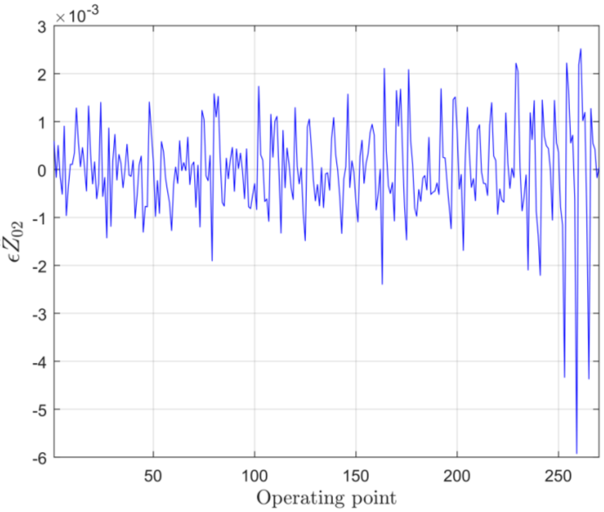


Figure 3.4 Baseline estimation errors $\varepsilon\hat{Z}_{02}$ for filtered noise

Table 3.4 RSME of baseline estimates $\hat{Z}_0$ (all noise schemes).

Noise schemes	$\varepsilon\hat{Z}_{01}$	$\varepsilon\hat{Z}_{02}$	$\varepsilon\hat{Z}_{03}$	$\varepsilon\hat{Z}_{04}$	$\varepsilon\hat{Z}_{05}$	$\varepsilon\hat{Z}_{06}$	Average
No noise	0.0011	8.6995e-04	0.0011	1.8684e-15	0.0028	0.0011	0.0014
Filtered noise	0.0012	9.7631e-04	0.0011	2.4357e-16	0.0028	0.0014	0.0015
Full noise	0.0033	0.0023	0.0012	1.6543e-16	0.0029	0.0041	0.0027

Model $\hat{Z} = f(\vec{U},\vec{Y})$ for faulty conditions

As in the case of a healthy engine behavior, estimated unmeasured values $\hat{Z} = f(\vec{U},\vec{Y})$ (under the influence of faulty conditions) can be computed using a full second-order polynomial. For each unmeasured variable, the polynomial now needs 66 coefficients and the function has the following structure:

$$\hat{Z}\left(\vec{U},\vec{Y}\right) = a_0 + a_1u_1 + a_2u_2 + ... + a_{10}y_6 + a_{11}u_1u_2 + a_{12}u_1u_3 + ... + a_{55}y_5y_6 + a_{56}u_1^2 + a_{57}u_2^2 + $$
$$+ a_{58}u_3^2 + a_{59}u_4^2 + a_{60}y_1^2 + a_{61}y_2^2 + a_{62}y_3^2 + a_{63}y_4^2 + a_{64}y_5^2 + a_{65}y_6^2$$
(9)

The coefficients for all variables are determined by the LSM in the same mode as for baseline model $\vec{\hat{Z}}_0(\vec{U})$. In this case, the coefficients are computed using a $(n \times r)$-matrix $\mathbf{Z}$ of true values in a concatenated form considering all operating points from Samples 0 to 6. As an example of the results, the plot of estimate $\hat{Z}_2$ (combustion chamber discharge temperature T_{CC}) is shown in Figure 3.5. The faulty condition corresponds here to the fault parameter of compressor air consumption G_c, and the random errors in input data are simulated according to the "Filtered noise" scheme. One can see that the behavior of estimate $\hat{Z}_2$ is similar to the behavior of the baseline estimate $\hat{Z}_{02}$ shown in Figure 3.3. In other words, the fault introduced does not have any visible influence on the unmeasured variables. This is one more illustration that unmeasured variables themselves are not pertinent fault indicators, and it is necessary to calculate their deviations.

To have exact deviations, not only the baseline model must be adequate, but the current values $\vec{Z}$ should be precisely estimated as well. Estimation errors $\varepsilon\hat{Z}$ for the unmeasured variable Z_2 are shown in Figure 3.6. It is visible that the estimating accuracy is sufficiently high: an average error of about 0.05% and a maximum error of 0.17%.

Table 3.5 shows the RMSE of estimates $\hat{Z}$ computed for Samples 0-6, the "Filtered noise" scheme for all samples and all unmeasured variables. As observed, the errors are considerably small for all the variables excepting the variable Z_5 of power turbine efficiency.

For the three noise schemes, Table 3.6 presents the RMSE of estimates $\hat{Z}$ conserving the format of Table 3.4 for comparison purposes. As observed in Table 3.6 for the "No noise" scheme, approximation errors decrease significantly. The explanation is a greater flexibility of the function $\vec{\hat{Z}} = f(\vec{U},\vec{Y})$ (66 unknown parameters in one polynomial against 15 parameters in the baseline function). Nevertheless, the impact of measurement noise on the accuracy of unmeasured variables is notable. On average, the error $\varepsilon\hat{Z}$ results in 0.0038 for the "Full noise" scheme and 0.0027 for the baseline. However, the scheme "Filtered noise" provides more acceptable results. As before, the variable Z_5 has the highest errors.

After determining the models for unmeasured variables under healthy and faulty conditions, the final step consists in calculating the deviations.

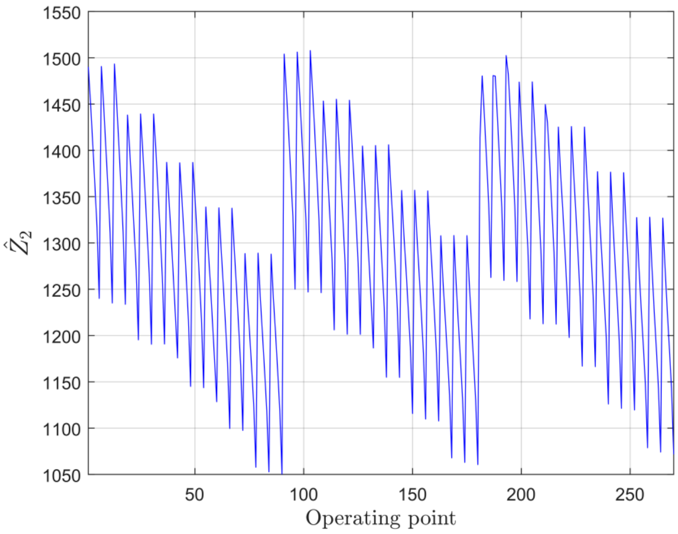


Figure 3.5 Estimates $\hat{Z}_2$ for faulty condition $G_c = -0.03$ and filtered noise.

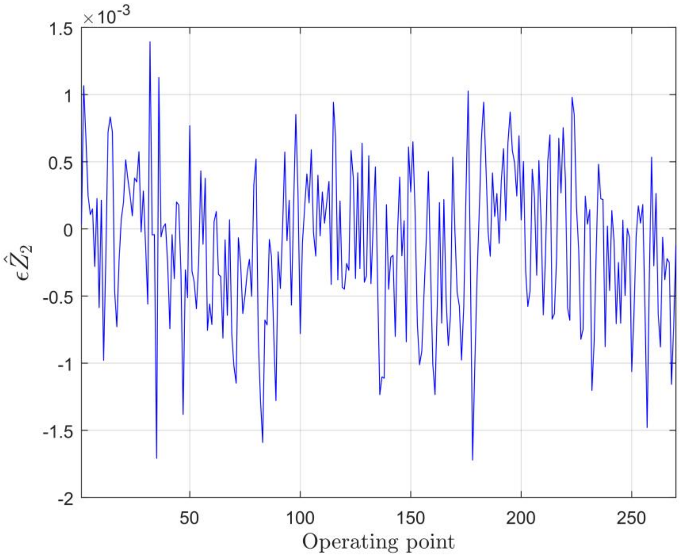


Figure 3.6 Estimation errors $\varepsilon\hat{Z}_2$ for faulty condition $G_c = -0.03$ and filtered noise.

Table 3.5 RMSE of estimates $\hat{Z}$ ("Filtered noise" scheme).

	$\varepsilon\hat{Z}_1$	$\varepsilon\hat{Z}_2$	$\varepsilon\hat{Z}_3$	$\varepsilon\hat{Z}_4$	$\varepsilon\hat{Z}_5$	$\varepsilon\hat{Z}_6$	
0	4.404e-04	4.858e-04	9.925e-04	5.658e-04	0.0025	8.066e-04	0.0012
1	4.283e-04	5.544e-04	0.0012	7.748e-04	0.0027	8.29e-04	0.0013
2	4.636e-04	4.785e-04	0.0013	5.391e-04	0.0023	8.507e-04	0.0012
3	5.198e-04	4.411 e-04	9.876e-04	4.517e-04	0.0023	0.0012	0.0012
4	4.693e-04	4.221e-04	0.0015	0.0016	0.0027	0.0018	0.0016
5	4.012e-04	4.775e-04	9.06e-04	5.476e-04	0.0025	0.0014	0.0013
6	4.789e-04	4.543e-04	9.796e-04	4.925e-04	0.0031	8.374e-04	0.0014

Table 3.6 RMSE of estimates $\hat{Z}$ (all noise schemes).

Noise schemes	$\varepsilon\hat{Z}_1$	$\varepsilon\hat{Z}_2$	$\varepsilon\hat{Z}_3$	$\varepsilon\hat{Z}_4$	$\varepsilon\hat{Z}_5$	$\varepsilon\hat{Z}_6$	Average
No noise	3.881e-05	2.026e-04	5.96e-04	3.817e-04	8.152e-04	4.133e-04	4.7939e-04
Filtered noise	4.587e-04	4.75e-04	0.0011	8.134e-04	0.0025	0.0011	0.0013
Full noise	0.0019	0.0015	0.0034	0.0024	0.0072	0.0036	0.0038

Fault-induced deviations $\delta\hat{Z}$

Once the baseline model $\vec{\hat{Z}}_0(\vec{U})$ and the model $\vec{\hat{Z}} = f(\vec{U},\vec{Y})$ are available, the computation of fault-induced deviations $\delta\vec{\hat{Z}}$ is now available. According to Eq. (3.5), the deviation estimate is given by $\delta\hat{Z} = \dfrac{\hat{Z}(\vec{U},\vec{Y}) - \hat{Z}_0(\vec{U})}{\hat{Z}_0(\vec{U})}$. The final set of experiments consists in the comparison of these estimates $\delta\hat{Z}$ with true deviations δZ obtained by direct use of the thermodynamic model data. To this end, estimated and true deviations for all the six unmeasured variables Z were computed for the healthy and six faulty conditions (Samples 0-6). To illustrate the results, deviation estimates $\delta\hat{Z}_2$ induced by the fault $G_c = -0.03$ and estimation errors $\varepsilon\delta\hat{Z}_2$ are plotted in Figure 3.7. These results correspond to the "Filtered noise" scheme. The plots show that the fault-induced deviations are by far greater than their errors. Thus, despite the errors in actual and baseline values of unmeasured variables, their deviations have acceptable accuracy and carry important diagnostic information.

Table 3.7 presents the RMSE of deviation estimates $\delta\hat{Z}$ for Samples 0-6, "Filtered noise" scheme and all unmeasured variables. Observing these data, one can see that the errors are acceptably low for all variables (excepting the variable Z_5), and the deviations of unmeasured variables can be used to diagnose real GTE. Table 3.8 contains the RMSE of deviation estimates $\delta\hat{Z}$ for all noise schemes. These errors allow drawing the following conclusions that are in accordance with the results of the errors of unmeasured variables themselves. Total errors for the "Full noise" scheme

(0.0039 on average) seem to be too great; however, the scheme “Filtered noise” results in acceptable total deviation errors (0.0018 on average). Thus, if a diagnostic system filters raw input data, the deviations computed for important unmeasured variables can be used in the system to extend its capabilities. Once more, the variable Z_5 of power turbine efficiency has the highest errors. The point is that this mayor element is a relatively independent component, and under given operating conditions, gas path variables excepting the power turbine output do not depend on the power turbine efficiency.

To reduce the errors of estimated unmeasured quantities $\vec{\hat{Z}}$ and $\delta\vec{\hat{Z}}$, another algorithm is proposed in the following subsection.

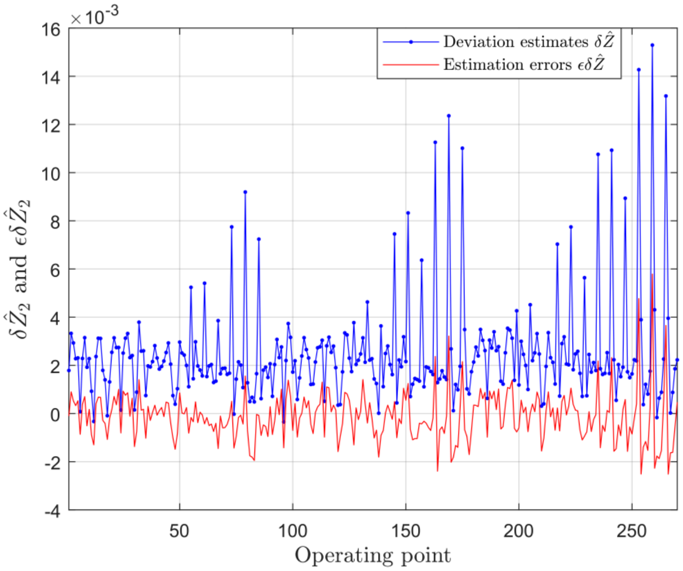


Figure 3.7 Deviation estimates $\delta\hat{Z}_2$ and estimation errors $\varepsilon\delta\hat{Z}_2$ (filtered noise).

Table 3.7 RMSE of deviation estimates $\delta\hat{Z}$ (filtered noise).

	$\varepsilon\delta\hat{Z}_1$	$\varepsilon\delta\hat{Z}_2$	$\varepsilon\delta\hat{Z}_3$	$\varepsilon\delta\hat{Z}_4$	$\varepsilon\delta\hat{Z}_5$	$\varepsilon\delta\hat{Z}_6$	
0	0.0012	0.0010	0.0016	5.6589e-04	0.0037	0.0012	0.0018
1	0.0012	0.0010	0.0017	7.7482e-04	0.0037	0.0013	0.0019
2	0.0012	0.0010	0.0017	5.3919e-04	0.0035	0.0011	0.0018
3	0.0013	9.4205e-04	0.0015	4.5174e-04	0.0035	9.8323e-04	0.0017
4	0.0011	9.5001e-04	0.0020	0.0016	0.0038	0.0020	0.0021
5	0.0011	0.0010	0.0016	5.4767e-04	0.0038	0.0020	0.0020
6	0.0012	9.8818e-04	0.0016	4.9254e-04	0.0039	0.0011	0.0019

Table 3.8 RMSE of deviation estimates $\delta\hat{Z}$ (all noise schemes).

Noise schemes	$\varepsilon\delta\hat{Z}_1$	$\varepsilon\delta\hat{Z}_2$	$\varepsilon\delta\hat{Z}_3$	$\varepsilon\delta\hat{Z}_4$	$\varepsilon\delta\hat{Z}_5$	$\varepsilon\delta\hat{Z}_6$	Average
No noise	0.0011	9.0857e-04	0.0013	3.8031e-04	0.0029	0.0011	0.0015
Filtered noise	0.0011	9.9572e-04	0.0016	7.9948e-04	0.0036	0.0014	0.0018
Full noise	0.0022	0.0017	0.0035	0.0024	0.0073	0.0037	0.0039

3.5.2 Algorithm 2

In Algorithm 1, the function $\vec{\hat{Z}} = f(\vec{U},\vec{Y})$ may have elevated errors because it includes 10 arguments, most of them are interrelated, and their magnitudes are very different. To avoid these difficulties, we propose Algorithm 2 that is based on a simplified function $\delta\vec{\hat{Z}} = f(\delta\vec{Y})$. The new function has only six arguments, and they do not depend on operating conditions. The algorithm includes four steps.

Step 1. On the data of a healthy engine (Sample 0), two polynomial functions $\vec{\hat{Y}}_0 = f(\vec{U})$ and $\vec{\hat{Z}}_0 = f(\vec{U})$ are determined in the same manner as in Algorithm 1. The former is a typical baseline model described in Section 2.2. It is quite accurate and widely used in gas turbine diagnostics. The latter has the same structure and therefore should also possess a high accuracy. The data of estimation errors in Table 3.4 confirm the accuracy of the function $\vec{\hat{Z}}_0 = f(\vec{U})$.

Step 2. Using known functions $\vec{\hat{Y}}_0 = f(\vec{U})$ and $\vec{\hat{Z}}_0 = f(\vec{U})$ as well as faulty engine values *Y* and *Z* from Samples 1-6, deviations $\delta Y = \left(Y - \hat{Y}_0(\vec{U})\right)/\hat{Y}_0(\vec{U})$ and $\delta Z = \left(Z - \hat{Z}_0(\vec{U})\right)/\hat{Z}_0(\vec{U})$ are computed for both measured and unmeasured variables.

Step 3. By approximating the deviations obtained in the previous step, deviations of unmeasured variables are determined as the following polynomial function of monitored variable deviations:

$$\delta\hat{Z} = f(\delta\vec{Y}) \tag{3.8}$$

Step 4. Finally, the unmeasured variables can be calculated by the expression:

$$\hat{Z} = \hat{Z}_0(1+\delta\hat{Z}) \tag{3.9}$$

Figure 3.8 presents an example of deviations estimated by Eq. (3.8) as well as their estimation errors $\varepsilon\delta\hat{Z}$ for filtered noise. Table 3.9 contains the RMSE of estimates $\delta\hat{Z}$ for all noise schemes. Comparing Figure 3.7 and Figure 3.8, we can note some reduction of the errors $\varepsilon\delta\hat{Z}_2$. According to

Table 3.8 and Table 3.9, this positive effect is about 10% on average for all noise schemes. In terms of average errors for all variables, the error reduction is even more considerable, about 18%. Thus, Algorithm 2 certainly yields more accurate estimates of deviations for unmeasured variables.

Since deviations of both measured and unmeasured variables are induced by faults, the variability of faults conditions affects the accuracy of the model $\delta\vec{\hat{Z}} = f(\delta\vec{Y})$ used in Algorithm 2. The next subsection describes Algorithm 3 that addresses this problem.

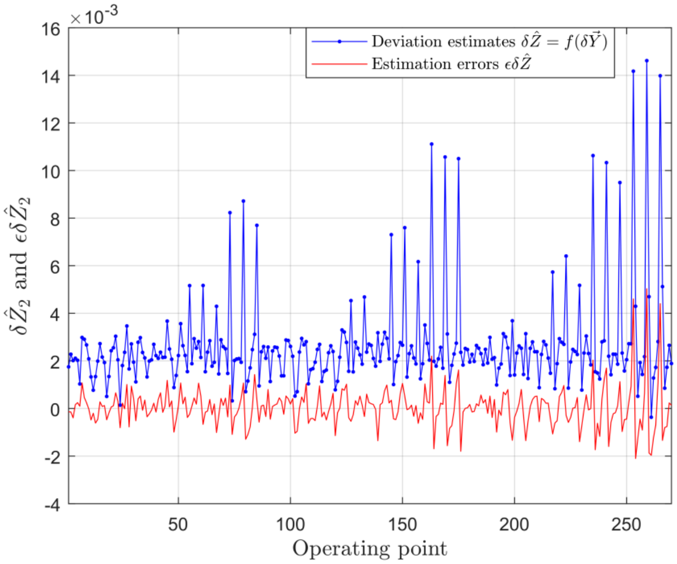


Figure 3.8 Deviation estimates $\delta\hat{Z}_2$ and estimation errors $\varepsilon\delta\hat{Z}_2$ for filtered noise (Algorithm 2).

Table 3.9 RMSE of deviation estimates $\delta\hat{Z}$ (all noise schemes, Algorithm 2).

Noise schemes	$\varepsilon\delta\hat{Z}_1$	$\varepsilon\delta\hat{Z}_2$	$\varepsilon\delta\hat{Z}_3$	$\varepsilon\delta\hat{Z}_4$	$\varepsilon\delta\hat{Z}_5$	$\varepsilon\delta\hat{Z}_6$	Average
No noise	0.0011	8.7563e-04	0.0013	2.5686e-04	0.0020	0.0012	0.0012
Filtered noise	0.0012	8.7412e-04	0.0015	5.8956e-04	0.0025	0.0014	0.0015
Full noise	0.0025	0.0015	0.0030	0.0018	0.0055	0.0033	0.0032

3.5.3 Algorithm 3

Since real engine deterioration corresponds to specific fault conditions, the idea arises to verify the algorithm of estimating unmeasured variables under such fault conditions, more narrow than the conditions analyzed before. This promises further enhancement of estimation accuracy. For stationary power plants as in the case of the engine under analysis, compressor fouling presents the principal deterioration mechanism. In a simplified form, the fouling can be represented by considering the information of Samples 1 and 2 individually generated by compressor air consumption and efficiency single faults respectively. In this way, in Algorithm 3 we repeat all the steps of Algorithm 2 with the exception that only these two samples are considered to form the model $\delta\vec{\hat{Z}} = f(\delta\vec{Y})$. The resulting errors $\varepsilon\delta\hat{Z}$ are presented in Figure 3.9 by the plots for a combustion chamber discharge temperature. Table 3.10 contains the RMSE of deviation estimates $\delta\hat{Z}$ for all variables and noise schemes. It follows from the figure and the second column of Table 3.10 that the level of errors has not changed for the combustion chamber discharge temperature. However, Table 3.10 also shows that averaged errors have reduced considerably: by 20% for the filtered noise and by 56% for the full noise. That is, the deviations computed for unmeasured variables are very accurate not only for filtered input data, but also for raw measured variables. Thus, from the point of accuracy, there are no limitations to use Algorithm 3 for computing important unmeasured variables and their deviations in real diagnostic systems.

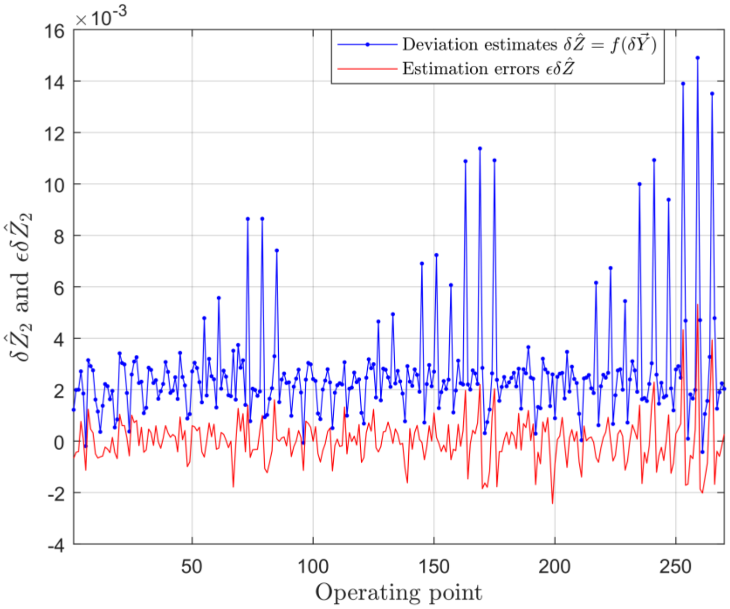


Figure 3.9 Deviation estimates $\delta\hat{Z}_2$ and estimation errors $\varepsilon\delta\hat{Z}_2$ for filtered noise (Algorithm 3).

Table 3.10 RMSE of deviation estimates $\delta\hat{Z}$ (all noise schemes, Algorithm 3).

Noise schemes	$\varepsilon\delta\hat{Z}_1$	$\varepsilon\delta\hat{Z}_2$	$\varepsilon\delta\hat{Z}_3$	$\varepsilon\delta\hat{Z}_4$	$\varepsilon\delta\hat{Z}_5$	$\varepsilon\delta\hat{Z}_6$	Average
No noise	0.0011	8.7467e-04	0.0012	1.7786 e-15	0.0019	0.0011	0.0012
Filtered noise	0.0011	8.7837e-04	0.0013	1.1558e-04	0.0012	0.0010	0.0010
Full noise	0.0017	0.0014	0.0022	4.2291e-04	9.3095e-04	0.0013	0.0014

3.6 ESTIMATION OF UNMEASURED QUANTITIES FOR MORE COMPLEX FAULT SCENARIOS

In practice, GTE problems can have a multi-fault nature. In order to know if unmeasured quantities are correctly estimated under more complex scenarios, samples with multiple faults are included in the analysis. Based on the previous computations, Algorithm 2 has been chosen because it is more accurate than Algorithm 1 and more general than Algorithm 3. Two cases are considered:

-Case 1: Input data without division. Healthy and faulty engine scenarios, i.e. Samples 0-12 (see Table 3.3) are united in a total array used for both training and validation.

-Case 2: Input data with division. Samples 0-12 united in a total array are randomly partitioned into two sets: training set of 90% and validation set of 10%, and a ten-fold cross validation is performed.

For Case 1 and a filtered noise scheme, Figure 3.10 shows all the deviation estimates $\delta\hat{Z}$ and estimation errors $\varepsilon\delta\hat{Z}$ of variable Z_2. We can see that each of the 12 faulty scenarios (samples 1-12) consisting of 270 operating points is well distinguishable here. For nine faults the deviations are by far greater than their errors, for one fault the deviations are still distinguishable against the errors, and for two faults the deviations and the errors are comparable. Bad visibility of these two faults is explained by their low influence on the specific variable Z_2, while the errors have practically the same low level as with other faults. Low visibility of the two faults for the variable Z_2 does not mean that these faults cannot be diagnosed because the deviations can be much greater for the other variables as shown in Figure 3.11 and Figure 3.12.

Table 3.11 presents the RMSE of deviation estimates $\delta\hat{Z}$ for all unmeasured variables and noise schemes. Comparing

Table 3.11 and Table 3.9, we can state that the error level has not increased because of adding new fault scenarios. Let us now analyze the results for Case 2 that can be considered as more reliable due to the cross validation use.

For Case 2 the errors of the training and validation stages were firstly compared and it was found that they are practically equal. Let us now analyze the validation errors in more detail. Table 3.12 contains the RMSE of deviation estimates $\delta\hat{Z}$ for validation stage. Figure 3.13 shows deviation estimates $\delta\hat{Z}$ and their estimations errors for validation stage using filtered noise. Because the order of faults and operating points is random here we cannot analyze particular faults. However, in general, most of deviations are clearly detectable and estimations errors $\varepsilon\delta\hat{Z}$ remain small. Comparing Table 3.12 and

Table 3.11, one can see that the differences are negligible. The similarity between Case 1 and Case 2 and between the training and validation results within Case 2 confirms that all the calculations are correct and equally accurate. This also means that the results previously obtained in Section 4 are also valid despite the absence of a separate validation stage.

We can conclude that the inclusion of multiple faults to the input data, which leads to a more complex fault scenario, does not affect the estimation of unmeasured quantities. The high accuracy is explained here by a great excess of input data relative to unknown quantities that is beneficial to the LSM accuracy.

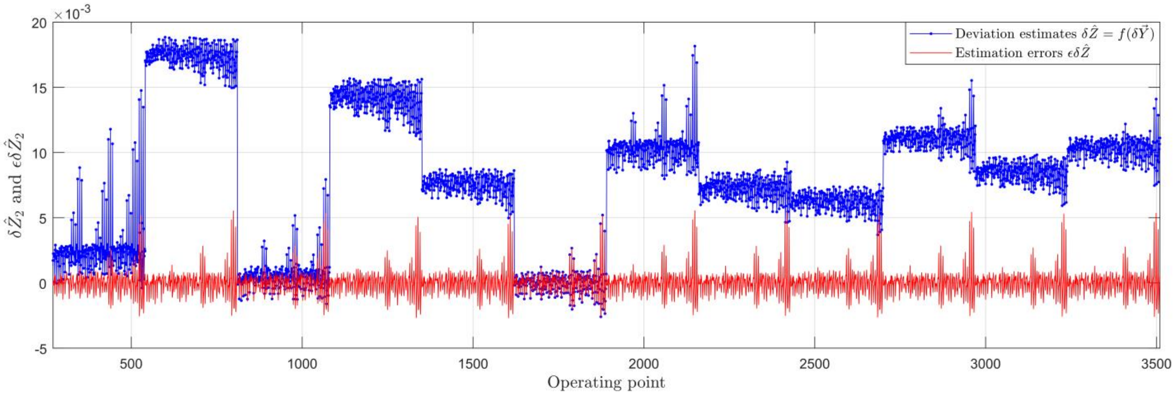


Figure 3.10 Deviation estimates $\hat{\delta Z}_2$ and estimation errors $\varepsilon\hat{\delta Z}_2$ for Case 1 (filtered noise).

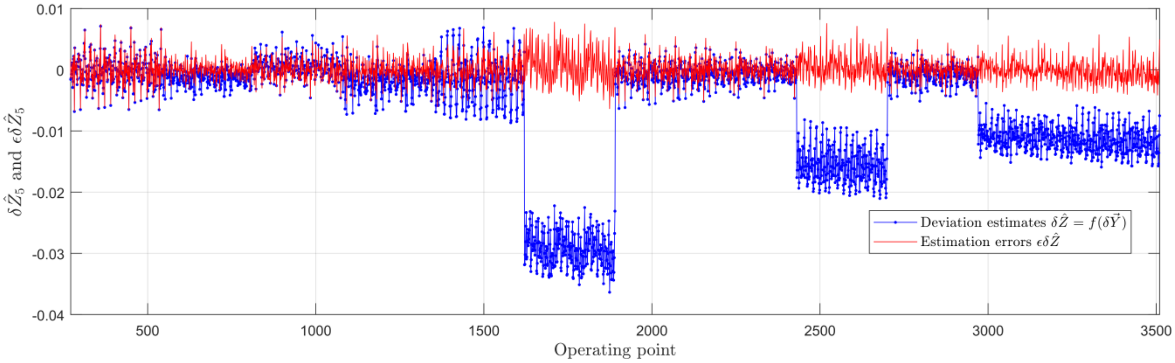


Figure 3.11 Deviation estimates $\hat{\delta Z}_5$ and estimation errors $\varepsilon\hat{\delta Z}_5$ for Case 1 (filtered noise).

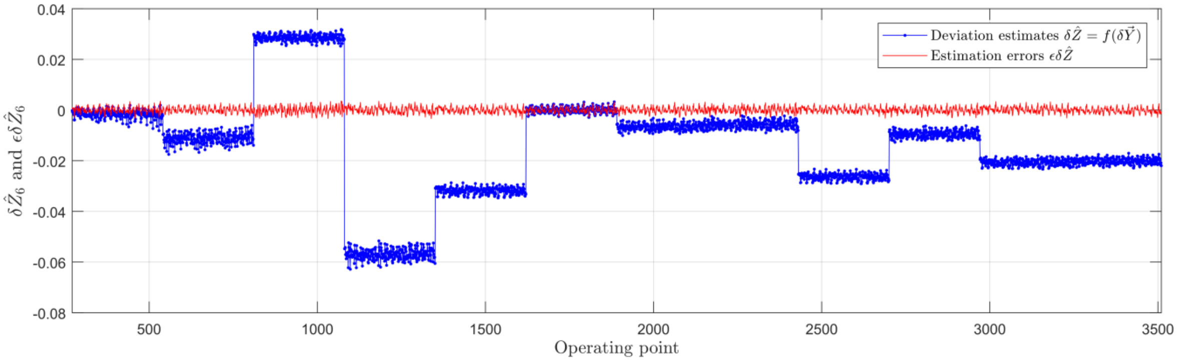


Figure 3.12 Deviation estimates $\hat{\delta Z}_6$ and estimation errors $\varepsilon\hat{\delta Z}_6$ for Case 1 (filtered noise).

Table 3.11 RMSE of deviation estimates $\hat{\delta Z}$ for Case 1.

Noise schemes	$\varepsilon\hat{\delta Z}_1$	$\varepsilon\hat{\delta Z}_2$	$\varepsilon\hat{\delta Z}_3$	$\varepsilon\hat{\delta Z}_4$	$\varepsilon\hat{\delta Z}_5$	$\varepsilon\hat{\delta Z}_6$	Average
No noise	0.0010	0.0008	0.0013	0.0003	0.0019	0.0012	0.0012
Filtered noise	0.0013	0.0009	0.0016	0.0007	0.0024	0.0014	0.0015
Full noise	0.0030	0.0017	0.0035	0.0024	0.0056	0.0033	0.0035

Table 3.12 RMSE of deviation estimates $\delta\hat{Z}$ for Case 2 (validation stage).

Noise schemes	$\varepsilon\delta\hat{Z}_1$	$\varepsilon\delta\hat{Z}_2$	$\varepsilon\delta\hat{Z}_3$	$\varepsilon\delta\hat{Z}_4$	$\varepsilon\delta\hat{Z}_5$	$\varepsilon\delta\hat{Z}_6$	Average
No noise	0.0010	0.0008	0.0013	0.0003	0.0019	0.0012	0.0012
Filtered noise	0.0013	0.0009	0.0016	0.0007	0.0023	0.0014	0.0015
Full noise	0.0030	0.0017	0.0034	0.0024	0.0055	0.0033	0.0034

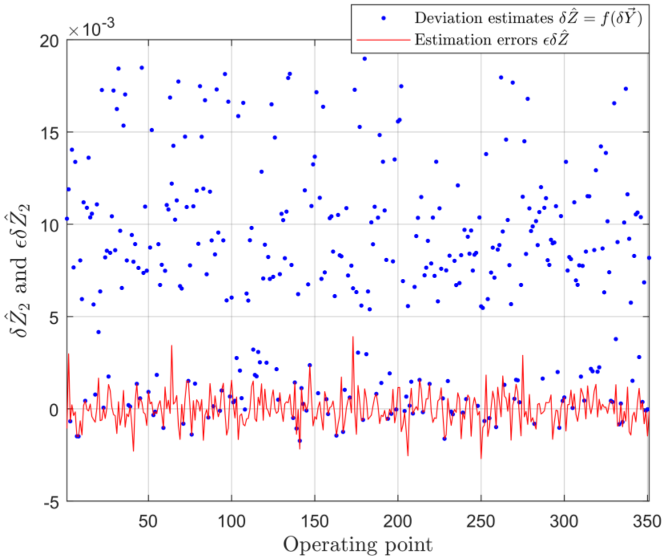


Figure 3.13 Deviation estimates $\delta\hat{Z}_2$ and estimation errors $\varepsilon\delta\hat{Z}_2$ for Case 2 and validation stage (filtered noise).

Chapter 4: Evaluation of Gas Turbine Diagnostic Techniques under Variable Fault Conditions

4.1 OVERVIEW

The present chapter proposes to an algorithm to evaluate different gas turbine diagnostic techniques under a principle of variable fault classification to study possible fault scenarios present in real gas turbine maintenance and create complex and realistic fault classifications. Through an adaptable algorithm, the variable classification makes it easy to set the type of class used, different fault parameters, class quantity, fault development directions, fault severity boundary type, engine components and scheme of deviation noise. Based on this principle, twelve classifications variations have been created for examining themselves and comparing the techniques. These classifications contain single or multiple classes as well as their mixtures. In addition, the chapter introduces and investigates a new boundary for fault severity. With this boundary, the fault class description becomes more realistic thus providing more confidence to diagnosis results. The chapter also addresses the influence of real deviation errors on operation of diagnostic techniques and final diagnosis accuracy. A special procedure is developed to compare diagnostic techniques and compute the probability of correct diagnosis (true positive rate), which is used as the evaluation criterion. Four comparative studies are considered. They analyze the influence of different pattern numbers, operating modes, multiple-class boundaries and deviation noise schemes. Within each comparative study, the techniques are evaluated for many classification variations. Such analysis allows drawing solid conclusions on techniques accuracy.

Figure 4.1 shows all the influence factors (highlighted boxes) considered in the development of the algorithm and they involve: a turboshaft engine for industrial application; diagnosis of gas path and measurement systems; the stages of feature extraction and fault identification; the use of LSM to identify degraded models; one-point diagnostic analysis; the use of physics-based, data-driven, non-linear and linear models; the use of MLP, RBN, PNN and SVM as fault recognition techniques; and the identification of both single and multiple faults.

4.2 FOUNDATIONS OF ANN AND SVM

As mentioned above, ANNs and SVMs have been chosen in the present chapter for gas turbine fault recognition. The following subsections briefly describe them. Additional information about these techniques can be found in the literature, for example in [59,60,136–139].

4.2.1 Multi-Layer Perceptron

The MLP with a backpropagation algorithm is an artificial neural network consisting of a predefined set of input-target pairs that uses propagation-adaptation in the training stage. Let us consider two layers: a hidden layer and an output layer as shown in Figure 4.2.

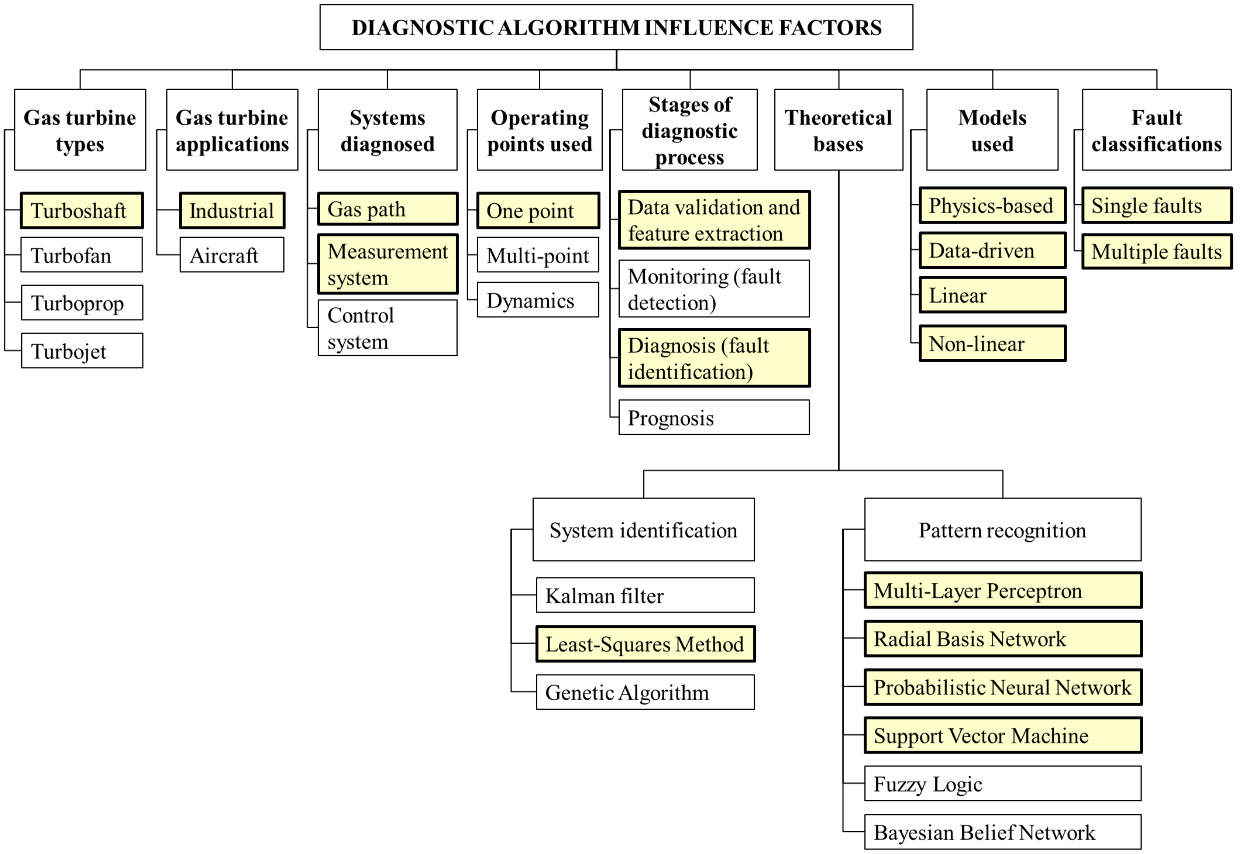


Figure 4.1 Diagnostic algorithm influence factors used throughout Chapter 4.

For the hidden layer, its input vector $\mathbf{n}_1$ is given by the sum of an input pattern vector $\mathbf{Z}$ multiplied by weights contained in the matrix $\mathbf{W}_1$ plus a bias vector $\mathbf{b}_1$. Then, $\mathbf{n}_1$ is transformed using a differentiable transfer function producing the output vector $\mathbf{a}_1$. For the output layer, the procedure is repeated using $\mathbf{a}_1$ as an input to the layer and producing a network output vector $\mathbf{a}_2$ by transforming a vector $\mathbf{n}_2$. The intention of the training is to minimize a delta, which is the difference between the output vector $\mathbf{a}_2$ and its target vector. The training algorithm uses an error function or a surface associated with the network that tries to find the stable state of minimum error through the descending path of the error surface. Therefore, it feeds back the system error to update all weights and biases in a proportional value to the decreasing gradient of the error function. This updating results in the network's ability to learn relationships between the inputs and outputs. When a new input is presented, the outputs of the nearby learning input vectors determine the new output. In our study, the training is performed in a batch mode, i.e., a total error between all targets and network outputs of the learning set is computed and used to update weights in each iteration (also called epoch).

4.2.2 Radial Basis Network

The RBN receives its name because it includes a layer with radial basis function (RBF) neurons. The input n to a radial basis transfer function is a Euclidean distance between a weight vector $\mathbf{w}$ and an input vector $\mathbf{Z}$, multiplied by a bias $b = 1/\sigma$ (i.e., $n = \|\mathbf{w} - \mathbf{Z}\| b$), where σ is the spread that allows changing the RBF neuron influence area. The transfer function computes a neuron's output as $a = \exp\left(-(n)^2\right)$. The idea of the RBF neurons is to measure how close the input vector $\mathbf{Z}$ and a weight vector are from each other. If $a = 0$, they will be totally different but if $a = 1$, they will be identical. Figure 4.3 shows the RBN containing two layers. The first one is a radial basis layer with S_1 neurons. A vector $\mathbf{n}_1$ of distances is created between one vector of an input weight matrix $\mathbf{W}_1$ and the input vector $\mathbf{Z}$, multiplied by a bias vector $\mathbf{b}_1$. Then, a radial basis transfer function converts $\mathbf{n}_1$ into a hidden layer output vector $\mathbf{a}_1$. The second layer is the output layer and it generates linear combinations of activations of the radial basis layer. A vector $\mathbf{n}_2$ is formed multiplying the RBF output vector $\mathbf{a}_1$ and a weight matrix $\mathbf{W}_2$ plus a bias vector $\mathbf{b}_2$. A linear transfer function transforms $\mathbf{n}_2$ resulting in an output vector $\mathbf{a}_2$ of S_2 elements. In an "exact design" of the RBN algorithm, every input vector of the training set forms a new RBF neuron. In an "efficient design" one neuron at a time is iteratively added to the radial basis layer. This new neuron is created by the input vector that obtains the smallest network error. The neuron addition is stopped when a network error decreases below an error goal or when a maximum neuron number has been reached.

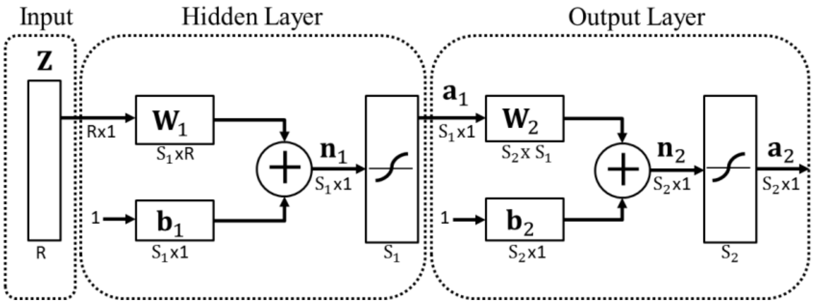


Figure 4.2 Structure of MLP (adapted from [139]).

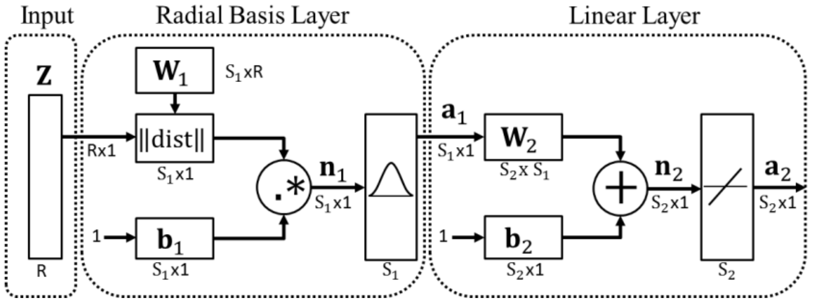


Figure 4.3. Structure of RBN (adapted from [139]).

4.2.3 Probabilistic Neural Network

The main characteristic of the PNN (Figure 4.4) is that its hidden layer works in the same manner as the layer of the RBN exact design. However, the output layer is different. Each output neuron corresponds to one class and is called a classification neuron. Each RBF neuron, which is based on one training pattern, is connected with only one classification neuron corresponding to the class to which the pattern belongs. The sum of all contributions a_{1_j} related to the training patterns of the class is a probability of this class. To compute an entire input vector for the classification layer, a matrix $\mathbf{W}_2$ composed of 0's and 1's is multiplied by $\mathbf{a}_1$ giving as a result a vector $\mathbf{n}_2$ of probabilities of all classes. Finally, to classify an input vector $\mathbf{Z}$, a competitive transfer function selects the class with the maximum probability producing a 1 for this class and 0's for the remaining thus forming an output vector $\mathbf{a}_2$.

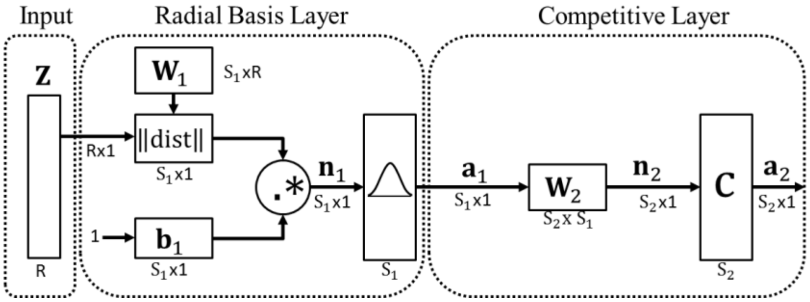


Figure 4.4 Structure of PNN (adapted from [139]).

4.2.4 Support Vector Machines

Any hyperplane can be written in the space $\mathbb{R}^P$ as the set of points **Z** satisfying:

$$\mathbf{w}^{\mathrm{T}}\mathbf{Z} + b = 0 \tag{4.1}$$

where **w** is a vector perpendicular to the hyperplane and *b* is the bias. Let us present training data of two classes as pattern vectors $\mathbf{Z}_i \in \mathbb{R}^P$, i=1,…,N, and their corresponding labels $y_i \in (-1,1)$ indicating the class to which the pattern $\mathbf{Z}_i$ belongs. If the training data are linearly separable, two parallel hyperplanes without points between them can be built to divide the data. The hyperplanes can be given by $\mathbf{w}^{\mathrm{T}}\mathbf{Z}_i+b=1$ and $\mathbf{w}^{\mathrm{T}}\mathbf{Z}_i+b=-1$. The margin is defined to be the distance between them and is equal to $2/\|\mathbf{w}\|$ (Figure 4.5). Intuitively, it measures how good the separation between two classes is. The points divided in this manner satisfy the following constraint:

$$y_i\left(\mathbf{w}^{\mathrm{T}}\mathbf{Z}_i + b\right) \geq 1 \quad \text{for}\, i = 1,\ldots, N \tag{4.2}$$

The basic idea of SVMs is to find the hyperplanes that produce the maximal margin. In this way, SVMs need to solve the following optimization problem:

$$\min \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} \tag{4.3}$$

$$\text{subject to:} \quad y_i\left(\mathbf{w}^{\mathrm{T}}\mathbf{Z}_i + b\right) \geq 1 \quad \text{for}\, i = 1,\ldots, N$$

Introducing the Karush-Kuhn-Tucker (KKT) multipliers $\alpha_i \geq 0$, Eq. (4.3) can be transformed into:

$$L = \min_{\mathbf{w},b} \max_{\alpha} \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} - \sum_{i=1}^{N} \alpha_i \left(y_i\left(\mathbf{w}^{\mathrm{T}}\mathbf{Z}_i + b\right) - 1\right) \tag{4.4}$$

As can be seen, Eq. (4.4) depends on **w**, *b* and *α*. At the extremes we have:

$$\nabla_{\mathbf{w}} L = \mathbf{w} - \sum_{i=1}^{N} \alpha_i y_i \mathbf{Z}_i = 0 \tag{4.5}$$

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{N} \alpha_i y_i = 0 \tag{4.6}$$

Substituting Eq. (4.5) and (4.6) in Eq. (4.4), we obtain:

$$L = \min_{\alpha} \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j \alpha_i \alpha_j \mathbf{Z}_i^{\mathrm{T}} \mathbf{Z}_j - \sum_{i=1}^{N} \alpha_i \tag{4.7}$$

subject to: $\alpha_i \geq 0$ and $\sum_{i=1}^{N} \alpha_i y_i = 0 \quad \text{for } i = 1, \ldots, N$

Eq. (4.7) can be also expressed in a simplified form as L=min ½ ($\alpha^{\mathrm{T}}Q\alpha - 1^{\mathrm{T}}\alpha$). Here, Q is the matrix of quadratic coefficients. This expression is minimized now only in function of α and the solution can be obtained by Quadratic Programming. After solving Eq. (4.7), the optimal w can be obtained from Eq. (4.5). Only the training examples Z_i that have non-zero α_i contribute to the solution and they are called support vectors. Finally, the decision function to classify a new pattern vector **Z** is:

$$\operatorname{sgn}\left(\mathbf{w}^{\mathrm{T}}\mathbf{Z} + b\right) = \operatorname{sgn}\left(\sum_{i=1}^{N} \alpha_i y_i \mathbf{Z}_i^{\mathrm{T}} \mathbf{Z} + b\right) \tag{4.8}$$

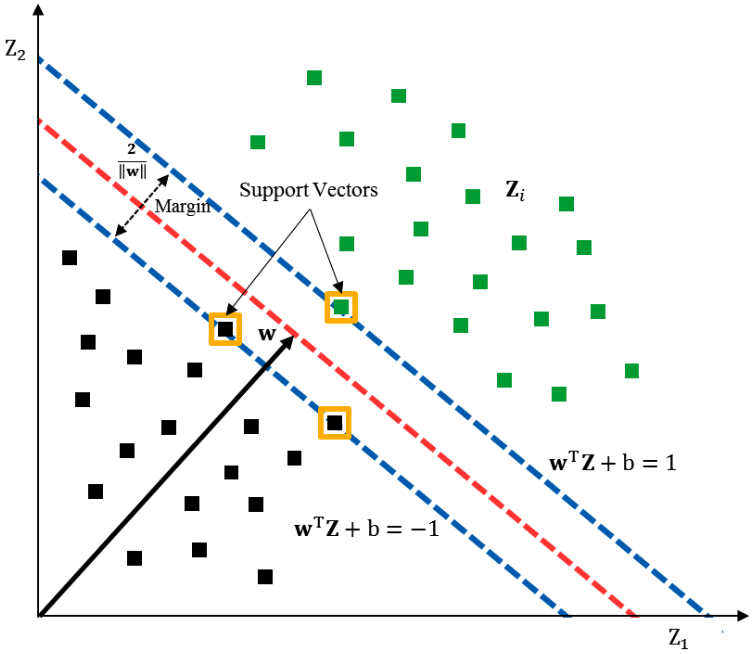

Z_2
$\frac{2}{\|w\|}$
Margin
Support Vectors
$\mathbf{Z}_i$
w
$\mathbf{w}^{\mathrm{T}}\mathbf{Z} + \mathrm{b} = 1$
$\mathbf{w}^{\mathrm{T}}\mathbf{Z} + \mathrm{b} = -1$
Z_1

Figure 4.5 Hyperplanes and margin of separation.

When applications handle non-separable cases [60] and a complete separation between classes is not always possible [59], the optimization problem becomes:

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_{i=1}^{N} \xi_i \tag{4.9}$$

subject to:

$$y_i\left(\mathbf{w}^{\mathrm{T}}\Phi(\mathbf{Z}_i) + b\right) \geq 1 - \xi_i \quad \text{and} \quad \xi_i \geq 0 \quad \text{for } i = 1, \ldots, N \tag{4.10}$$

where $\Phi(\mathbf{Z}_i)$ maps $\mathbf{Z}_i$ from original input space into a higher-dimensional feature space to separate data there with a maximum-margin hyperplane as shown in Figure 4.6, C is a penalty parameter and ξ_i

denotes non-negative slack variables to measure the degree of misclassification (Figure 4.7). The decision function is given by:

$$\operatorname{sgn}\left(\mathbf{w}^{\mathrm{T}}\Phi(\mathbf{Z})+b\right)=\operatorname{sgn}\left(\sum_{i=1}^{N}\alpha_i y_i \mathrm{K}(\mathbf{Z}_i,\mathbf{Z})+b\right) \tag{4.11}$$

where K($\mathbf{Z}_i$,$\mathbf{Z}_j$) is the kernel function related to the transformation Φ($\mathbf{Z}_i$) by the equation:

$$\mathrm{K}\left(\mathbf{Z}_i,\mathbf{Z}_j\right)=\Phi(\mathbf{Z}_i)^{\mathrm{T}}\Phi\left(\mathbf{Z}_j\right) \tag{4.12}$$

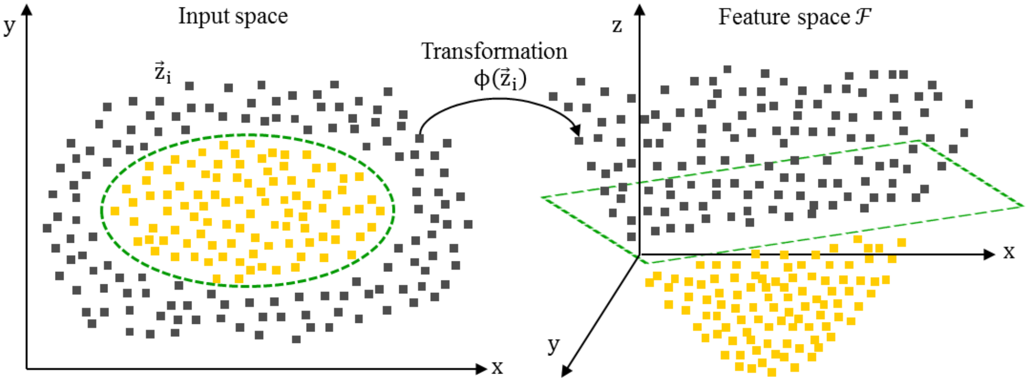


Figure 4.6 Transformation of input data into a feature space.

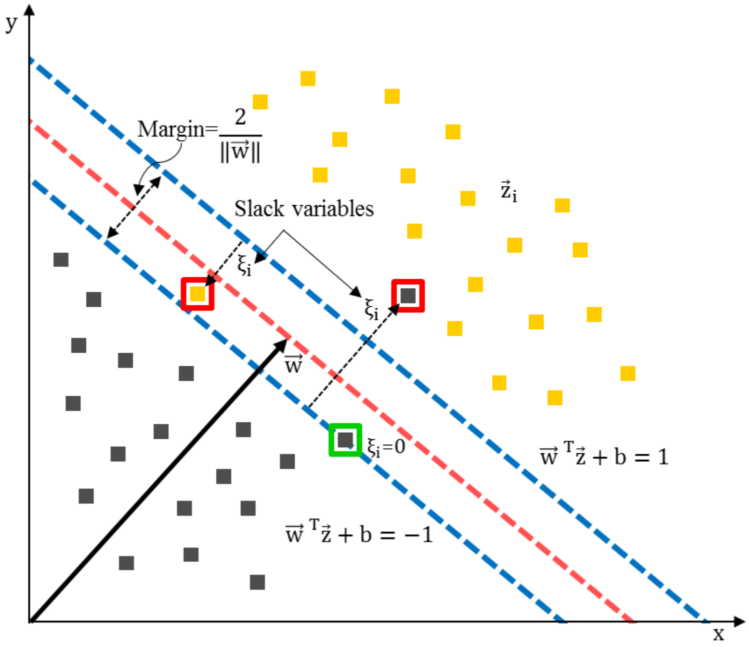

y
Margin=$\frac{2}{\|\vec{w}\|}$
Slack variables
$\vec{z}_i$
ξ_i
ξ_i
$\vec{w}$
ξ_i=0
$\vec{w}^{\mathrm{T}}\vec{z}+b=1$
$\vec{w}^{\mathrm{T}}\vec{z}+b=-1$
x

Figure 4.7 Misclassification errors and slack variables ξ_i.

Some common kernels include polynomial kernel, RBF kernel, sigmoid kernel, etc. Although SVMs are intended for binary models, they can address multi-class problems. There are different strategies to solve them. For example, in the One-Versus-One (OVO) strategy (Figure 4.8), $q(q-1)/2$ classifiers are constructed (q is the class number) and each one uses data from two classes.

At prediction stage, all binary classifiers emit votes when a pattern vector **Z** is presented. The pattern is assigned to the class with the maximum number of votes.

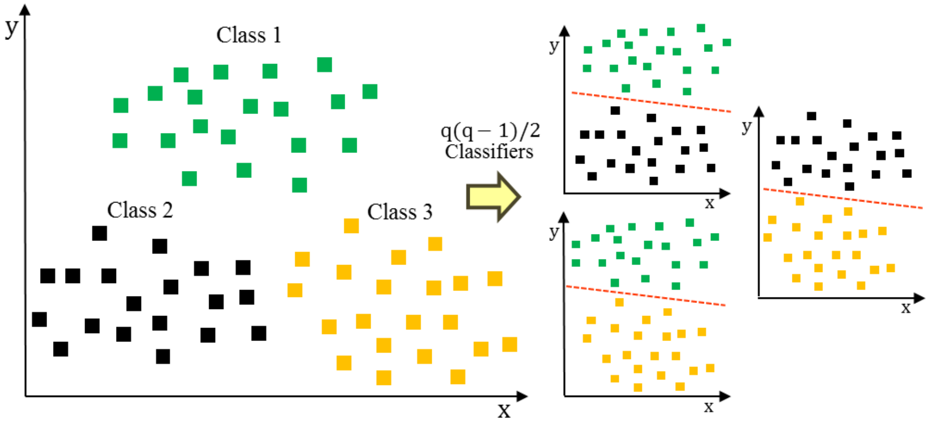


Figure 4.8 Multiclass problem using the OVO strategy.

4.3 DIAGNOSTIC TECHNIQUE EVALUATION PROCEDURE

To be evaluated, the recognition techniques are integrated into a stochastic evaluation procedure, which consists of the following main blocks: deviations, fault classification, training, validation, tuning and final diagnosis accuracy $\overline{P}$ (Figure 4.9). The procedure is implemented in Matlab using machine learning toolboxes.

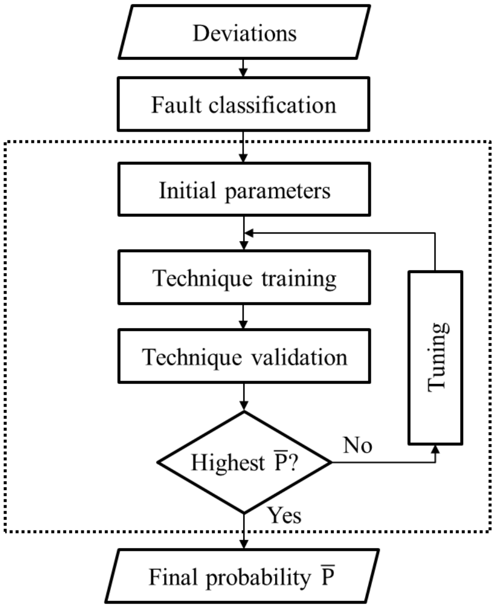


Figure 4.9 Diagnostic technique evaluation procedure.

4.3.1 Deviations

For the proposed methodology, the simulated deviations take the form:

$$\delta Y_i = \frac{Y_i\left(\mathbf{U},\boldsymbol{\Theta}_0+\Delta\boldsymbol{\Theta}\right)-Y_{0i}\left(\mathbf{U},\boldsymbol{\Theta}_0\right)}{Y_{0i}\left(\mathbf{U},\boldsymbol{\Theta}_0\right)} \qquad i=1,\ldots,m \tag{4.13}$$

where $\mathbf{Y}_0(\mathbf{U})$ is the baseline model, a vector $\mathbf{\Theta}_0$ corresponds to a healthy engine whereas fault parameters $\Delta\mathbf{\Theta}$ consider fault influence. These model-based deviations present base points to compute deviations for any value and combination of fault parameters. The deviation computation for an arbitrary fault parameter value is performed by a piecewise-linear interpolation between the base points. The deviation corresponding to some fault parameters is determined by the sum of their individual influences. Simulated deviations can be more realistic adding a normally distributed random noise ε_i. Additionally, to have a homogeneous diagnostic space, deviations are normalized resulting in:

$$Z_i^* = \left(\frac{Y_i(\mathbf{U},\mathbf{\Theta}_0 + \Delta\mathbf{\Theta}) - Y_{0i}(\mathbf{U},\mathbf{\Theta}_0)}{Y_{0i}(\mathbf{U},\mathbf{\Theta}_0)} + \varepsilon_i \right) \Big/ \sigma_{\delta Yi} \tag{4.14}$$

where $\sigma_{\delta Yi}$ is the amplitude of possible random fluctuations in the original deviation δY_i. Normalized deviations of all monitored variables constitute an $(m \times 1)$-vector $\mathbf{Z}^*$ forming a diagnostic space where the fault classification is constructed. A pattern to be recognized represents a vector in this space.

Although the use of simulated deviation measurement noise in gas turbine diagnostic algorithms is a common practice, real deviation errors can present different distributions that can affect the final diagnosis reliability. A procedure proposed to extract error components from deviations working with real data can be found in [140]. It is necessary a degraded engine model $\mathbf{Y}(\mathbf{U},\bar{t})$ obtained by the LSM and input data including multiple operating points with different degradation severity. Using this model, a real deviation error can be given by:

$$E_{\delta Y} = \frac{Y^* - Y(\mathbf{U},\bar{t})}{Y(\mathbf{U})} \tag{4.15}$$

It is worth mentioning that each real deviation error needs to be matched with its corresponding simulated variable and only use the measurement errors contained in the same interval where the degraded model was created to avoid noise displacements when they are integrated into simulated fault developments. Figure 4.10 displays an example of real deviation errors $\varepsilon_{\delta Y}$ in an interval of 2608 operating points for a specified monitored variable while Figure 4.11 displays these errors for two variables.

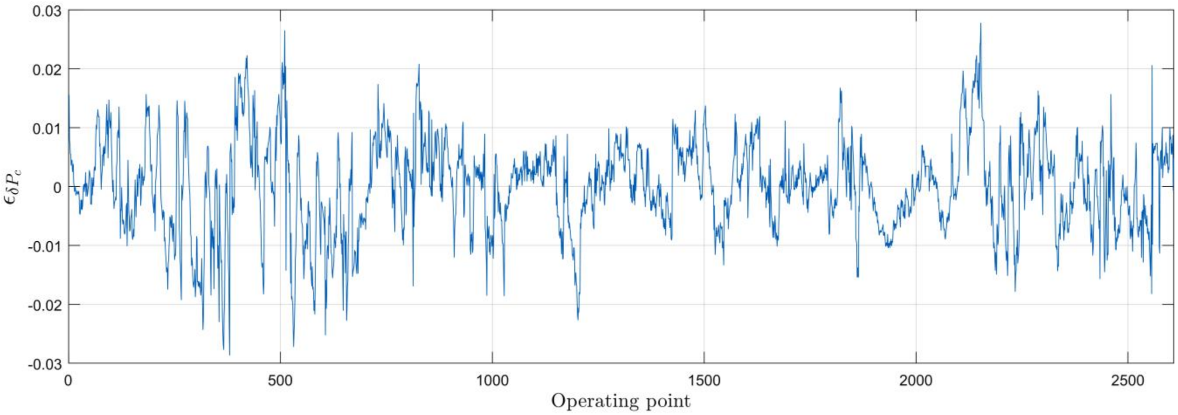


Figure 4.10 Real deviation noise vs. operating points

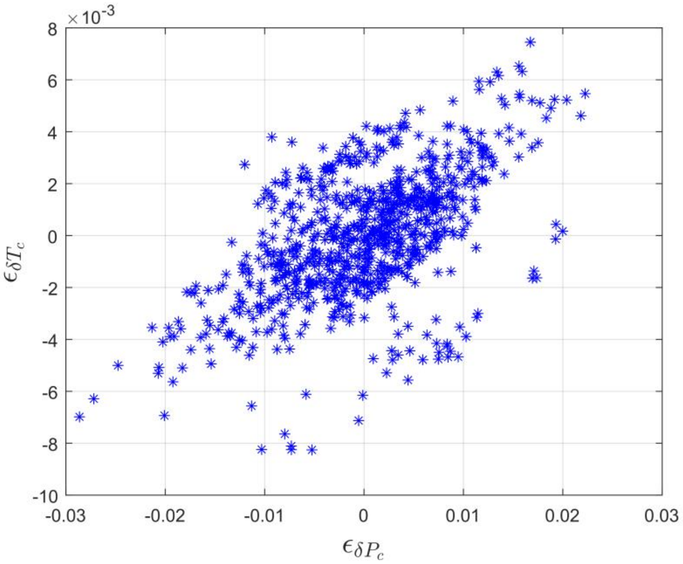


Figure 4.11 Real deviation noise for two monitored variables

4.3.2 Fault classification construction

After generating the model-based normalized deviations, they are used to build fault classifications required for diagnostics. Given that faults vary significantly in practice, it is necessary to describe them using a limited number of classes. Each fault class is constructed from patterns, either with the change of one fault parameter (single fault class) or with the independent change of some fault parameters (multiple fault class). This last type of class can be explained by the fact that faults can simultaneously appear in different engine components. A uniform distribution of fault parameter values inside of interval (0, ±5%) is employed to describe random fault severity. The limit “0” gives the possibility to simulate no-fault states while the limit “±5%” corresponds to the maximal change of the component performances, at which gas turbines lose their operation capacity due to deterioration and faults [118]. To know if a current pattern $\mathbf{Z}^*$ belongs to a specific class D_j, the criterion $R_j = R(\mathbf{Z}^*, D_j),\ j = 1,..., q$ is applied. When all values R_j are obtained, a decision rule can be applied as:

$$d = d_l \text{ if } R_l = \max(R_1, R_2,..., R_q) \tag{4.16}$$

where d is a possible diagnosis corresponding to a correct classification. This subsection only introduces the general idea of fault classification construction. The principle of a variable structure classification and a new fault severity boundary are explained in detail in Section 4.5.

4.3.3 Training and validation

A learning set $\mathbf{Z}_L$ includes patterns of all classes and is employed to train the techniques under analysis. Every technique is trained on known entry pairs: the input pattern vector $\mathbf{Z}^*$ and its target. Since it is not sufficient to achieve high accuracy in training, a common strategy is to have additional data for validation and pay attention to its accuracy. In this way, these new data called validation set $\mathbf{Z}_V$ are used to verify whether the technique can generalize the fault description. This set is created in the same way as $\mathbf{Z}_L$. The only exception is the use of different series of random numbers that are involved in the computation of fault severity and errors in the deviations. As in the case of the learning set, every pattern in the validation set belongs to a known class.

4.3.4 Evaluation criterion (diagnosis accuracy)

In an effort to tune and compare all the techniques proposed, an averaged accuracy performance is determined for each of them. The technique analyzed classifies the patterns of the set $\mathbf{Z}_V$, producing the diagnosis d_j. Comparing d_j with a known class D_l for all validation set patterns, a confusion matrix is formed whose diagonal contains a vector $\mathbf{P}$ of correct pattern classification probabilities (a.k.a. true positive rates) for each fault class. A mean number $\overline{P}$ of these probabilities determines the total accuracy of engine fault recognition. It is a criterion to tune and evaluate the techniques. By analyzing the confusion matrix no diagonal elements, the direct influence of each fault class on the recognition accuracy of the other classes becomes visible.

4.3.5 Tuning

In order to perform an adequate evaluation, internal parameters of each technique should be tailored to ensure the maximal probability $\overline{P}$. For MLP, the principal parameters to tune are the number of iterations, the type of backpropagation training algorithm and the number of hidden layer neurons. In the case of RBN, the spread σ and the number of hidden layer neurons are varied independently until the best combination producing the highest $\overline{P}$ is selected. As for PNN, the only parameter to tailor is the spread σ. Finally, SVMs use k-fold cross validation to improve the prediction accuracy. Since in the present work SVM uses the RBF kernel, the parameters to tune are σ and the regularization parameter C. In the k-fold cross validation, the set $\mathbf{Z}_L$ is randomly partitioned into k subsets of equal size. One subset is used as validation data for testing the model trained on the remaining k−1 subsets. This process is repeated k iterations with each of the k subsets used exactly once as the validation data. The k results from the iterations can then be averaged to produce a single estimation. This procedure helps to prevent the overfitting problem. Commonly k=10 is recommended. It is not known with anticipation which C and σ are the most appropriate to accurately predict unknown data. In order to find these parameters, a grid-search with exponentially growing sequences of C and σ is applied. It allows finding the combination of parameters that yields the lowest generalization error giving the highest diagnosis accuracy. After finding the best parameters, users can repeat the learning on the entire training set and generate the final SVM model. A graphical example of the SVM tuning is depicted in Figure 4.12 and Figure 4.13. Since doing a complete grid-search for parameter selection is a time-consuming process, a general grid is recommended first (Figure 4.12). After identifying a better zone on the grid, a fine search on that region is conducted (Figure 4.13).

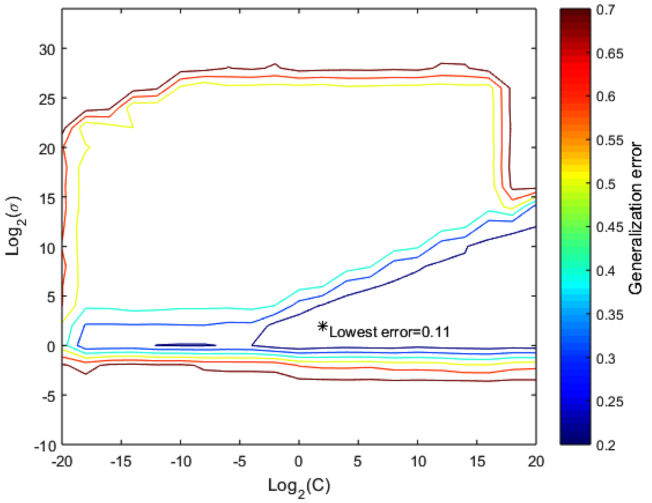


Figure 4.12 General grid-search for SVM parameters.

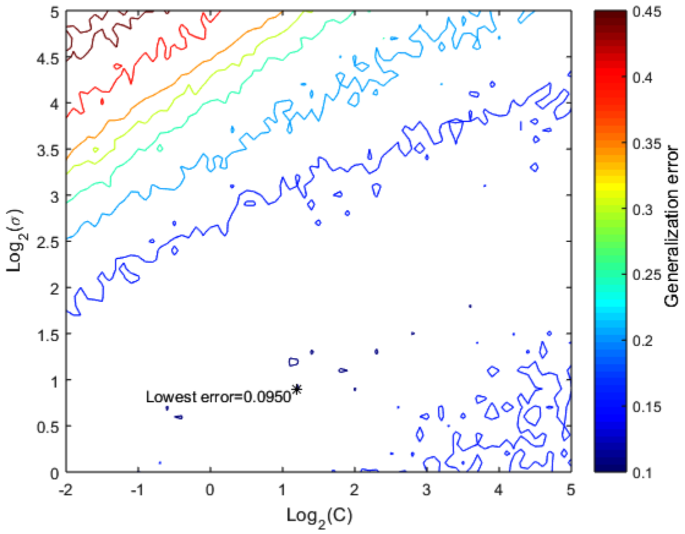


Figure 4.13 Fine grid-search for SVM parameters.

4.4 SELECTED ENGINE VARIABLES

The non-linear thermodynamic model corresponding to GT1 is used to construct the necessary fault classification. To simulate gas path and measurement system faults, the eighteen fault parameters from Table 4.1 are employed. The selection and significance of these fault parameters is based on the fact that they are commonly used in real gas turbine condition monitoring systems (e.g. efficiencies and flow capacities) to diagnose engine component faults [72].

Table 4.1 Parameters to simulate faults.

No.	Fault parameter description	Symbol	Severity
	Gas path parameters		
1	Compressor air flow	Gc	0 to −5%
2	Compressor efficiency	ηc	0 to −5%
3	Compressor turbine gas flow	Gt	0 to ±5%
4	Compressor turbine efficiency	ηt	0 to −5%
5	Power turbine gas flow	Gpt	0 to ±5%
6	Power turbine efficiency	ηpt	0 to −5%
7	Combustion chamber pressure recovery factor	σcc	0 to ±5%
8	Combustion efficiency	ηcc	0 to −5%
9	Inlet pressure loses factor	σin	0 to −5%
	Sensor parameters		
10	Compressor pressure sensor	Pc	0 to ±5%
11	Compressor turbine pressure sensor	Pt	0 to ±5%
12	Compressor temperature sensor	Tc	0 to ±5%
13	Compressor turbine temperature sensor	Tt	0 to ±5%
14	Power turbine temperature sensor	Tpt	0 to ±5%
15	Fuel flow sensor	Gf	0 to ±5%
16	Inlet pressure sensor	Pin	0 to ±5%
17	Inlet temperature sensor	Tin	0 to ±5%
18	Compressor turbine speed sensor	nhp	0 to ±3%

Table 4.2 shows three variables used as operating conditions (vector $\vec{U}$). The six gas path monitored variables of Table 4.3 are commonly used as input data for diagnosing the engine. They correspond to an engine standard measurement system. For purposes of simplifying the construction of the fault classification, a linear model $\mathbf{Z} = \mathbf{H}\delta\mathbf{\Theta}$ is utilized. Table 4.4 presents the values of the normalized influence matrix $\mathbf{H}$ for such a model.

Table 4.2 Operating conditions (Vector $\vec{U}$)

No.	Description	Unit	Symbol
1	Ambient pressure	kPa	P_H
2	Inlet temperature	K	T_H
3	Gas generator rotation speed	rpm	n_C

Table 4.3 Gas path monitored variables (Vector $\vec{Y}$)

No.	Description	Unit	Symbol	$\sigma_{\delta Y}$
1	Compressor total pressure	kPa	P_C	0.015
2	High pressure turbine total pressure	kPa	P_{HPT}	0.015
3	Compressor total temperature	K	T_C	0.025
4	High pressure turbine total temperature	K	T_{HPT}	0.015
5	Power turbine total temperature	K	T_{PT}	0.020
6	Fuel gas mass flow	kg/hour	G_f	0.020

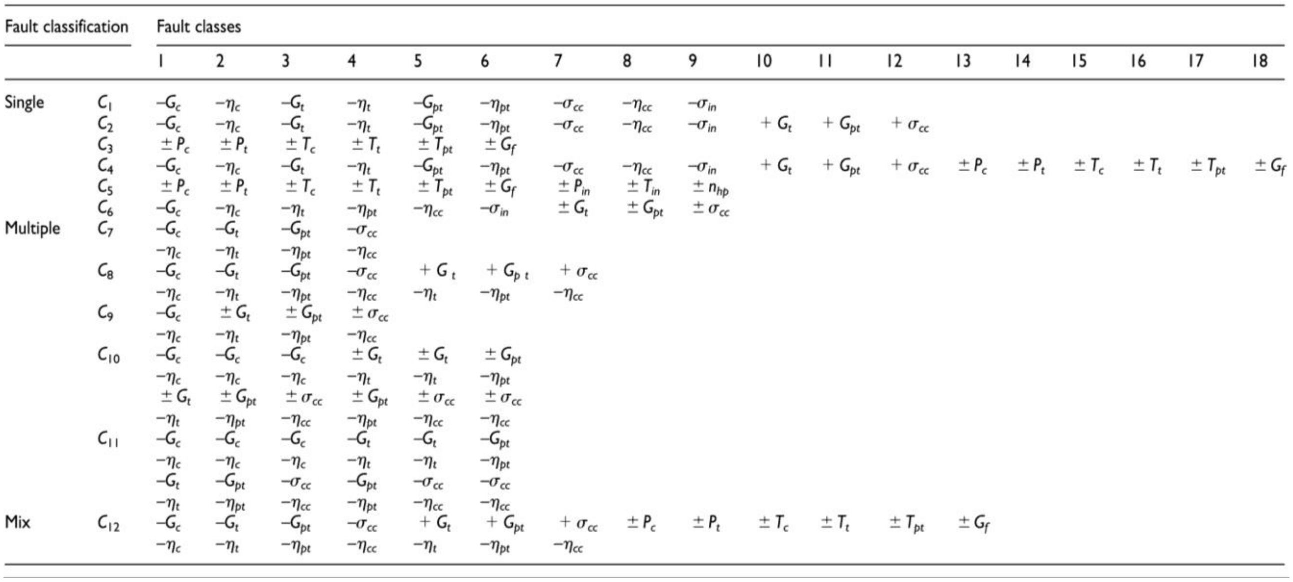
Table 4.4 Normalized influence matrix $\mathbf{H}$

ID	Type	Fault	Monitored variables					
			1	2	3	4	5	6
1	Gas path	Gc	80.2589	82.3136	13.6146	41.4010	19.4840	93.2054
2	Gas path	ηc	-38.7440	-38.9984	-28.8398	-85.0133	-61.2900	-97.5788
3	Gas path	Gt	-49.9866	17.2695	-8.22443	30.2896	21.1504	36.5676
4	Gas path	ηt	-43.3746	-53.6576	-7.1372	-116.0066	-83.2026	-132.6212
5	Gas path	Gpt	-24.8753	-97.3616	-4.0944	-66.6453	-35.9668	-76.1832
6	Gas path	ηpt	0.8955	3.3098	0.1474	2.1045	-15.7498	2.3448
7	Gas path	σcc	-78.0856	-14.0003	-12.8417	-37.9824	-27.8890	-41.4072
8	Gas path	ηcc	-0.4513	-0.8411	-0.0743	0.3002	0.0790	-52.1216
9	Gas path	σin	65.7392	67.7760	0.12607	1.7612	-9.7517	50.6790
10	Sensor	Pc	66.6666	0	0	0	0	0
11	Sensor	Pt	0	66.6666	0	0	0	0
12	Sensor	Tc	0	0	40	0	0	0
13	Sensor	Tt	0	0	0	66.6666	0	0
14	Sensor	Tpt	0	0	0	0	50	0
15	Sensor	Gf	0	0	0	0	0	50
16	Sensor	Pin	67.5466	70.4733	0.1640	2.6666	1.1950	52.9750
17	Sensor	Tin	-174.9133	-174.0666	4.1200	-32.4200	3.9000	-173.4050
18	Sensor	nhp	137.0266	142.4333	37.9040	115.6333	68.6750	197.9950

4.5 FAULT CLASSIFICATION VARIATIONS

Based on the idea that gas turbine fault classifications vary widely in practice, a principle of variable classification is proposed. For that purpose, an algorithm allows changing in a flexible and easy way the following elements: type of class used (single, multiple or mixed classes), pattern numbers, fault severity, class quantity, fault development directions (positive or negative changes), operating mode, noise scheme in deviations, type of boundary and engine components. Thus, the algorithm developed can work with more realistic fault classes. With the intention of studying the influence of classification structure on the final diagnostic accuracy of each technique, twelve fault classifications are introduced using this algorithm. These classifications are specified in Figure 4.14 and briefly described below. The classifications plotted in the diagnostic space $\mathbf{Z}$ present great differences in classification-to-classification pattern distributions. Thus, the recognition techniques will be evaluated under multiple and very different conditions.

Figure 4.14 Fault classification variations.



Fault classification		Fault classes																	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Single	C_1	$-G_c$	$-\eta_c$	$-G_t$	$-\eta_t$	$-G_{pt}$	$-\eta_{pt}$	$-\sigma_{cc}$	$-\eta_{cc}$	$-\sigma_{in}$									
	C_2	$-G_c$	$-\eta_c$	$-G_t$	$-\eta_t$	$-G_{pt}$	$-\eta_{pt}$	$-\sigma_{cc}$	$-\eta_{cc}$	$-\sigma_{in}$	$+G_t$	$+G_{pt}$	$+\sigma_{cc}$						
	C_3	$\pm P_c$	$\pm P_t$	$\pm T_c$	$\pm T_t$	$\pm T_{pt}$	$\pm G_f$												
	C_4	$-G_c$	$-\eta_c$	$-G_t$	$-\eta_t$	$-G_{pt}$	$-\eta_{pt}$	$-\sigma_{cc}$	$-\eta_{cc}$	$-\sigma_{in}$	$+G_t$	$+G_{pt}$	$+\sigma_{cc}$	$\pm P_c$	$\pm P_t$	$\pm T_c$	$\pm T_t$	$\pm T_{pt}$	$\pm G_f$
	C_5	$\pm P_c$	$\pm P_t$	$\pm T_c$	$\pm T_t$	$\pm T_{pt}$	$\pm G_f$	$\pm P_{in}$	$\pm T_{in}$	$\pm n_{hp}$									
	C_6	$-G_c$	$-\eta_c$	$-\eta_t$	$-\eta_{pt}$	$-\eta_{cc}$	$-\sigma_{in}$	$\pm G_t$	$\pm G_{pt}$	$\pm \sigma_{cc}$									
Multiple	C_7	$-G_c$ $-\eta_c$	$-G_t$ $-\eta_t$	$-G_{pt}$ $-\eta_{pt}$	$-\sigma_{cc}$ $-\eta_{cc}$														
	C_8	$-G_c$ $-\eta_c$	$-G_t$ $-\eta_t$	$-G_{pt}$ $-\eta_{pt}$	$-\sigma_{cc}$ $-\eta_{cc}$	$+G_t$ $-\eta_t$	$+G_{pt}$ $-\eta_{pt}$	$+\sigma_{cc}$ $-\eta_{cc}$											
	C_9	$-G_c$ $-\eta_c$	$\pm G_t$ $-\eta_t$	$\pm G_{pt}$ $-\eta_{pt}$	$\pm \sigma_{cc}$ $-\eta_{cc}$														
	C_{10}	$-G_c$ $-\eta_c$ $\pm G_t$ $-\eta_t$	$-G_c$ $-\eta_c$ $\pm G_{pt}$ $-\eta_{pt}$	$-G_c$ $-\eta_c$ $\pm \sigma_{cc}$ $-\eta_{cc}$	$\pm G_t$ $-\eta_t$ $\pm G_{pt}$ $-\eta_{pt}$	$\pm G_t$ $-\eta_t$ $\pm \sigma_{cc}$ $-\eta_{cc}$	$\pm G_{pt}$ $-\eta_{pt}$ $\pm \sigma_{cc}$ $-\eta_{cc}$												
	C_{11}	$-G_c$ $-\eta_c$ $-G_t$ $-\eta_t$	$-G_c$ $-\eta_c$ $-G_{pt}$ $-\eta_{pt}$	$-G_c$ $-\eta_c$ $-\sigma_{cc}$ $-\eta_{cc}$	$-G_t$ $-\eta_t$ $-G_{pt}$ $-\eta_{pt}$	$-G_t$ $-\eta_t$ $-\sigma_{cc}$ $-\eta_{cc}$	$-G_{pt}$ $-\eta_{pt}$ $-\sigma_{cc}$ $-\eta_{cc}$												
Mix	C_{12}	$-G_c$ $-\eta_c$	$-G_t$ $-\eta_t$	$-G_{pt}$ $-\eta_{pt}$	$-\sigma_{cc}$ $-\eta_{cc}$	$+G_t$ $-\eta_t$	$+G_{pt}$ $-\eta_{pt}$	$+\sigma_{cc}$ $-\eta_{cc}$	$\pm P_c$	$\pm P_t$	$\pm T_c$	$\pm T_t$	$\pm T_{pt}$	$\pm G_f$					

4.5.1 Single fault classifications

As illustrated in Figure 4.15, a single fault class D_1 is formed by only one health parameter. The point “0” indicates an engine normal state. The line 0-L_1 or Ω_1 reflects changes in deviations $\mathbf{Z}$ to a maximal fault severity limit L_1. The region Ω_1^* contains random deviation noise produced by measurement errors. A multiple class D_2 is built by summing the influence of the independent variations of two or more parameters forming the regions Ω_2 (without random errors) and Ω_2^* (with errors).

As shown in Figure 4.14, Classification 1 consists of nine single faults. Each fault is created by varying one gas path fault parameter in the negative direction. Classification 2 considers erosion and burnouts of hot part elements that can cause the increase of their flow performances. For this reason, positive changes for flow parameters of the compressor turbine, power turbine, and combustion chamber are introduced. With these parameters, three new classes are formed and added to Classification 1 resulting in twelve classes. Due to the frequency of sensor malfunctions, they are recommended to be diagnosed along with gas path faults. Since great measurement biases are easy to identify, only hidden incipient sensor faults are considered (small bias interval of ±5%).

In this way, for six monitored variables, six corresponding single classes form Classification 3. Figure 4.16 shows these six sensor faults; however, only those coinciding with their monitored variables can be completely observed (green and yellow classes). Classification 4 joins Classifications 2 and 3 to build eighteen single classes representing gas path faults and sensor malfunctions. Also, sensor malfunctions of operating condition parameters are simulated to take into account their influence on all monitored variables. Three single classes of this sensor fault type are created and joined to the previous six sensor faults (monitored variables) forming nine classes for Classification 5. Classification 6 considers nine classes: one compressor airflow fault, four efficiency faults for all components, one inlet pressure loses factor fault and finally, three faults with double direction for compressor turbine, power turbine and combustion chamber.

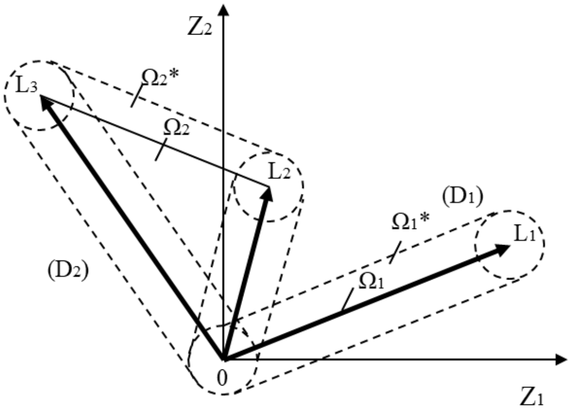


Figure 4.15 Single and multiple classes (adapted from [21]).

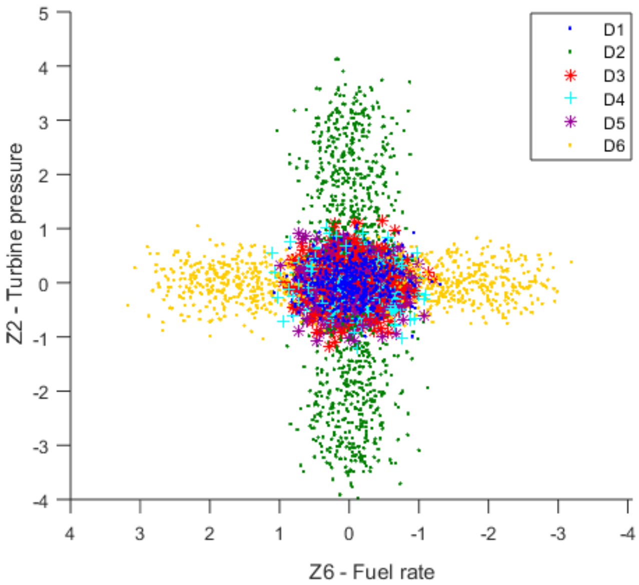


Figure 4.16 Fault classification 3.

4.5.2 Multiple fault classifications

Classification 7 includes four multiple classes grouped by engine component: compressor, combustion chamber, compressor turbine, and power turbine (Figure 4.17). These classes are formed by independent variation of two fault parameters of the same component (See Figure 4.14).

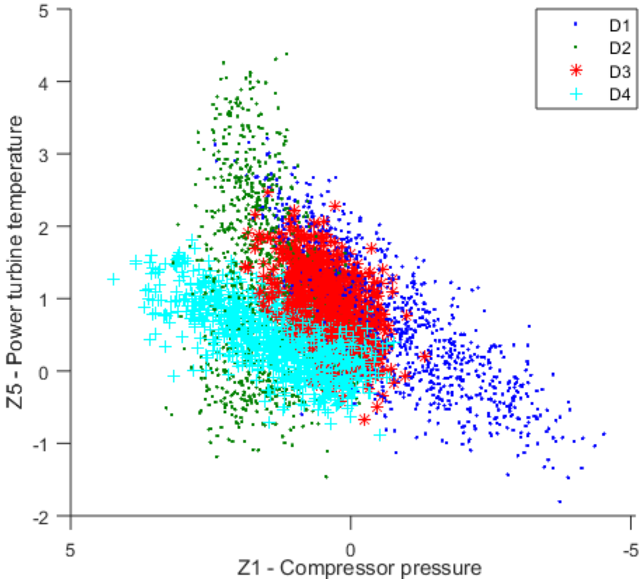


Figure 4.17 Fault classification 7.

Classification 8 contains seven classes formed by three multiple classes with positive changes for flow parameters and their respective efficiencies (Classification 2), and four classes from Classification 7. For Classification 9, four classes are formed as Classification 7 with the difference that flow parameters change in two directions for compressor turbine, power turbine and combustion chamber. Classification 10 contains six classes, each one created by four fault parameters (some of them with two fault development directions) of two engine components. It is formed by all possible combinations of compressor, combustion chamber, compressor turbine and power turbine (Figure 4.18). Classification 10 is closer to what really happens in a real gas turbine engine because it considers faults that can occur in two components at the same time. As for Classification 11, it is built in the same manner as the previous classification with the difference that the six classes include negative fault parameter changes (Figure 4.19).

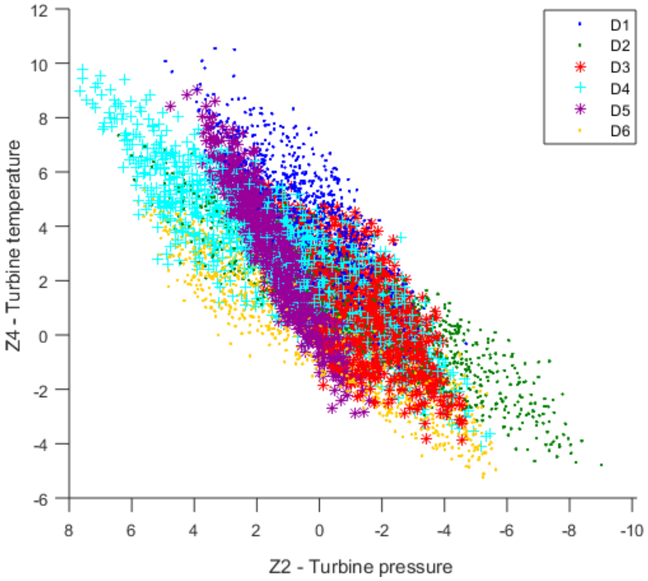


Figure 4.18 Fault classification 10.

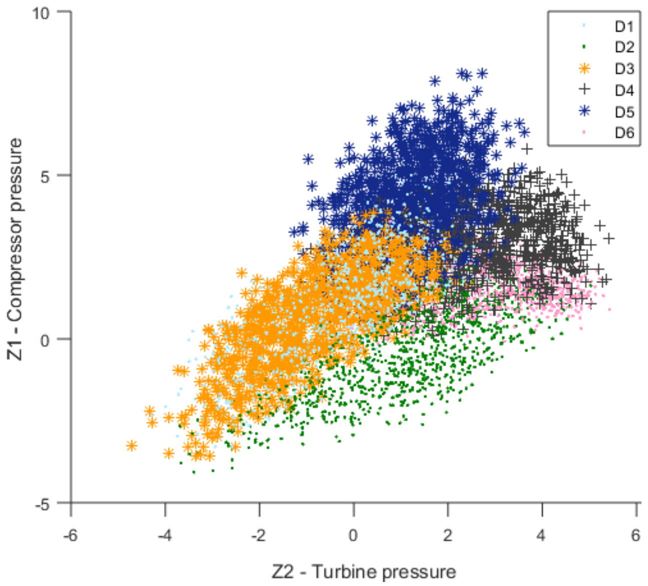


Figure 4.19 Fault classification 11.

4.5.3 Smooth fault severity boundary

When multiple faults are simulated by summing the influence of each fault parameter, there is a risk that the simulated fault exceeds the severity limit of real faults. To better understand the problem, let us consider a multiple class D_3 created by two fault parameters represented by vectors 0-L_1 and 0-L_2 as illustrated in Figure 4.20. The point “0” corresponds here to an engine normal state. Each of the vectors 0-L_1 and 0-L_2 reflects theoretical changes of one fault parameter. Fault severity increases to the engine health limit formed by points L_1, L_2 and vector lengths l_1 and l_2. It is clear that vectors $\mathbf{Z}$ (without errors ε_i) in the dotted part of the parallelogram can be longer than base vectors 0-L_1 and 0-L_2 produced by a maximal change of the corresponding fault parameters. In other words, simulated faults can have higher severity than real ones. In order to avoid this and to make a class formation more realistic, a linear boundary L_1L_2 that restricts fault pattern vectors inside the triangle 0-L_1L_2 was previously used4. However, that boundary is too restrictive when the angle θ_{12} increases.

It seems to us that a more appropriate boundary would be a smooth curve. For this reason, a new multiple-class boundary based on the Archimedean spiral is proposed (Figure 4.20). It is formed by the vector (blue line) that moves from L_1 to L_2 and gradually changes its length l from l_1 to l_2 proportionally to the turning angle. Thus, this length can be expressed as:

$$l = l_1 + (l_2 - l_1)\frac{\theta_i}{\theta_{12}} \qquad (4.17)$$

where θ_i is the angle between the current vector and the first base vector for a random pattern i, and θ_{12} is the angle between the two base vectors. Only the deviation vectors $\mathbf{Z}$ that are inside the curve are accepted. The described boundary can be easily extended to three fault parameters. The boundary vector of the length l determined in the plane of the first and second fault parameters (blue line) now is considered as a base vector. The second base vector 0-L_3 is produced by a third fault parameter. The boundary is determined in the plane of these two base vectors and is created in the same way, with a vector (orange line) that gradually changes its length from l_3 to l. For the case of four and more fault parameters, the boundary is determined similarly. A restrictive condition of this boundary is great for small angles between base vectors and decreases along with the angle increase.

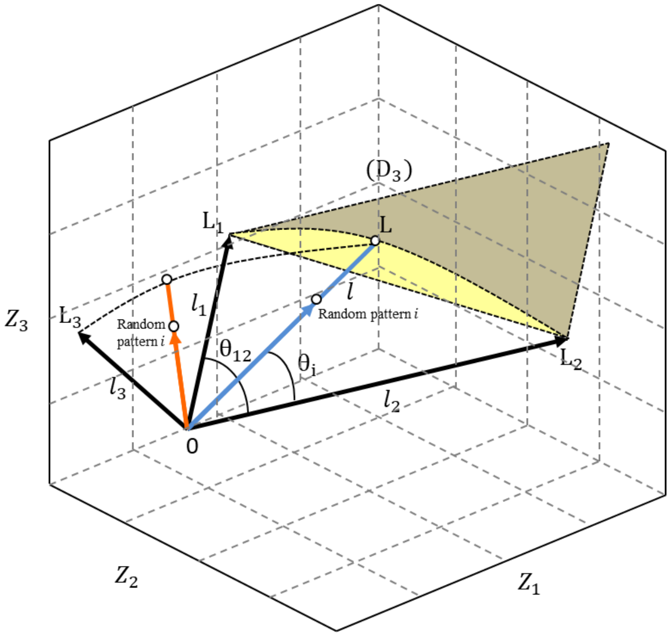


Figure 4.20 Three boundaries for multiple classes.

In order to determine the effect of the new boundary, the chapter analyzes the three boundaries described before. They are named as: “straight line” for the triangle area, “no boundary” for the parallelogram area and “Archimedean” for the new boundary. For all these boundaries, the corresponding classifications are constructed and the four mentioned techniques are applied

4.5.4 Single and multiple fault classification

Classification 12 works with thirteen classes formed with seven multiple classes from Classification 8 and six single classes from Classification 3. The next subsection presents different boundaries used for multiple fault classifications.

4.6 TECHNIQUE EVALUATION RESULTS

The probability of correct diagnosis (diagnosis accuracy indicator) is used as a criterion to evaluate the performance of each technique in gas turbine fault recognition. Four comparative studies are considered. They are formed by varying:

1. Pattern numbers
2. Operating modes
3. Fault boundaries
4. Deviation noise schemes

Within each study, in addition to the varying factor, the fault classification changes as well. The variation of the conditions allows drawing solid conclusions about the best technique. The studies are shortly described below.

4.6.1 Different pattern numbers

The accuracy of fault classes' description depends on the number of simulated patterns; nevertheless, sometimes it is not possible to obtain sufficient data to achieve it [141]. In order to address this hypothetical lack of information and analyze its effect on the diagnosis accuracy for each technique, ten pattern numbers are analyzed going from 100 to 1000 using four classification variations. Initially, calculations are based on 1 seed, which is a parameter for initiating a random number series (one calculation of $\overline{P}$). However, during experimentation, different seeds yield different probabilities of correct diagnosis. To reduce the error produced by this randomness, calculations with 100 seeds are performed and averaged. Figure 4.21 shows an example of comparison between the results obtained from 1 and 100 seeds for Classification 3 and two techniques. It is visible for MLP that for low numbers of patterns the difference between the two types of seed calculation is greater showing the inexactitude for 1-seed computation. For this reason, 1 seed calculations are only preliminary and are used to find the technique optimal parameters whereas 100 seed calculations are used in all comparison cases to draw more reliable results.

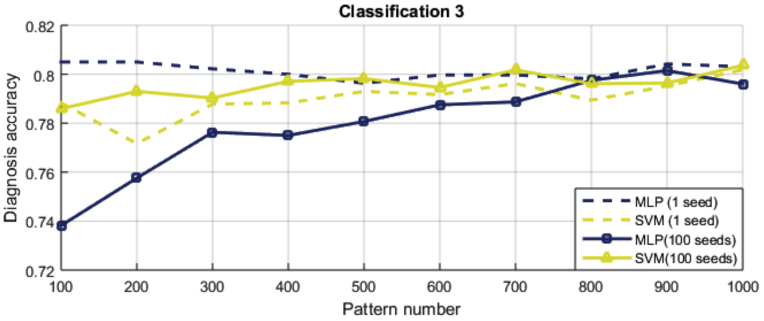


Figure 4.21 Diagnosis accuracy for 1 and 100 seeds.

Figure 4.22 and Table 4.5 show the results obtained for 100 seeds working with four classifications. The first impression is that RBN is the best technique in Classification 3; SVM in Classification 4; MLP and RBN in Classification 9; and MLP and SVM in Classification 10. Since this is not sufficient to select the best technique, total average probabilities considering all pattern numbers and all classification variations are obtained to know the overall fault recognition performance of each technique and shown in Table 4.6. The probabilities are: 0.7922 for MLP, 0.8039 for RBN, 0.8001 for PNN and 0.8122 for SVM. As can be seen, the ANN techniques have very similar diagnosis probabilities (a difference of 1.17% between them); however, SVM is slightly better than all of them (2% over MLP, 0.83 % over RBN and 1.21 % over PNN). Also, total average probabilities for only 100 patterns and all classifications are obtained for all the techniques (Table 4.6). The results are: 0.7612 for MLP, 0.7883 for RBN, 0.7739 for PNN and 0.7893 for SVM. Again, SVM obtained slightly better probabilities (2.81% over MLP, 0.1 % over RBN and 1.54 % over PNN). However, it is visible that the difference between SVMs and RBN is negligible. This is important because SVMs are generally claimed to have better generalization than ANNs when working with small samples. As a final remark, the increase of the pattern number influenced positively the resolution capability for all techniques (up to 9%). However, a drawback is that more execution time and computer memory are required. This is very notorious in parameter tuning stage because a lot of computations are performed before selecting the most appropriate model giving us the highest probability $\overline{P}$. For that reason, the pattern number is a compromise between diagnosis accuracy and computer requirements.

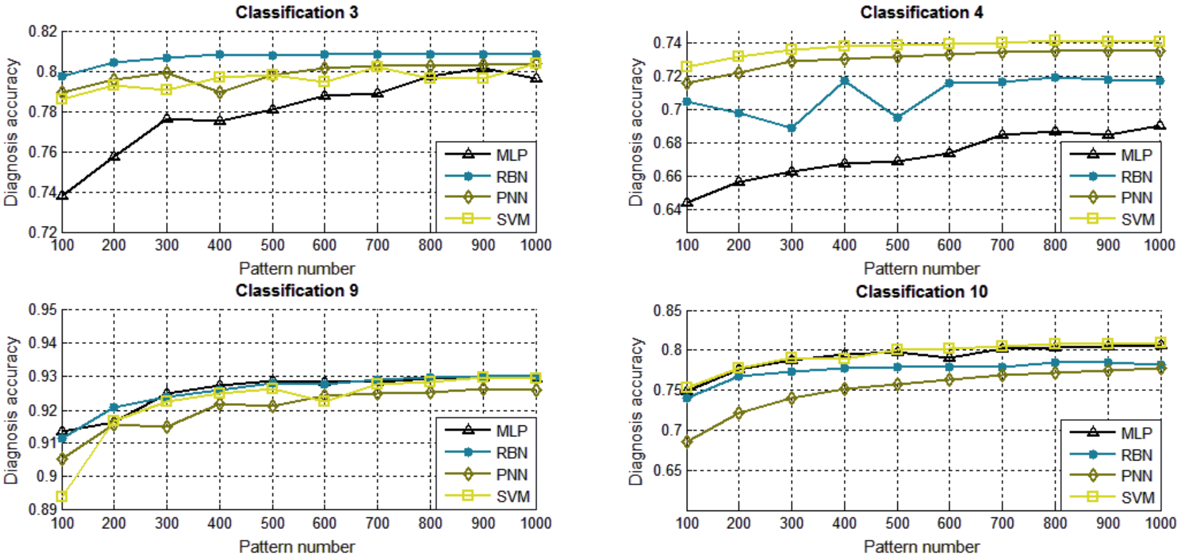


Figure 4.22 Diagnosis accuracy comparison between ANNs and SVMs for different pattern numbers (100 seeds).

Table 4.5 Diagnosis accuracy $\overline{P}$ for different pattern numbers (100 seeds).

Fault Classif.	Method	Number of patterns										Average
		100	200	300	400	500	600	700	800	900	1000	
C3	MLP	0.7382	0.7576	0.7763	0.7750	0.7807	0.7875	0.7887	0.7975	0.8015	0.7960	0.7799
	RBN	0.7973	0.8040	0.8068	0.8080	0.8076	0.8084	0.8083	0.8081	0.8081	0.8081	0.8065
	PNN	0.7897	0.7956	0.7991	0.7893	0.7978	0.8017	0.8024	0.8026	0.8034	0.8036	0.7985
	SVM	0.7859	0.7930	0.7903	0.7971	0.7982	0.7946	0.8018	0.7961	0.7964	0.8036	0.7957
C4	MLP	0.6446	0.6569	0.6630	0.6677	0.6692	0.6738	0.6850	0.6871	0.6848	0.6906	0.6722
	RBN	0.7049	0.6982	0.6888	0.7173	0.6954	0.7160	0.7161	0.7191	0.7179	0.7174	0.7091
	PNN	0.7154	0.7217	0.7289	0.7303	0.7316	0.7326	0.7339	0.7352	0.7350	0.7352	0.7300
	SVM	0.7250	0.7312	0.7359	0.7374	0.7381	0.7390	0.7398	0.7409	0.7403	0.7405	0.7368
C9	MLP	0.9135	0.9164	0.9249	0.9273	0.9287	0.9284	0.9284	0.9294	0.9297	0.9296	0.9256
	RBN	0.9117	0.9210	0.9240	0.9262	0.9280	0.9278	0.9290	0.9297	0.9300	0.9301	0.9258
	PNN	0.9054	0.9158	0.9150	0.9219	0.9213	0.9242	0.9251	0.9255	0.9265	0.9260	0.9207
	SVM	0.8939	0.9166	0.9225	0.9250	0.9263	0.9225	0.9279	0.9283	0.9297	0.9294	0.9222
C10	MLP	0.7486	0.7758	0.7874	0.7941	0.7978	0.7898	0.8022	0.8027	0.8048	0.8062	0.7909
	RBN	0.7395	0.7668	0.7735	0.7772	0.7786	0.7784	0.7790	0.7850	0.7845	0.7812	0.7744
	PNN	0.6852	0.7217	0.7398	0.7513	0.7569	0.7637	0.7688	0.7717	0.7753	0.7775	0.7512
	SVM	0.7526	0.7772	0.7904	0.7887	0.8002	0.8020	0.8049	0.8071	0.8084	0.8091	0.7941

Table 4.6 Overall fault recognition of the techniques.

Method	All pattern numbers & all classifications	100 patterns & all classifications
MLP	0.7922	0.7612
RBN	0.8039	0.7883
PNN	0.8001	0.7739
SVM	0.8122	0.7893

4.6.2 Different operating modes

Two gas turbine operating modes, Mode 1 and Mode 2, are studied. They are close to engine maximal and idle regimes and are set by different high pressure rotor speeds under standard atmospheric conditions. The analysis considers all the classification variations. Based on the above results for pattern numbers, this comparative study only works with 1000 patterns to have more accurate results. The results obtained are presented in Figure 4.23, Figure 4.24 and Table 4.7. Considering both modes, SVM is slightly better with a total average probability of 0.8146. RBN is the second best technique (being the winner in some classifications) with 0.8099. PNN is in the third position with 0.8065 and MLP is the last technique with 0.8038 of performance. Nevertheless, the difference between all the techniques is not so great (1.08%). Another important observation is that for Mode 2, the probabilities are lower than Mode 1 for most of classifications. However, the averaged difference between both modes is small (about 0.0088). Besides, the probability behavior of the techniques is almost the same for the two modes through all classifications. Taking into account that the random errors in the stochastic simulation remain small due to the 100 seeds calculation, the results presented can be more reliable. Thus, we can conclude that the change of operating mode of the analyzed gas turbine does not affect the performance of techniques.

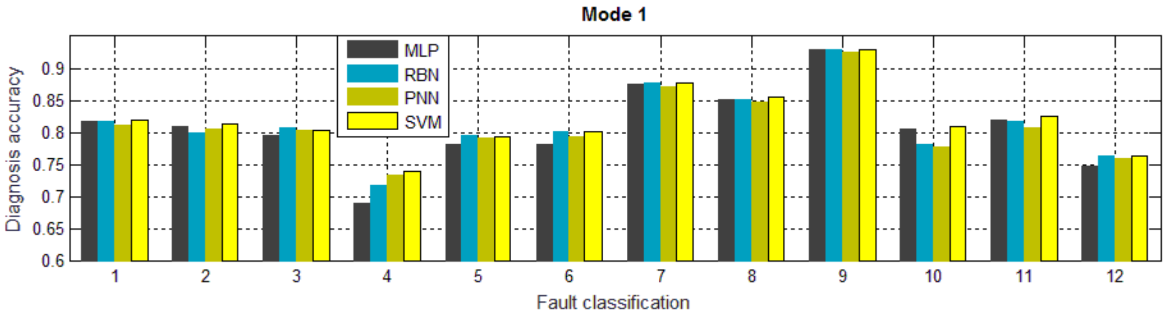


Figure 4.23 Diagnosis accuracy comparison between ANNs and SVMs for operating mode 1 (100 seeds).

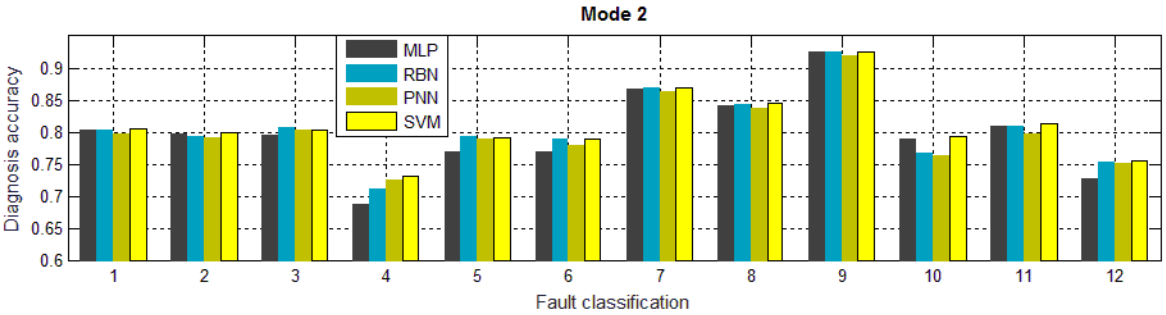


Figure 4.24 Diagnosis accuracy comparison between ANNs and SVMs for operating mode 2 (100 seeds).

4.6.3 Different fault boundaries

Three boundary options are examined: no boundary (parallelogram area), straight line (triangle area), and Archimedean spiral. They are applied to multiple faults of classification variations 7 and 11. For each boundary and variation, the four techniques are used by turn for computing diagnosis probabilities $\overline{P}$ considering 100 seeds. Figure 4.25, Figure 4.26 and Table 4.8 contain all the results that help draw the following conclusions. First, the total average probability for each technique is 0.8229 for MLP, 0.8222 for RBN, 0.8146 for PNN and 0.8248 for SVM. It is evident that the highest value is produced by SVM and the lowest one by PNN. However, the difference between the four recognition techniques remains small (1.02%). Second, the new boundary results in a visible change of the probability $\overline{P}$. This change can be greater (up to 25%) for particular cases, for example, the “Straight line” boundary in Classification 11 where probabilities are very low. Third, for all cases, the “Archimedean spiral” probability occupies an intermediate position between “No boundary” probability and “Straight line” probability. This is easily explained by the fact that the Archimedean spiral curve is situated between the straight line and the parallelogram sides.

Table 4.7 Diagnosis accuracy $\overline{P}$ for two operating modes (100 seeds).

Fault Classification	**Mode 1**				**Mode 2**			
	MLP	**RBN**	**PNN**	**SVM**	**MLP**	**RBN**	**PNN**	**SVM**
C1	0.8172	0.8173	0.8115	0.8190	0.8044	0.8047	0.7983	0.8064
C2	0.8100	0.8007	0.8049	0.8117	0.7974	0.7946	0.7923	0.7994
C3	0.7960	0.8081	0.8036	0.8036	0.7947	0.8079	0.8047	0.8042
C4	0.6906	0.7174	0.7352	0.7405	0.6876	0.7133	0.7254	0.7320
C5	0.7813	0.7966	0.7921	0.7936	0.7702	0.7930	0.7892	0.7913
C6	0.7818	0.8016	0.7942	0.8017	0.7702	0.7894	0.7808	0.7892
C7	0.8756	0.8770	0.8720	0.8770	0.8684	0.8697	0.8635	0.8698
C8	0.8507	0.8525	0.8474	0.8528	0.8420	0.8435	0.8378	0.8447
C9	0.9296	0.9301	0.9260	0.9294	0.9248	0.9248	0.9185	0.9248
C10	0.8062	0.7812	0.7775	0.8091	0.7897	0.7689	0.7635	0.7934
C11	0.8193	0.8186	0.8076	0.8248	0.8090	0.8094	0.7986	0.8143
C12	0.7482	0.7637	0.7607	0.7640	0.7273	0.7550	0.7518	0.7553
Average	0.8089	0.8137	0.8111	0.8189	0.7988	0.8061	0.8020	0.8104
Total Average	**MLP=0.8038; RBN=0.8099; PNN=0.8065; SVM=0.8146**							

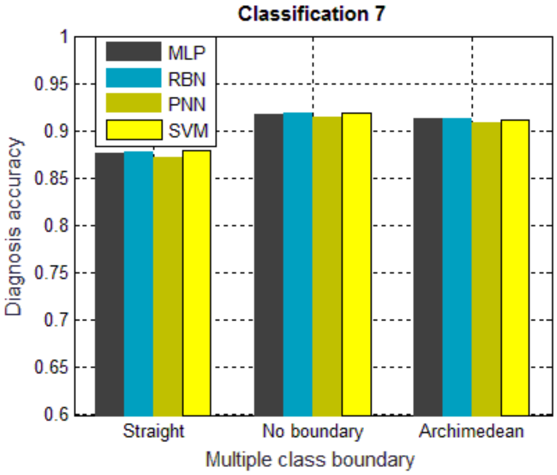


Figure 4.25 Diagnosis accuracy for different boundaries (classification 7).

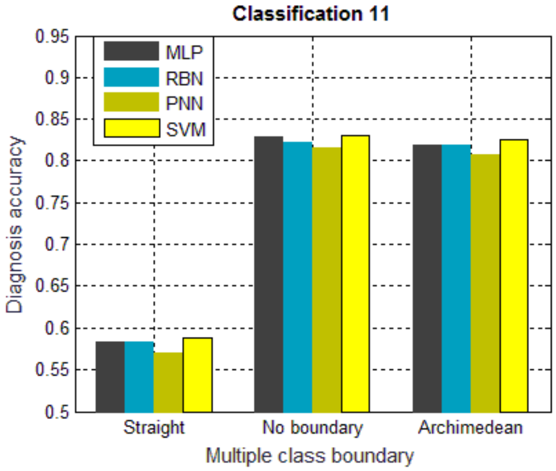


Figure 4.26 Diagnosis accuracy for different boundaries (classification 11).

Table 4.8 Diagnosis accuracy $\overline{P}$ for different boundaries (100 seeds).

Fault Classif.	**Multiple-class Boundary**	**MLP**	**RBN**	**PNN**	**SVM**
C7	Straight line	0.8756	0.8770	0.8720	0.8770
	No boundary	0.9174	0.9181	0.9140	0.9182
	Archimedean	0.9126	0.9130	0.9078	0.9110
C11	Straight line	0.5834	0.5839	0.5703	0.5875
	No boundary	0.8289	0.8223	0.8156	0.8301
	Archimedean	0.8193	0.8186	0.8076	0.8248
	Total average	**0.8229**	**0.8222**	**0.8146**	**0.8248**

4.6.4 Different deviation noise schemes

Two schemes of deviation noise are studied: simulated and real noise. The real noise was extracted from deviations using real data recorded hourly at steady state operating points according to Subsection 4.3. The following elements are considered for the comparison: all the fault classifications, maximal operating mode and 1000 patterns. Table 4.9 shows the results for both error schemes. Figure 4.27 presents the results for real deviation errors. The results for the simulated scheme are the same as Figure 4.23 shown before. Comparing both error representations, one can see a significant increase of diagnosis accuracy for all the techniques (4.66% for MLP, 1.33% for RBN, 2.87% for PNN, and 4.10 % for SVM). The total average probability of each technique is obtained as before by averaging both error schemes. The results are 0.8322 for MLP, 0.8203 for RBN, 0.8255 for PNN and 0.8394 for SVM. Once more the highest value is produced by SVM. However, this time RBN is the lowest one and MLP has a much better performance than in the case of simulated noise. As mentioned before, the difference between techniques is not so great for simulated errors (about 1.07%) while for real errors is a little bit greater (about 3.29%). This can be proven by analyzing classification-to-classification probabilities. In classification 2, 4, 6 and 12 there are evident differences between the highest value (SVM) and the lowest one (RBN). This means that the use of more realistic deviation noise representation does affect the performance of techniques. In contrast to simulated errors, where RBN is the second best technique, the use of real noise negatively affects the technique being the one with the lowest probability for that case. Besides, it requires more training time than usual.

Table 4.9 Diagnosis accuracy $\bar{P}$ for simulated and real deviation noise (100 seeds).

Fault Classif.	**Simulated noise**				**Real noise**			
	MLP	**RBN**	**PNN**	**SVM**	**MLP**	**RBN**	**PNN**	**SVM**
C1	0.8172	0.8173	0.8115	0.8190	0.8911	0.8663	0.8755	0.8959
C2	0.8100	0.8007	0.8049	0.8117	0.8765	0.8309	0.8694	0.8845
C3	0.7960	0.8081	0.8036	0.8036	0.8637	0.8442	0.8590	0.8498
C4	0.6906	0.7174	0.7352	0.7405	0.7704	0.7185	0.8057	0.8220
C5	0.7813	0.7966	0.7921	0.7936	0.8526	0.8136	0.8409	0.8440
C6	0.7818	0.8016	0.7942	0.8017	0.8603	0.8257	0.8549	0.8709
C7	0.8756	0.8770	0.8720	0.8770	0.9263	0.9225	0.9084	0.9235
C8	0.8507	0.8525	0.8474	0.8528	0.9162	0.8913	0.8973	0.9176
C9	0.9296	0.9301	0.9260	0.9294	0.9405	0.9335	0.9151	0.9307
C10	0.8062	0.7812	0.7775	0.8091	0.7738	0.7410	0.6984	0.7680
C11	0.8193	0.8186	0.8076	0.8248	0.8020	0.7852	0.7578	0.8038
C12	0.7482	0.7637	0.7607	0.7640	0.7923	0.7511	0.7959	0.8077
Average	0.8089	0.8137	0.8111	0.8189	0.8555	0.8270	0.8398	0.8599
Total Average	**MLP=0.8322; RBN=0.8203; PNN=0.8255; SVM=0.8394**							

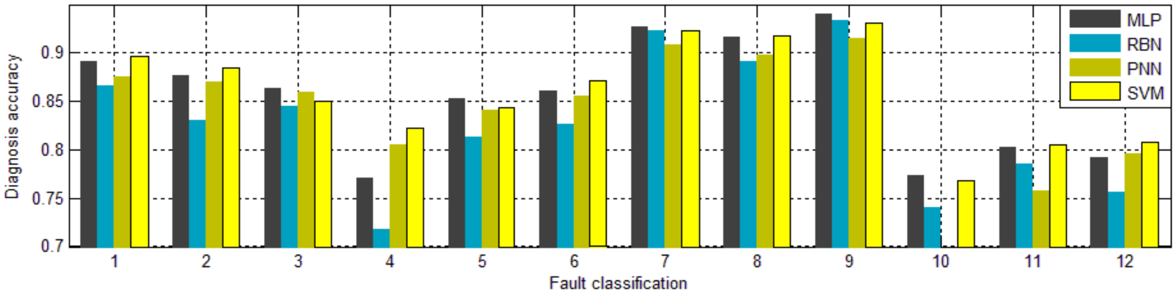


Figure 4.27 Diagnosis accuracy comparison between ANNs and SVMs for real deviation errors (100 seeds).

Chapter 5: Development of a Data-Driven Gas Turbine Diagnostic Algorithm Using the ProDiMES Software

5.1 OVERVIEW

The present chapter introduces and validates a gas turbine diagnostic approach based on data-driven models (polynomials) and recognition techniques (MLP, PNN, and SVM) using the ProDiMES software. The approach unites the diagnostic stages of feature extraction, anomaly detection, and fault identification resulting in an integrated algorithm. A complete benchmarking analysis of the proposed algorithm is made by using the two functionalities of ProDiMES: 1) independent development and evaluation and 2) blind test case side-by-side comparison. For the case of the independent development, engine fleet experimenting no-fault conditions, faults and degradation mechanisms are simulated through ProDiMES. A testing procedure that comprises many steps enables the tuning and comparison of the recognition techniques. The best configuration of the algorithm with the optimized feature extraction, fault detection and fault identification techniques is evaluated using the blind test case and compared with other diagnostic solutions on the basis of the ProDiMES metrics.

Figure 5.1 illustrates in highlighted boxes the main characteristics of the algorithm to develop in Chapter 5. The selection includes: a turbofan engine for aviation application; fault diagnosis of gas path, measurement and control systems; one-point (cruise) and multi-point (take-off and cruise) diagnostic analysis; an algorithm that works with feature extraction, monitoring and diagnosis as an integrated approach; the use of physics-based, data-driven and non-linear models; a fault classification based on single and multiple faults; recognition techniques such as MLP, PNN and SVM; and the LSM as a method for identifying baseline and degraded models.

5.2 STRUCTURE OF THE PRODIMES SOFTWARE

5.2.1 Benchmarking process

The software ProDiMES enables the users to independently develop, evaluate and compare proposed aircraft diagnostic solutions. The ProDiMES benchmarking process as shown in Figure 5.2 consists of two major functionalities: 1) independent development and evaluation and 2) blind test case side-by-side comparison. The top half of Figure 5.2 shows the capabilities of ProDiMES that allow users to independently develop and evaluate diagnostic solutions. Engine Fleet Simulator (EFS) generates snapshot measurement histories of an engine fleet over different flight intervals working with take-off and cruise operating points. Through an internal interface, it is possible to specify the total number of simulated engines and the number of flights per engine, the type and number of fault cases, the flight of fault initiation and the rate of fault evolution. The user's diagnostic solutions are used to interpret the engine histories and diagnose possible faults and the metrics enable the evaluation of the proposed diagnostic algorithms. This is achieved by the comparison of the diagnostic assessments and the engine ground-truth information (true fault/no fault condition). Finally, the results obtained from the metrics can be archived in a common format to ease their interpretation. After the independent development of the diagnostic solutions, users can apply their algorithms to a blind test case that is part of the ProDiMES software as shown in the bottom half of Figure 5.2. This test case allows the side-by-side comparison of diagnostic methodologies from other users. All the participants receive the same set of blind test case data but the corresponding ground-truth information is not provided. NASA receives the diagnostic assessment of each user, evaluates the results against the ground-truth, and returns the metrics along with the anonymous results of other researchers.

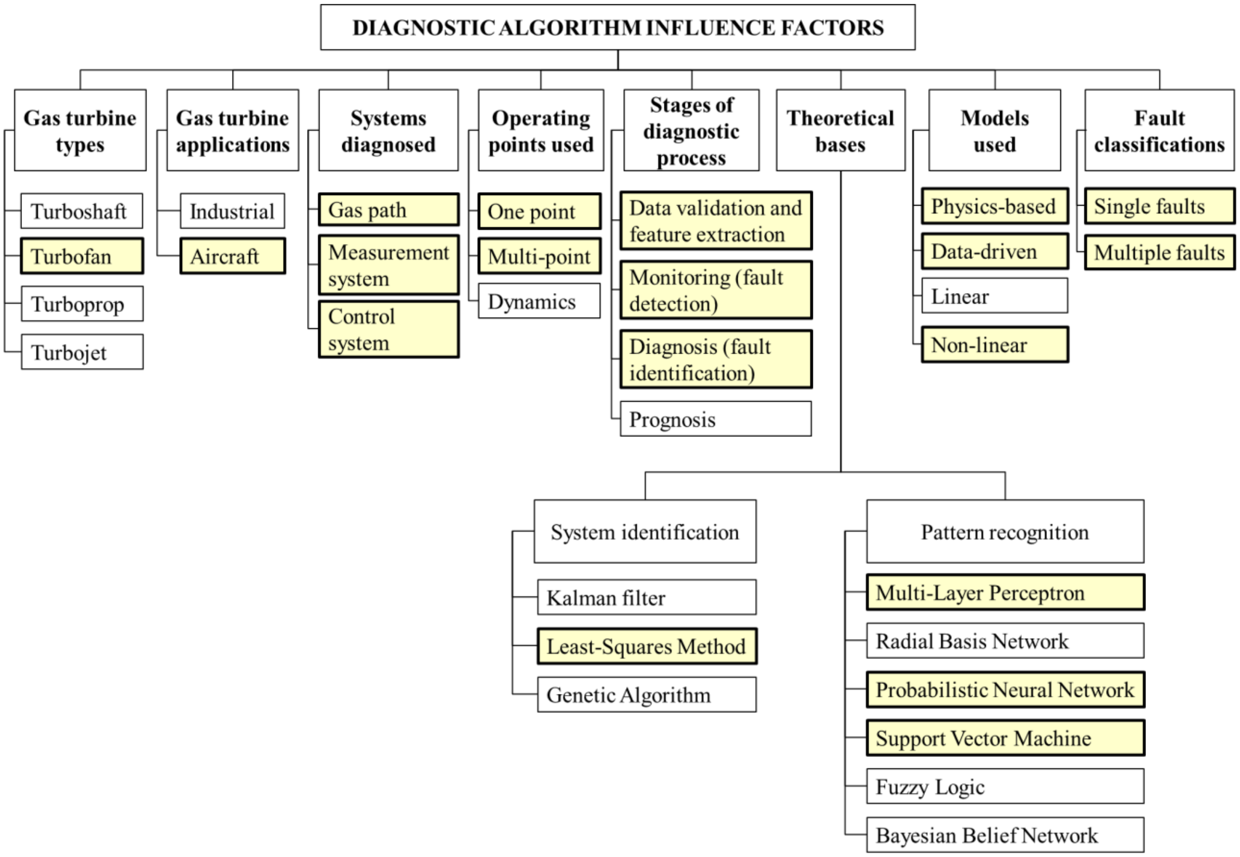


Figure 5.1 Diagnostic algorithm influence factors used throughout Chapter 5.

5.2.2 Engine fleet simulation

ProDiMES includes an engine fleet simulation (EFS) that generates simulated measurement parameter histories for each engine of the fleet. To simulate real engine behavior, it works with a deterioration profile, noise level and operation mode (takeoff and cruise) unique for each engine. The EFS is implemented in Matlab and consists of a graphical user interface (GUI), a case generator and a C-MAPPS steady state engine model (GT3) as depicted in Figure 5.3.

Graphical user interface (GUI): In the graphical user interface, the user can control the type and the number of faults occurring in the fleet of engines. The maximal number of possible simulated faults is 18 plus a no-fault case.

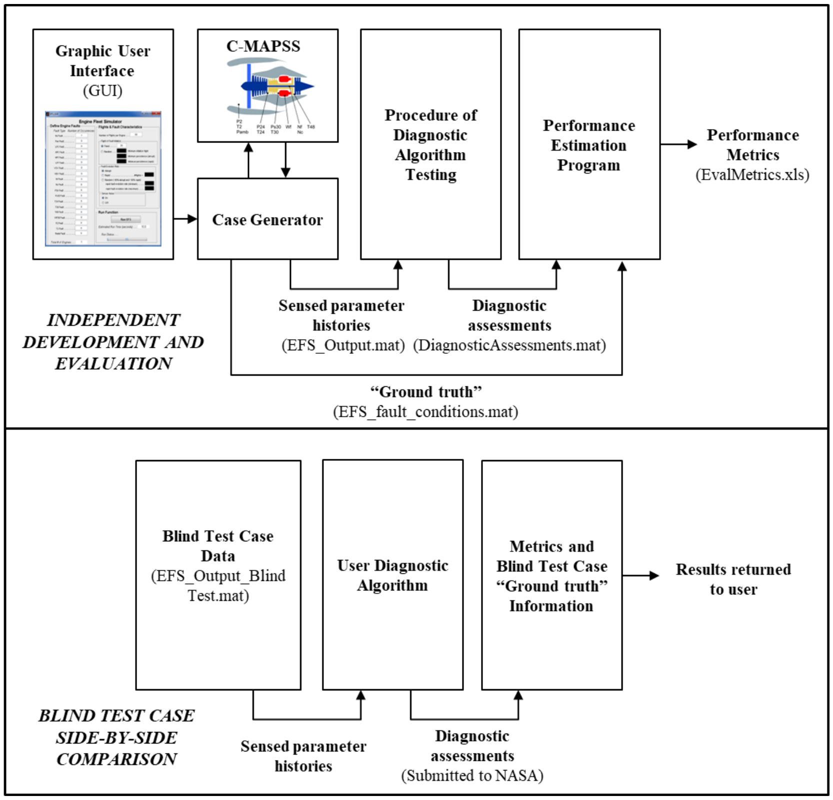

Graphic User Interface (GUI)
C-MAPSS
P2
T2
Pamb
P24
T24
Ps30
T30
Wf
Nf
Nc
T48
Case Generator
Procedure of Diagnostic Algorithm Testing
Performance Estimation Program
Performance Metrics (EvalMetrics.xls)
Sensed parameter histories (EFS_Output.mat)
Diagnostic assessments (DiagnosticAssessments.mat)
INDEPENDENT DEVELOPMENT AND EVALUATION
"Ground truth" (EFS_fault_conditions.mat)
Blind Test Case Data (EFS_Output_Blind Test.mat)
User Diagnostic Algorithm
Metrics and Blind Test Case "Ground truth" Information
Results returned to user
BLIND TEST CASE SIDE-BY-SIDE COMPARISON
Sensed parameter histories
Diagnostic assessments (Submitted to NASA)

Figure 5.2. ProDiMES benchmarking process (adapted from [72]).

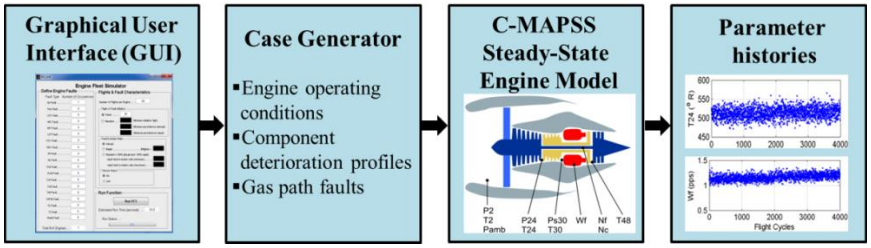


Figure 5.3. Structure of EFS.

Table 5.1 shows these faults and their magnitudes, within which the faults are distributed uniformly. Module faults (ID 1-5) corresponding to Fan, LPC, HPC, HPT and LPT are simulated by adjusting at the same time efficiency η and flow capacity γ parameters. Actuator faults (ID 6-7) corresponding to VSV and VBV result from a mis-scheduling between the commanded and current actuator position. There are also 11 different sensor faults (ID 8-18) whose magnitudes are in units of average measurement noise standard deviation σ. It is important to mention that EFS does not work with multiple faults. Instead, each individual engine only experiences a single fault type. The number of engines in the fleet is determined by the sum of the number of occurrences for each fault. Other aspects to consider are: the number of flights to generate the output data (the maximal number is 5000 flights per engine); the fault evolution type (abrupt or rapid); the flight of fault initiation (the 11th flight is the lowest value) and the sensor noise (on or off). The EFS works with eleven sensed variables. The seven measured variables shown in Table 5.2 are available for monitoring and are represented by a vector $\vec{Y}$. Table 5.3 shows four measured variables used as operating conditions represented by a vector $\vec{U}$. All the variables correspond to an engine standard measurement system.

Table 5.1. ProDiMES simulated faults.

ID	Fault description	Fault Magnitude
0	No-fault	---
1	Fan fault	1 to 7%
2	LPC fault	1 to 7%
3	HPC fault	1 to 7%
4	HPT fault	1 to 7%
5	LPT fault	1 to 7%
6	VSV fault	1 to 7%
7	VBV fault	1 to 19%
8	Nf sensor fault	± 1 to 10 σ
9	Nc sensor fault	± 1 to 10 σ
10	P24 sensor fault	± 1 to 10 σ
11	Ps30 sensor fault	± 1 to 10 σ
12	T24 sensor fault	± 1 to 10 σ
13	T30 sensor fault	± 1 to 10 σ
14	T48 sensor fault	± 1 to 10 σ
15	Wf sensor fault	± 1 to 10 σ
16	P2 sensor fault	± 1 to 10 σ
17	T2 sensor fault	± 1 to 10 σ
18	Pamb sensor fault	± 1 to 19 σ

*LPC=Low Pressure Compressor, HPC=High Pressure Compressor, HPT=High Pressure Turbine, LPT=Low Pressure Turbine, VSV= Variable Stator Vane, VBV=Variable Bleed Valve.

Table 5.2. ProDiMES monitored variables.

ID	Variable	Symbol
1	Physical core speed	Nc
2	Total pressure at LPC outlet	P24
3	Static pressure at HPC outlet	Ps30
4	Total temperature at LPC outlet	T24
5	Total temperature at HPC outlet	T30
6	Total temperature at HPT outlet	T48
7	Fuel flow	Wf

Table 5.3 ProDiMES operating conditions.

ID	Variable	Symbol
1	Physical fan speed	Nf
2	Total pressure at fan inlet	P2
3	Total temperature at fan inlet	T2
4	Ambient pressure	Pamb

Case generator: The case generator produces the parameter histories after the user has selected the number and type of faults in the GUI stage. One important characteristic is the random generation of unique faults, degradation profiles and operating history for each engine in the fleet. This also includes the following assignations: the date when the collection of the engine data starts, the city pairs for the takeoffs of the engine, the ambient pressure, the atmospheric temperature, the Mach number and power setting parameters for takeoff and cruise._The level and rate of gradual performance deterioration for each engine are also considered by the case generator. They emulate the degradation that an aircraft engine experiences during its lifetime due to different effects such as fouling, erosion, and corrosion of blades and vanes. The gradual deterioration is not considered a fault and its development is much slower than the produced by faults.

C-MAPSS steady-state engine model: The commercial modular aero-propulsion system simulation (C-MAPSS) steady state is a high-bypass turbofan engine model created for diagnostics research. This model is run inside the EFS and receives the outputs from the case generator to produce the simulated measurement parameter histories for each engine, at takeoff and cruise of each flight. It works with two spool speeds (fan and core speed).

5.2.3 Description of Evaluation Metrics

The ProDiMES software assesses the overall detection and classification performance of candidate diagnostic methods through the following metrics [72]:

1) True Positive Rate: Number of correct fault detections divided by the number of fault cases.
2) False Negative Rate: Number of incorrect no fault detections divided by the number of fault cases.
3) False Positive Rate: Number of incorrect fault detections divided by the number of no fault cases.
4) True Negative Rate: Number of correct no fault detections divided by the number of no fault cases.
5) Correct Classification Rate: Number of correct classifications of a fault divided by the number of cases of that fault.
6) Misclassification Rate: Number of incorrect classifications of a fault divided by the number cases of that fault).
7) Detection latency: The average number of flights a fault must persist prior to true positive detection by the diagnostic algorithm.
8) Classification latency: The average number of flights a fault must persist prior to correct fault classification by the diagnostic algorithm.
9) Kappa Coefficient: Provides a measure of an algorithm's ability to correctly classify a fault, which takes into account the expected number of correct classifications occurring by chance. The Kappa Coefficient, denoted here as κ, is calculated from the elements of the un-normalized confusion matrix, *C*, as shown in the equation below. The two subscript indices represent the row and column corresponding to individual confusion matrix elements.

$$k = \frac{N(\text{correctlyclassified}) - N(\text{expected correct by chance})}{N(\text{total}) - N(\text{expected correct by chance})} \tag{5.1}$$

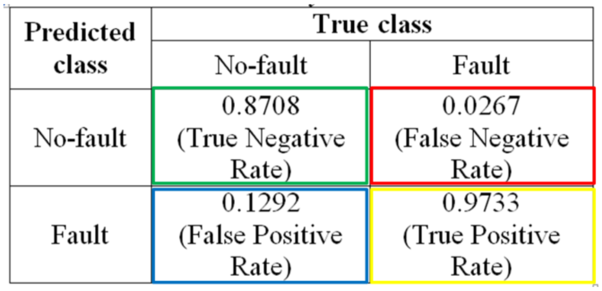
where:

$$N(\text{correctlyclassified}) = \sum_{p=1}^{n} C_{pp}$$

$$N(\text{total}) = \sum_{p=1}^{n}\sum_{q=1}^{n} C_{pq}$$

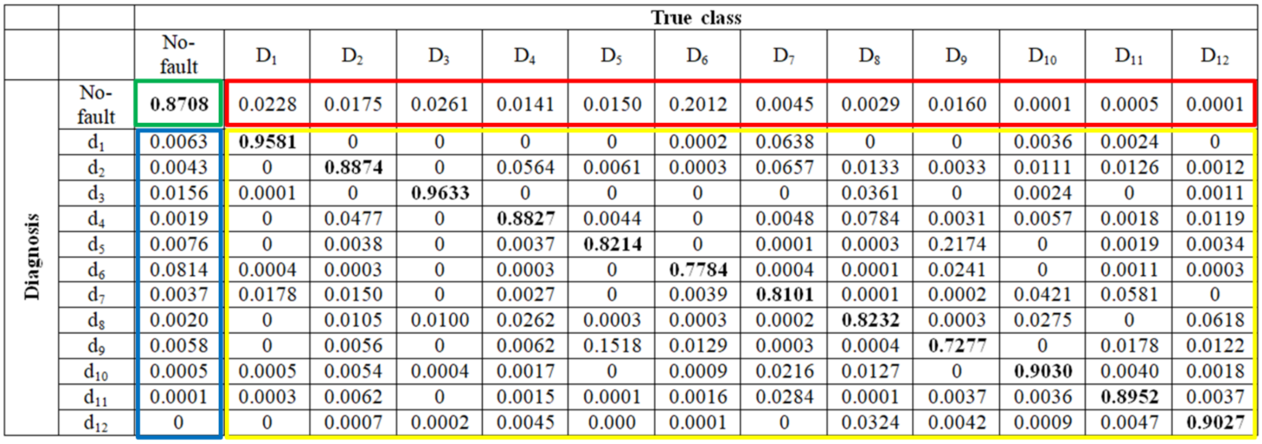
$$N(\text{expected correct by chance}) = \sum_{p=1}^{n}\left\{\sum_{q=1}^{n}\frac{C_{pq}}{N(\text{total})}\sum_{q=1}^{n} C_{qp}\right\}$$

The detection metrics (1 through 4 in the above list) form the Anomaly Detection Matrix (Figure 5.4), and the classification metrics (5 and 6 in the above list) form the Classification Confusion Matrix (Figure 5.5). It is worth noting that the elements of the anomaly detection matrix can be obtained from the confusion matrix as exemplified by each color of cells in Figure 5.4 and Figure 5.5. In this way, the stages of anomaly detection and fault identification are realized together in the same algorithm.



Predicted class	**True class**	
	No-fault	Fault
No-fault	0.8708 (True Negative Rate)	0.0267 (False Negative Rate)
Fault	0.1292 (False Positive Rate)	0.9733 (True Positive Rate)

Figure 5.4. Example of an anomaly detection matrix



		True class												
		No-fault	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}
Diagnosis	No-fault	**0.8708**	0.0228	0.0175	0.0261	0.0141	0.0150	0.2012	0.0045	0.0029	0.0160	0.0001	0.0005	0.0001
	d_1	0.0063	**0.9581**	0	0	0	0	0.0002	0.0638	0	0	0.0036	0.0024	0
	d_2	0.0043	0	**0.8874**	0	0.0564	0.0061	0.0003	0.0657	0.0133	0.0033	0.0111	0.0126	0.0012
	d_3	0.0156	0.0001	0	**0.9633**	0	0	0	0	0.0361	0	0.0024	0	0.0011
	d_4	0.0019	0	0.0477	0	**0.8827**	0.0044	0	0.0048	0.0784	0.0031	0.0057	0.0018	0.0119
	d_5	0.0076	0	0.0038	0	0.0037	**0.8214**	0	0.0001	0.0003	0.2174	0	0.0019	0.0034
	d_6	0.0814	0.0004	0.0003	0	0.0003	0	**0.7784**	0.0004	0.0001	0.0241	0	0.0011	0.0003
	d_7	0.0037	0.0178	0.0150	0	0.0027	0	0.0039	**0.8101**	0.0001	0.0002	0.0421	0.0581	0
	d_8	0.0020	0	0.0105	0.0100	0.0262	0.0003	0.0003	0.0002	**0.8232**	0.0003	0.0275	0	0.0618
	d_9	0.0058	0	0.0056	0	0.0062	0.1518	0.0129	0.0003	0.0004	**0.7277**	0	0.0178	0.0122
	d_{10}	0.0005	0.0005	0.0054	0.0004	0.0017	0	0.0009	0.0216	0.0127	0	**0.9030**	0.0040	0.0018
	d_{11}	0.0001	0.0003	0.0062	0	0.0015	0.0001	0.0016	0.0284	0.0001	0.0037	0.0036	**0.8952**	0.0037
	d_{12}	0	0	0.0007	0.0002	0.0045	0.000	0.0001	0	0.0324	0.0042	0.0009	0.0047	**0.9027**

Figure 5.5. Example of a Classification confusion matrix

5.3 TESTING PROCEDURE

As described before, the measurements are generated for cruise and takeoff steady state operating points of each flight. The case generator saves the measurements (snapshots) of an entire engine fleet in the file *EFS_Output.mat* (see Figure 5.2) while a fault condition history ("ground truth") is recorded in the file *EFS_fault_conditions.mat*. In this subsection we use the cruise data for tuning and testing the proposed algorithm. Then, as required by the ProDiMES methodology, the algorithm is adapted to multipoint operation on cruise and takeoff measurements, and the blind test metrics are obtained for the adapted algorithm and compared with the metrics of other known diagnostic developments. As can be seen in Figure 5.2 for the independent development, the snapshot and the ground truth information presents input data for the testing procedure of the proposed diagnostic algorithm. The algorithm embraces determination of a baseline model, computation of measurement deviations using this model, fault classification formation from deviation vectors, and application of a pattern recognition technique for fault diagnosis. Within a testing procedure, the algorithm components are optimized and final diagnostic assessments made by the algorithm are written in the file *DiagnosticAssessments.mat*. This file and the file *EFS_fault_conditions.mat* are then directed to the ProDiMES program that computes performance metrics for the algorithm under analysis.

The testing procedure needs to be explained in more detail. Let us use the diagram drawn in Figure 5.6 to gain better understanding of the procedure operation and interaction with the benchmarking platform. The procedure employs the data of three cases of engine fleet simulation by ProDiMES. Case 1 results in the no-fault data for baseline model determination and verification (file *EFS_Output_1.mat*). Case 2 data (files *EFS_Output_2.mat* and *EFS_fault_conditions_2.mat*) correspond to healthy and all faulty engine conditions and are used to form the classification for fault detection and identification at a learning stage. Case 3 data (*EFS_Output_3.mat* and *EFS_fault_conditions_3.mat*) are used to the same end at a validation stage. As shown in Figure 5.6, the testing procedure is divided into eight main steps described below. To gain high adequacy of the baseline model and accuracy of the deviations, cycles of model adjustment are performed. This means the repetition of steps 1 and 2 for different variation of a reference sample and a verification sample.

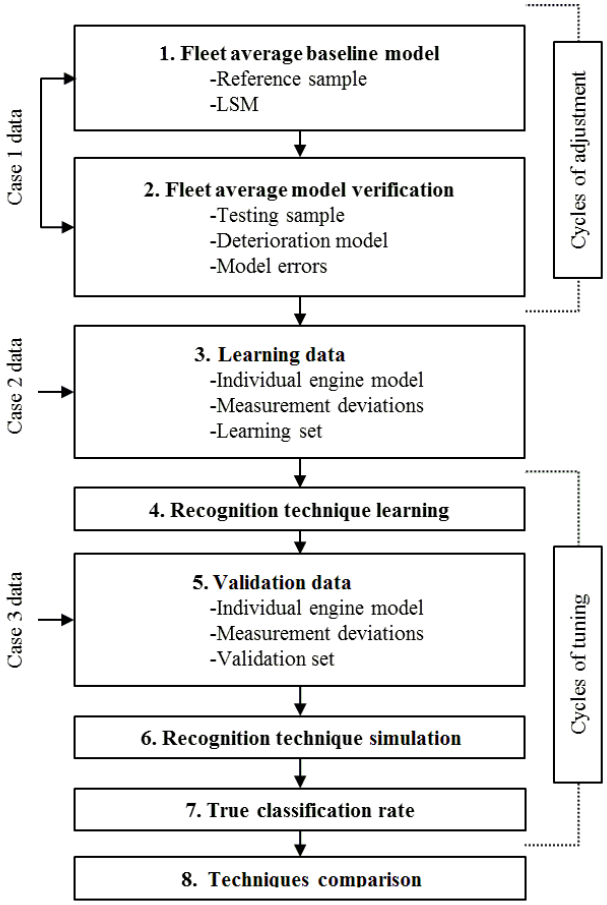


Figure 5.6. Steps of the testing procedure.

Learning and validation data (steps 3 and 5 accordingly) present a common classification that unites all healthy and faulty engine conditions (classes) presented in

Table 5.1. Using this classification (steps 4 and 6), a recognition technique determines the class (healthy or faulty) that is closest to a pattern (deviation vector) to be recognized. In this way, the same technique performs both fault detection and fault identification as one stage. Optimization of a diagnostic process is conducted by cycles of tuning (steps 4-7) in the following sequence. A set of candidate techniques is firstly determined. Each technique is then tailored to the diagnostic problem to solve. Finally, the techniques are compared. The best technique generates diagnostic assessments for the Performance Estimation Program (see Figure 5.2). The resulting performance metrics of the algorithm under analysis are eventually compared with the ProDiMES metrics of other diagnostic

developments. The same input data (case 3 data and blind test data) and the same accuracy criteria (metrics) allow objective benchmarking.

5.3.1 Fleet-average baseline model

This section describes the methodology used and the results obtained in steps 1 and 2 of the testing procedure.

Fleet-average baseline model creation (Step 1)

To extract useful diagnostic information from raw measurements, deviations are computed in the form of $\delta Y_i^* = \frac{Y_i^* - Y_{0i}(\overrightarrow{U})}{Y_{0i}(\overrightarrow{U})}$. Since reliability of fault detection and identification depends on the deviations' quality, deviation errors should be as small as possible. The description of the healthy engine performance (baseline) is of great importance for a correct diagnostics. Considering one monitored gas path variable as function of four operating condition arguments in the vector $\vec{U}$ (Nf , $P2$, $T2$ and P_{amb}), a baseline model through polynomials can be expressed as:

$$\begin{aligned}\hat{Y}_0(\vec{U}) = a_1 + a_2 Nf + a_3 P_2 + a_4 T2 + a_5 P_{amb} + a_6 NfP2 + a_7 NfT2 + a_8 NfP_{amb} + \\ + a_9 P2T2 + a_{10} P2P_{amb} + a_{11} T2P_{amb} + a_{12} Nf^{\,2} + a_{13} P2^2 + a_{14} T2^2 + a_{15} P_{amb}^2\end{aligned} \tag{5.2}$$

where $a_1,..., a_{15}$ are the model coefficients calculated using the LSM for all monitored variables.

Accuracy of the baseline model significantly depends on a reference sample used to determine its coefficients, and it is a challenging issue to compose a proper sample. The point is that, on the one hand, the reference sample must be large to satisfy approximation accuracy, on the other hand, the number of flights to collect data is limited to avoid a negative influence of engine deterioration.

Three variations of the sample were created for cruise conditions and preliminary considered in [142]. Table 5.4 specifies their characteristics. To form these samples, the file *EFS_Output_1.mat* has been generated by the Case Generator for 300 no faulty engines. To minimize the influence of engine deterioration, the flights that correspond to the beginning of maintenance are included in samples 1 and 2. To exclude the deterioration effect in sample 3, a polynomial function for a deteriorated engine was firstly created, and the necessary baseline model was extracted then from this function. As a result of comparison, it was found that the model based on sample 3 had low accuracy while the models created with samples 1 and 2 were equally accurate. As sample 1 has a by far lower dimension, the model based on its data has been chosen for further analysis.

Table 5.4 Baseline model variations

Reference sample variation	**Number of engines**	**Flights per engine**
1	100	90
2	300	270
3	1	5000

Let us analyze the accuracy of the selected baseline model through the quality of the deviations calculated with this model. To clearly evaluate approximation capability of polynomials, the model was firstly determined on reference sample data generated by ProDiMES without noise. Figure 5.7

illustrates the deviations obtained on these data. The deviations of three monitored variables (P24 - total pressure at LPC outlet, Ps30 - static pressure at HPC outlet, and T24 - total temperature at LPC outlet) are plotted here against the first 500 flights of the sample. We can see that for all of the variables the random deviation variations (approximation noise) are quite small, just about ±0.0004. Thus, polynomial approximation accuracy is quite high. As a result, one can clearly distinguish the periods of slight gradual engine degradation during 90 flights of each engine and small shifts due to engine-to-engine differences.

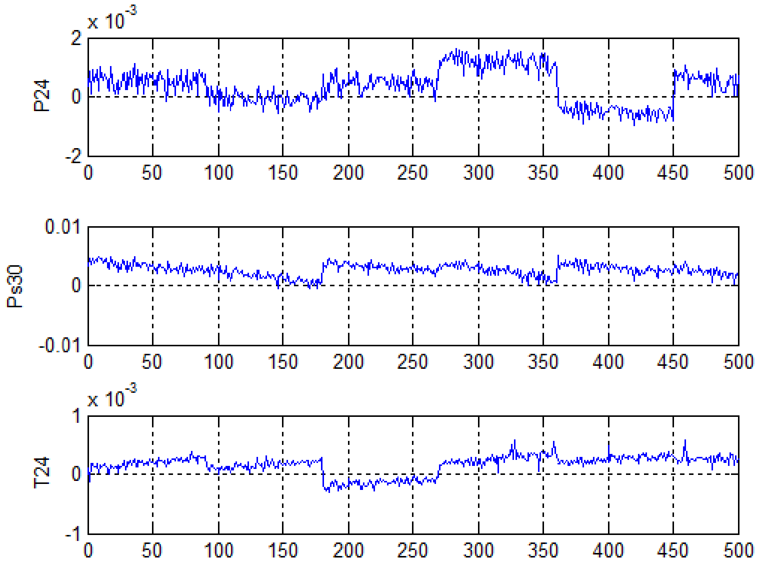


Figure 5.7. Reference sample deviations (noise-free Sample 1)

Fleet-average baseline model verification (Step 2)

Since the reference model is intended to diagnose the engine for the entire life of the engine, all 5000 available flights were used to form a test sample. As an example, deviations of the same variables are plotted in Figure 5.8 against 5000 successive flights of an engine, including the first 90 flights of the reference sample. The red lines here present the approximation of the noisy deviations (blue line) by fourth-order polynomial functions of a flight number. Therefore, the reading lines show a systematic influence of the deterioration, and the differences between the deviations and the approach line are considered as deviation noise. In Figure 5.8 it can be seen that, for at least two variables, the initially small noise for the reference flights grows gradually along with the increase in the deviations due to the deterioration of the engine. The explanation is that, for a deteriorated engine, the influence of the operating conditions on the monitored variables differs from the reference model, that is, the model becomes less accurate. This effect should be taken into account in a diagnostic algorithm.

The next issue to analyze is measurement noise. It was firstly included in the reference set. As a result, the baseline model became less accurate and the noise in testing deviations doubled. Then the errors were also added to the testing sample. As seen in Figure 5.9 that illustrates the resulting testing deviations, the noise has a fixed level and is so significant that the varying approximation inaccuracy observable in Figure 5.8 is now completely hidden. The noise was found to be 10 times larger with respect to the no-noise case illustrated by Figure 5.8. In this way, measurement errors in current (testing) data are the primary cause of the deviation noise. The influence of measurement errors of the data used to determine the baseline model is by far smaller, and inadequacy of polynomials presents the smallest error component.

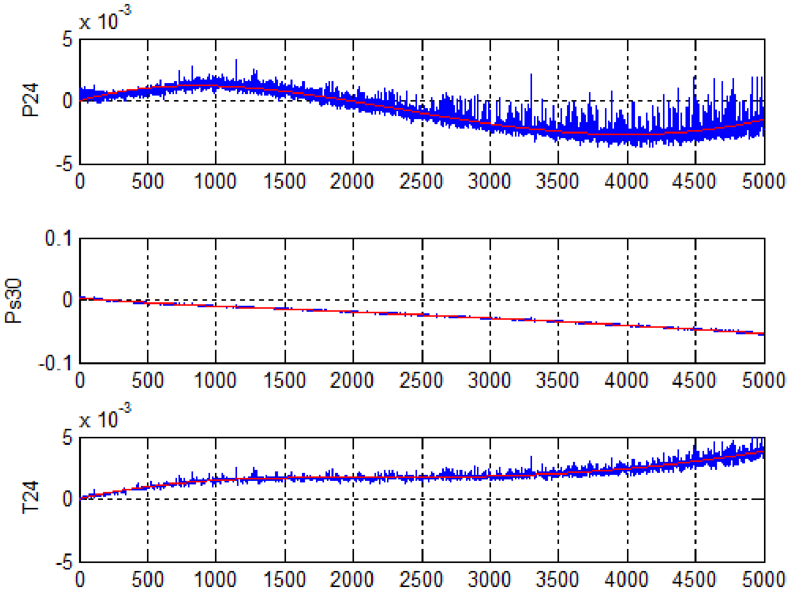


Figure 5.8. Deviations for noise-free testing data

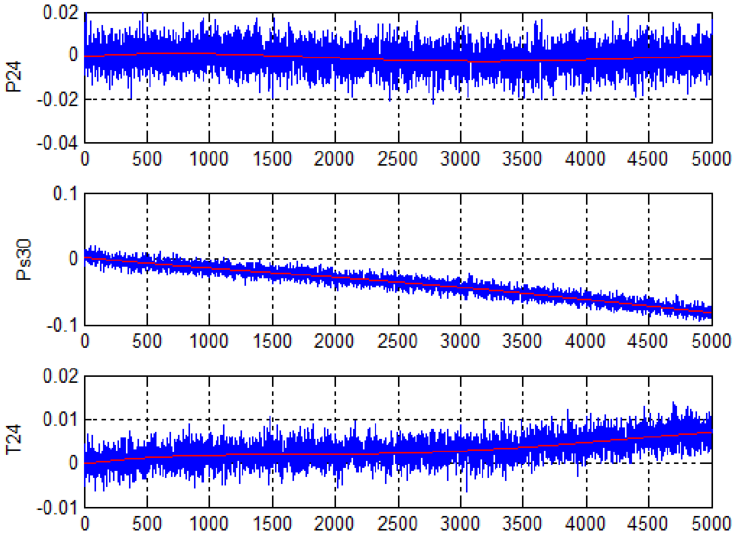


Figure 5.9. Deviations for testing data with measurement noise

5.3.2 Pattern-based fault classification

Having determined a baseline model, we are able now to compute deviations of monitored variables for different healthy and faulty conditions. The sets of deviation vectors (patterns) are employed for recognition techniques' learning and validation.

Learning data (Step 3)

The learning data are computed according to the below procedure using the files *EFS_Output_2.mat* and *EFS_fault_conditions_2.mat* as input information. Engines simulated by ProDiMES have an individual initial level of performances and an individual degradation severity. Thus, the deviations directly computed through the fleet average baseline model will be affected by the engine individuality and degradation effects in addition to the fault influence. To exclude these effects and construct the fault classification from the patterns affected only by the faults, a simple correction is used. Since in ProDiMES the data generated for the first 10 flights of each engine are free of fault influence, it is proposed to use these data for computing an average correction coefficient.

$$K_i = \frac{1}{10}\sum_{j=1}^{10}\frac{Y_{ij}^*}{Y_{oi}(\vec{U}_j)} \tag{5.3}$$

for each monitored variable Y_i. Then, for any flight, an individual baseline value

$$YI_{0i} = K_i Y_{0i}(\vec{U}) \tag{5.4}$$

is computed and then used to determine a deviation of the corresponding monitored variable

$$Z_i = \frac{Y_i^* - YI_{0i}}{a_{Yi} YI_{0i}} \tag{5.5}$$

As operating conditions and monitored variables simulated by ProDiMES include measurement noise, the deviation induced by engine deterioration and faults also has random errors. A coefficient a_{Yi} normalizes these errors, and, as a result, the deviations of all monitored variables have the same error span (-1, 1). These deviations form a diagnostic space and constitute a vector $\vec{Z}$ that is a pattern to be recognized. To have a representative fault classification, the files *EFS_Output_2.mat* and *EFS_fault_conditions_2.mat* were obtained for no-fault and all 18 faulty engine health conditions available in ProDiMES. Using these files, the 19 corresponding pattern-based classes are created from the patterns. To this end, the measurements from many engines and flights corresponding to each fault condition are transformed in a sample of patterns that is a representation of the corresponding class. In this way, a fault classification is a totality of these samples. It is used to learn a fault recognition technique and is called a learning set $\mathbf{Z_L}$.

The following figures exemplify the fault class formation using patterns of $\mathbf{Z_L}$ in the space deviations: Figure 5.10 shows class 1 (no fault) and class 4 (HPC fault); Figure 5.11 shows faults 2-8 including component and actuator faults; Figure 5.12 shows sensor faults 10-14.

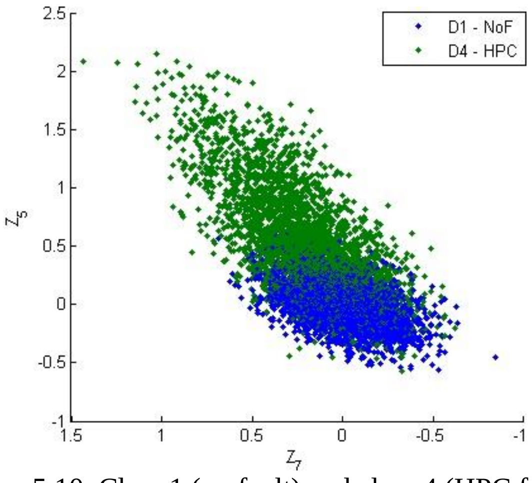


Figure 5.10. Class 1 (no fault) and class 4 (HPC fault)

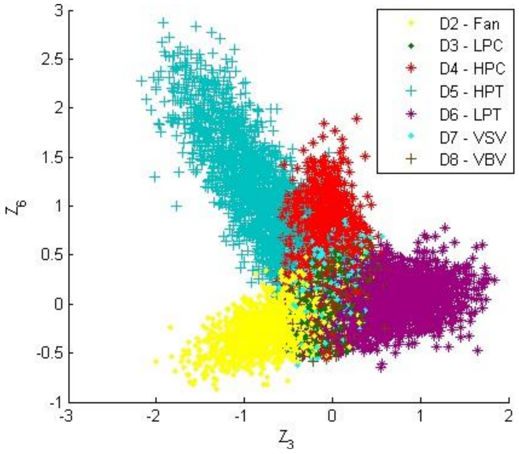


Figure 5.11 Component and actuator faults

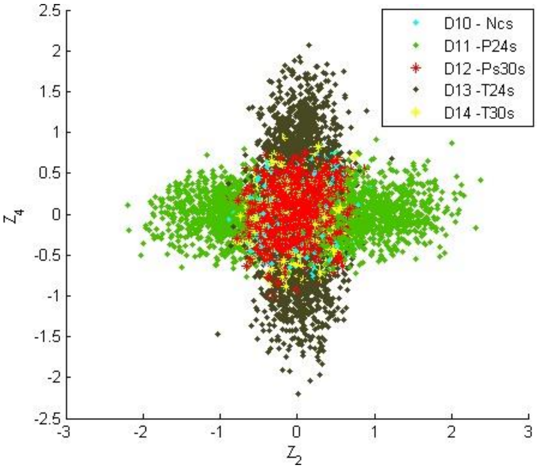


Figure 5.12 ProDiMES sensor faults.

Recognition technique learning (Step 4)

In our previous studies SVM were found to be the most accurate diagnostic technique for an industrial power plant. However, the differences between the competing techniques were small, and to diagnose an aircraft engine within this study, two neural networks, MLP and PNN, have additionally been chosen. To create, learn, and simulate three techniques, the Matlab Neural Network Toolbox functions are used. Although each technique learned on the data of the set $\mathbf{Z_L}$ is ready to diagnose gas turbine faults, it should be examined on new data that allow to avoid overlearning and correctly estimate the techniques accuracy. The new data set is called a validation set $\mathbf{Z_V}$.

Validation data (Step 5)

The new set of patterns $\mathbf{Z_V}$ requires new input data, and new files *EFS_Output_3.mat* and *EFS_fault_conditions_3.mat* have been generated by ProDiMES. However, the set $\mathbf{Z_V}$ should have the same pattern distribution that the set $\mathbf{Z_L}$ has. Therefore, the settings used to generate learning parameters and equations (3-5) to transform them are conserved for the validation data.

Details of learning and validation sets

The ProDiMES software includes example files *EFS_Output.mat* and *EFS_fault_conditions.mat*. An example diagnostic solution employs these files as input information to diagnose a fleet of engines with different fault, and a set of solution accuracy performance metrics is determined. In order to have comparable metrics, the examined diagnostic technique examined uses the same ProDiMES files as validation files *EFS_Output_3.mat* and *EFS_fault_conditions_3.mat*. The settings to generate these files are brought together in Table 5.5.

Table 5.5 Settings for the validation files

Setting name	Setting value
Number of engine health conditions	19
Number of engines per health condition	10
Number of flights per engine	50
Fault initiation	Random
Minimum initiation flight	11
Fault evolution	Random
Rapid fault evolution rate (minimum)	9
Rapid fault evolution rate (maximum)	9
Sensor noise	On

Since the first 10 flights of each engine are fault-free, the data of the subsequent 40 flights are employed to compute patterns $\vec{Z}$. In this way, a total number of validation patterns is 19×10×40=7600. The learning files *EFS_Output_2.mat* and *EFS_fault_conditions_2.mat* were created using the same settings of Table 5.5 excepting the number of engines. An increased number 100 was used to have a large learning set of 76000 patterns and learn better the diagnostic techniques. As fault initiation is random, some flights since flight 11 may be free of faults. During forming the fault classification, the patterns of such flights for all 18 fault conditions were added to a no fault class increasing drastically its size. The no-fault class of great volume helps us to better adapt each recognition technique to this class. As a result, a probability of false alarms (false positive rate) goes down and approaches to the level of 0.001 exposed by the ProDiMES example solution. This is necessary to be able to compare other diagnostic performances.

5.3.3 Validation process

Validation of a fault recognition technique allows us to confirm its expected diagnostic performances. A validation process begins with the simulation of technique operation.

Recognition technique simulation (Step 6)

Each recognition technique, namely MLP, PNN, or SVM is simulated by employing the corresponding Matlab functions and the validation set patterns. For every pattern a diagnostic decision (classification decision) is taken. Using these decisions and ground truth information contained in the file EFS_fault_conditions_3.mat, a diagnostic performance, namely true classification rate, is determined.

True classification rate (Step 7)

The true classification rates are diagonal elements of a well-known confusion matrix and are probabilities of correct diagnosis of the patterns of corresponding classes. Since the classification includes healthy and faulty conditions, a probability $\bar{P}$, mean value of the diagonal elements, is an accuracy indicator for both fault detection and fault identification. This probability is used as a criterion to tune each diagnostic technique and then compare it with the other techniques tuned. The tuning is an important stage of a total diagnostic process and is described in the below sections separately for each technique under analysis.

5.3.4 Tuning

MLP

The dimensions of the input and output layers are known, 7 (number of monitored variables) and 19 (number of classes) respectively. The number of hidden layer neurons *Nn* determines network flexibility. This parameter as well as the number of learning cycles (epochs) *Nepochs* is determined during MLP tuning. Based on our previous experience, we began the tuning with the values *Nn=24* and *Nepochs=200* that were optimal for the diagnosis of an industrial gas turbine by MLP. However, it soon became clear that the network was not completely learned with these values. In particular, some classes with a small fault magnitude had the true positive rate equal to zero. To cope with the problem and find the conditions of the best learning, a series of calculations have been conducted with a gradual increase of the parameters *Nn* and *Nepochs*. As initial values of weight coefficients are maintained random, all the learning process and the final probabilities $\bar{P}$ are random as well. To reduce variations in the probabilities, each probability calculation with fixed settings *Nn* and *Nepochs* was repeated 10 times, and the corresponding values $\bar{P}$ were averaged resulting in an average probability P_{av}. Figure 5.13 illustrates this numeric experiment by plotting P_{av} against *Nepochs* for different *Nn*. As can be seen, the average probability still has a scatter about ±0.005. However, it does not hide a systematic influence of the parameters *Nepochs* and *Nn*: the probability P_{av} grows along with increasing these parameters. Such a trend has a natural explanation. A larger number of neurons means a higher network flexibility and better description of a complex fault classification. In its turn, a larger number *Nn* requires more epochs. On the basis of all the plots presented, an optimal values *Nn=66 and Nepochs=5000* have been chosen for further use.

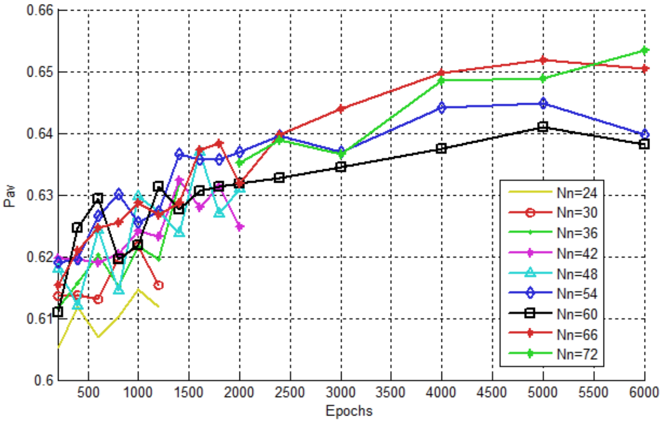


Figure 5.13. MLP tuning.

PNN

From the beginning of the tuning, it became clear that a PNN function could not operate with all 76000 learning patterns because of an evident lack of core memory. To proceed with the tuning, a reduced learning set of 30400 patterns that correspond to 40 engines per fault was created. The spread *b* is the unique parameter to tune PNN. The probability $\bar{P}$ is used as a tuning criterion because it is a deterministic quantity and does not need averaging. Figure 5.14 illustrating a dependency $\bar{P}$ *= f(b)* helps to choose the optimal value *b* = 0.2 used in further calculations.

SVM

Figure 5.15 illustrates the tuning performed for SVM. We can see here convergent lines of a constant classification error and the point of the minimum error 0.31 that gives us the optimal values of *C* and *σ*. Figure 5.16 shows an example of fault class construction using the learning set in the space of two normalized deviations. Also, it is visible how SVMs work trying to separate both classes as much as possible using the support vectors obtained from the training to create the boundary of classification.

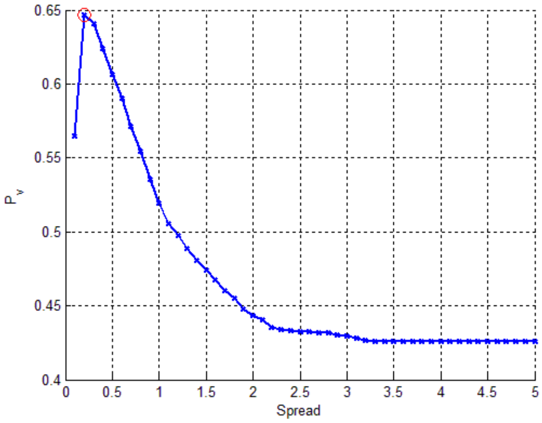


Figure 5.14. PNN tuning

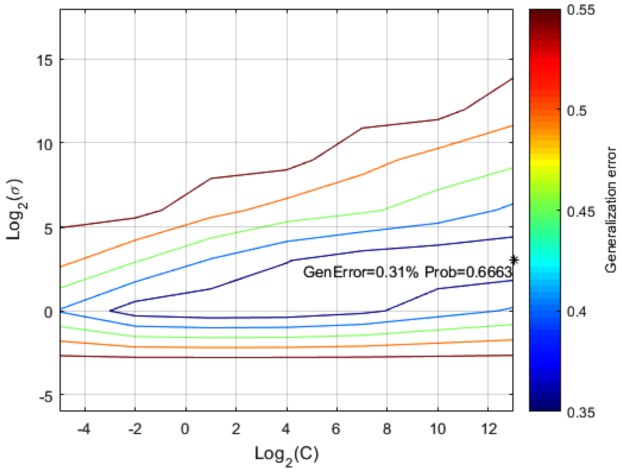


Figure 5.15. SVM tuning

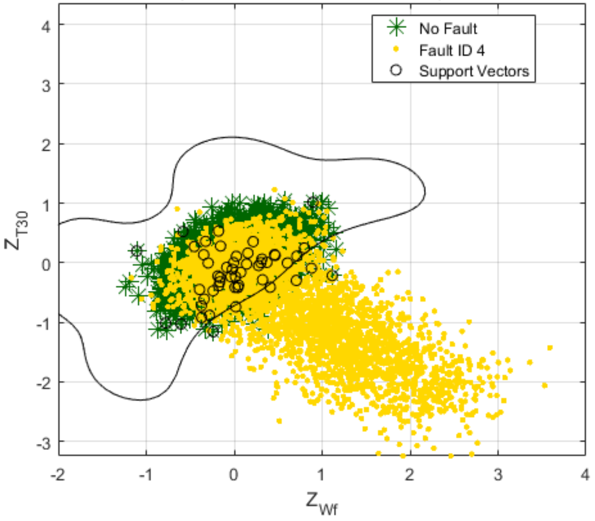


Figure 5.16. Example of class separation with SVM.

5.3.5 Comparison of the techniques (Step 8)

Comparison between the proposed candidate techniques

Thus, the candidate techniques, MLP, PNN, and SVM, are tuned, and they can now be compared. Let us firstly compare these techniques with each other. As PNN was tuned for the case of the reduced learning set, the other two techniques were additionally tuned on the data of this set. The resulting probabilities $\bar{P}$ (mean true classification rates) were placed in the first row of Table 5.6. It can be seen that SVM has the highest performance while PNN has the lowest one. Taking into account this negative rating and impossibility to use more perfect 100 engines-based learning set, from here we exclude PNN from the consideration. As shown in the second row for the case of 100 engines, both MLP and SVM further enhanced the performances thus increasing their advantages over PNN. Since the example solution uses exponential smoothing to improve diagnostic performances, we have also applied this filtering to the deviations of each engine in the learning and validation sets. As can be seen in the third row, the probabilities of both techniques have increased once more, and we leave them for further comparison using the data from the ProDiMES software and user' guide.

Table 5.6 Mean true classification rates $\bar{P}$ for the candidate techniques

Comparison Conditions	**MLP**	**PNN**	**SVM**
40 engines	0.6507	0.6466	**0.6663**
100 engines	0.6660	**-**	**0.6701**
100 engines+ Smoothing	0.7075	**-**	**0.7239**

Comparison with the ProDiMES data

Since ProDiMES implies the use of both cruise and takeoff measurements to make one diagnosis, the diagnostic algorithm was adapted to this multipoint option. Additional requirement of ProDiMES consists in a drastic reduction of false alarms given by the condition TNR>0.999. To meet this condition, the number of no-fault patterns was significantly increased in the learning set selecting 5000 no-fault engines for MLP and 10000 no-fault engines for SVM. Figure 5.17 and Table 5.7 show some metrics for different numbers of healthy engines using SVM. The previous experience of algorithm testing on cruise data allowed us to rapidly optimize the multipoint algorithm. In particular, the optimal conditions for MLP are smoothing coefficient 0.35, neuron number *Nn* = 72, and Early Stopping option. Two first rows of Table 5.8 show the general results obtained for the MLP- and SVM-based versions of the multipoint algorithm. We can see that MLP has slightly higher TPR, but SVM outperforms according the other metrics and in general.

The third row presents the metrics of the ProDiMES example solution applied to the same validation data as with the proposed algorithm. Comparing these results with the previous ones, one can state that both proposed versions have by far higher performances. This statement cannot be considered as our great achievement. The statement only means that the algorithm meets minimal requirements because the example solution was not optimized and is given only to illustrate the benchmarking methodology. The data of the fourth row are the metrics presented in the author's guide [72]. We can see that these metrics and SVM ones have small differences. More detailed metrics are presented in Figure 5.18 and Figure 5.19. Comparing the metrics separately for abrupt and rapid faults, we can conclude that the differences between the diagnostic solutions are not too great. In general, the metrics of the SVM and from the author's guide can be considered comparable. However, it is a preliminary conclusion since the author' guide does not refer to the origin of the metrics, namely, diagnostic solution and the data to which it was applied. It also follows from Figure 5.18 and Figure 5.19 that the probabilities of omitted faults (false negative rate) in the column "No Fault" are significant for both solutions. This can be explained by the following reasons.

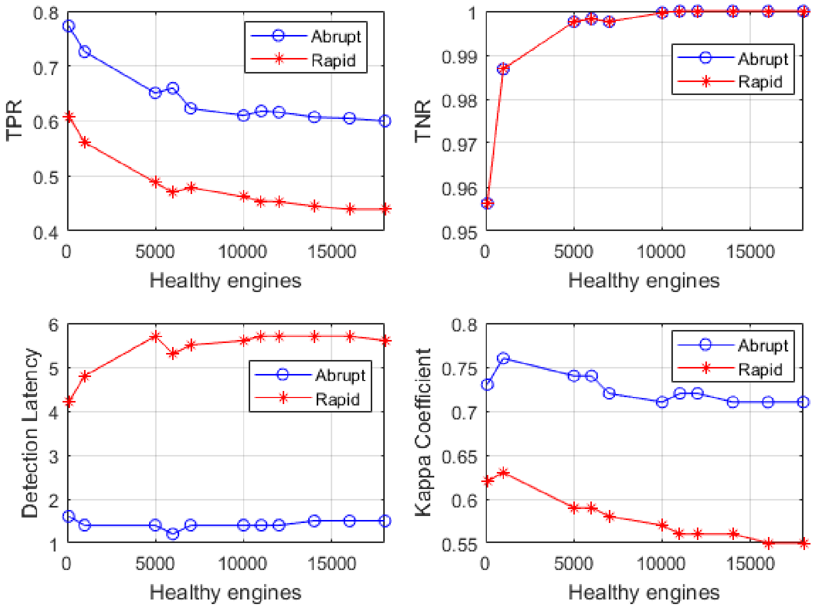


Figure 5.17. Metrics for different numbers of healthy engines using SVM.

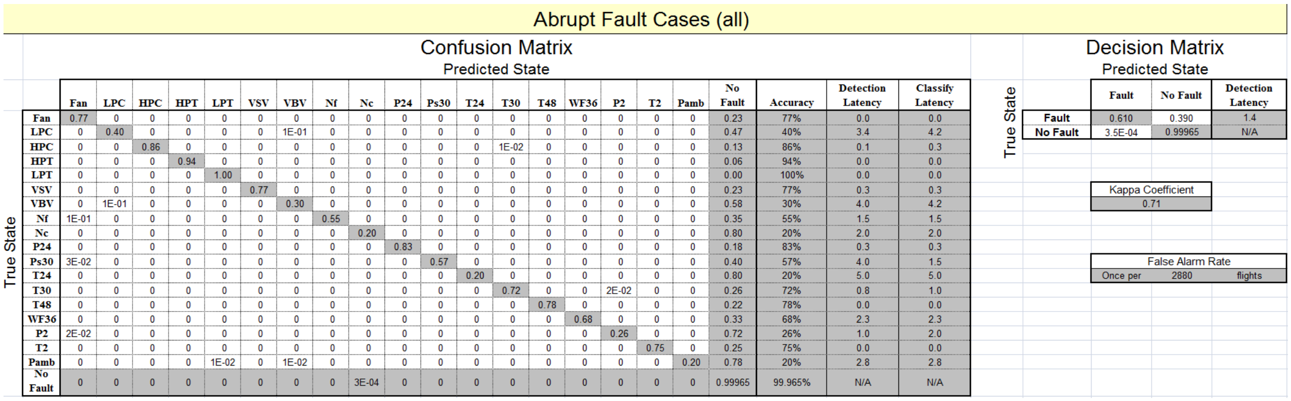
Table 5.7 Metrics for different numbers of healthy engines using SVM.

Number of healthy engines	Faults	TPR	TNR	Latency	Kappa Coefficient
100	Abrupt	0.7730	0.95625	1.6	0.73
	Rapid	0.6090	0.95625	4.2	0.62
1000	Abrupt	0.7260	0.98681	1.4	0.76
	Rapid	0.5610	0.98681	4.8	0.63
5000	Abrupt	0.6510	0.99757	1.4	0.74
	Rapid	0.4880	0.99757	5.7	0.59
6000	Abrupt	0.6600	0.99826	1.2	0.74
	Rapid	0.4700	0.99826	5.3	0.59
7000	Abrupt	0.6230	0.99757	1.4	0.72
	Rapid	0.4790	0.99757	5.5	0.58
10000	Abrupt	0.6100	**0.99965**	1.4	0.71
	Rapid	0.4620	**0.99965**	5.6	0.57
11000	Abrupt	0.6180	1	1.4	0.72
	Rapid	0.4540	1	5.7	0.56
12000	Abrupt	0.6160	1	1.4	0.72
	Rapid	0.4530	1	5.7	0.56
14000	Abrupt	0.6070	1	1.5	0.71
	Rapid	0.4450	1	5.7	0.56
16000	Abrupt	0.6050	1	1.5	0.71
	Rapid	0.4390	1	5.7	0.55
18000	Abrupt	0.6000	1	1.5	0.71
	Rapid	0.4390	1	5.6	0.55

Table 5.8 Performance metrics of the proposed method and from ProDiMES.

Algorithm	Faults	TPR	TNR	Detection Latency	Kappa
1-MLP-based algorithm	Abrupt	0.628	0.99931	1.8	0.69
	Rapid	0.466	0.99931	6.0	0.50
2-SVM-based algorithm	Abrupt	**0.610**	**0.99965**	**1.4**	**0.71**
	Rapid	**0.462**	**0.99965**	**5.6**	**0.57**
3-Example solution	Abrupt	0.505	0.99896	2.6	0.29
	Rapid	0.374	0.99896	7.0	0.21
4-Author's guide metrics	Abrupt	0.610	0.99997	2.5	0.73
	Rapid	0.419	0.99997	6.8	0.56

First, the false negative rate and false positive rate (probability of false alarms) are interconnected. The lower the latter, the greater the former. Thus, high probabilities of omitted faults are partly explained by a low probability of false alarms (0<0.001). Second, probabilities of omitted faults for such sensor faults as "P2", "Pamb", "Nc", and "T24" are particularly great (up to 0.86) that significantly affects the total level of fault classification accuracy. The explanation is related to the mode to determine a sensor fault in ProDiMES. Its magnitude is set by 10 standard deviations of measurement noise of the considered variable (monitored variable or operating condition). However, the fault is diagnosed in the space of deviations against the background of deviation noise. The latter depends on measurement errors of both monitored variables and operating conditions. For the mentioned four faults, the magnitude is too small in comparison with the deviation noise, only about 3 standard deviations. As a result, these faults are hardly diagnosed. Third, a total number 19 of considered classes is greater than in the most of investigations that objectively leads to a lower classification accuracy indicator. Presence of these faults in ProDiMES is natural because it promotes better benchmarking diagnostic algorithms. However, it seems to us that this number should be shortened in a real situation at the expense of the exclusion of hardly distinguishable faults. Let us now turn to the comparison of the proposed algorithm with the solutions of other researchers using the blind test case.



Abrupt Fault Cases (all)

Confusion Matrix

Predicted State

True State	Fan	LPC	HPC	HPT	LPT	VSV	VBV	Nf	Nc	P24	Ps30	T24	T30	T48	WF36	P2	T2	Pamb	No Fault	Accuracy	Detection Latency	Classify Latency
Fan	0.77	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.23	77%	0.0	0.0
LPC	0	0.40	0	0	0	0	1E-01	0	0	0	0	0	0	0	0	0	0	0	0.47	40%	3.4	4.2
HPC	0	0	0.86	0	0	0	0	0	0	0	0	0	1E-02	0	0	0	0	0	0.13	86%	0.1	0.3
HPT	0	0	0	0.94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.06	94%	0.0	0.0
LPT	0	0	0	0	1.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0.00	100%	0.0	0.0
VSV	0	0	0	0	0	0.77	0	0	0	0	0	0	0	0	0	0	0	0	0.23	77%	0.3	0.3
VBV	0	1E-01	0	0	0	0	0.30	0	0	0	0	0	0	0	0	0	0	0	0.58	30%	4.0	4.2
Nf	1E-01	0	0	0	0	0	0	0.55	0	0	0	0	0	0	0	0	0	0	0.35	55%	1.5	1.5
Nc	0	0	0	0	0	0	0	0	0.20	0	0	0	0	0	0	0	0	0	0.80	20%	2.0	2.0
P24	0	0	0	0	0	0	0	0	0	0.83	0	0	0	0	0	0	0	0	0.18	83%	0.3	0.3
Ps30	3E-02	0	0	0	0	0	0	0	0	0	0.57	0	0	0	0	0	0	0	0.40	57%	4.0	1.5
T24	0	0	0	0	0	0	0	0	0	0	0	0.20	0	0	0	0	0	0	0.80	20%	5.0	5.0
T30	0	0	0	0	0	0	0	0	0	0	0	0	0.72	0	0	2E-02	0	0	0.26	72%	0.8	1.0
T48	0	0	0	0	0	0	0	0	0	0	0	0	0	0.78	0	0	0	0	0.22	78%	0.0	0.0
WF36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.68	0	0	0	0.33	68%	2.3	2.3
P2	2E-02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.26	0	0	0.72	26%	1.0	2.0
T2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.75	0	0.25	75%	0.0	0.0
Pamb	0	0	0	0	1E-02	0	1E-02	0	0	0	0	0	0	0	0	0	0	0.20	0.78	20%	2.8	2.8
No Fault	0	0	0	0	0	0	0	0	3E-04	0	0	0	0	0	0	0	0	0	0.99965	99.965%	N/A	N/A

Decision Matrix

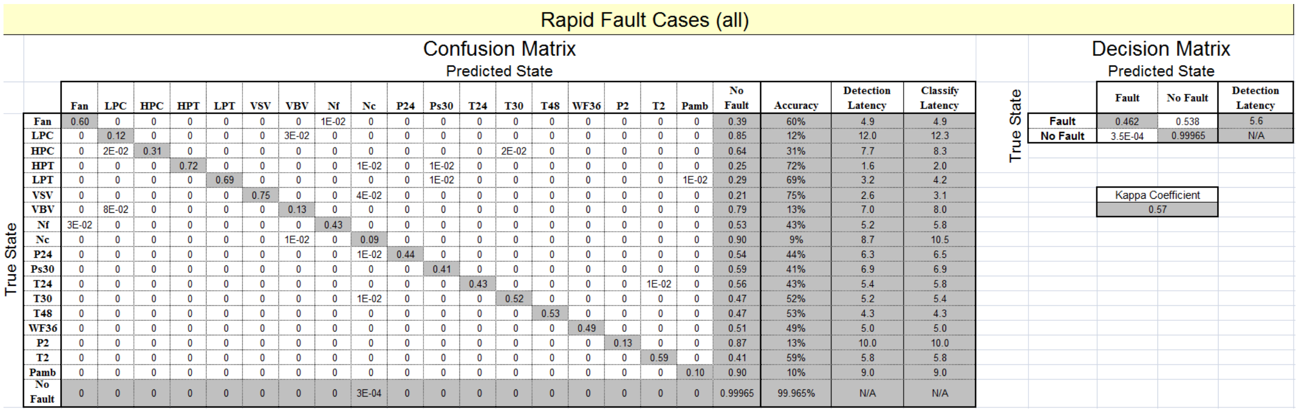
Predicted State

True State	Fault	No Fault	Detection Latency
Fault	0.610	0.390	1.4
No Fault	3.5E-04	0.99965	N/A

Kappa Coefficient
0.71

False Alarm Rate		
Once per	2880	flights

Figure 5.18. SVM-based algorithm metrics.



Rapid Fault Cases (all)

Confusion Matrix

Predicted State

True State

	Fan	LPC	HPC	HPT	LPT	VSV	VBV	Nf	Nc	P24	Ps30	T24	T30	T48	WF36	P2	T2	Pamb	No Fault	Accuracy	Detection Latency	Classify Latency
Fan	0.60	0	0	0	0	0	0	1E-02	0	0	0	0	0	0	0	0	0	0	0.39	60%	4.9	4.9
LPC	0	0.12	0	0	0	0	3E-02	0	0	0	0	0	0	0	0	0	0	0	0.85	12%	12.0	12.3
HPC	0	2E-02	0.31	0	0	0	0	0	0	0	0	0	2E-02	0	0	0	0	0	0.64	31%	7.7	8.3
HPT	0	0	0	0.72	0	0	0	0	1E-02	0	1E-02	0	0	0	0	0	0	0	0.25	72%	1.6	2.0
LPT	0	0	0	0	0.69	0	0	0	0	0	1E-02	0	0	0	0	0	0	1E-02	0.29	69%	3.2	4.2
VSV	0	0	0	0	0	0.75	0	0	4E-02	0	0	0	0	0	0	0	0	0	0.21	75%	2.6	3.1
VBV	0	8E-02	0	0	0	0	0.13	0	0	0	0	0	0	0	0	0	0	0	0.79	13%	7.0	8.0
Nf	3E-02	0	0	0	0	0	0	0.43	0	0	0	0	0	0	0	0	0	0	0.53	43%	5.2	5.8
Nc	0	0	0	0	0	0	1E-02	0	0.09	0	0	0	0	0	0	0	0	0	0.90	9%	8.7	10.5
P24	0	0	0	0	0	0	0	0	1E-02	0.44	0	0	0	0	0	0	0	0	0.54	44%	6.3	6.5
Ps30	0	0	0	0	0	0	0	0	0	0	0.41	0	0	0	0	0	0	0	0.59	41%	6.9	6.9
T24	0	0	0	0	0	0	0	0	0	0	0	0.43	0	0	0	0	1E-02	0	0.56	43%	5.4	5.8
T30	0	0	0	0	0	0	0	0	1E-02	0	0	0	0.52	0	0	0	0	0	0.47	52%	5.2	5.4
T48	0	0	0	0	0	0	0	0	0	0	0	0	0	0.53	0	0	0	0	0.47	53%	4.3	4.3
WF36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.49	0	0	0	0.51	49%	5.0	5.0
P2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.13	0	0	0.87	13%	10.0	10.0
T2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.59	0	0.41	59%	5.8	5.8
Pamb	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.10	0.90	10%	9.0	9.0
No Fault	0	0	0	0	0	0	0	0	3E-04	0	0	0	0	0	0	0	0	0	0.99965	99.965%	N/A	N/A

Decision Matrix

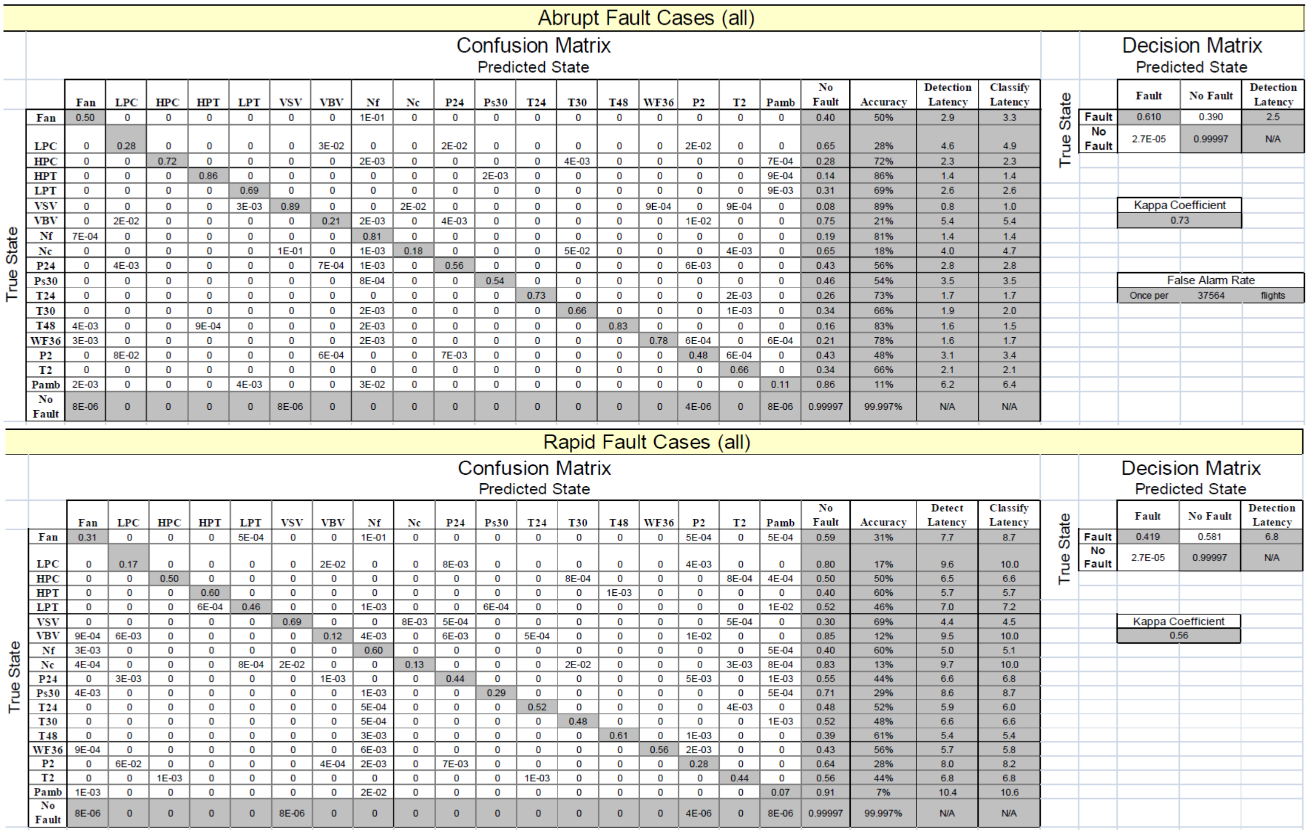
Predicted State

True State

	Fault	No Fault	Detection Latency
Fault	0.462	0.538	5.6
No Fault	3.5E-04	0.99965	N/A

Kappa Coefficient
0.57

Figure 5.18 SVM-based algorithm metrics (continuation).



Abrupt Fault Cases (all)

Confusion Matrix

Predicted State

True State

	Fan	LPC	HPC	HPT	LPT	VSV	VBV	Nf	Nc	P24	Ps30	T24	T30	T48	WF36	P2	T2	Pamb	No Fault	Accuracy	Detection Latency	Classify Latency
Fan	0.50	0	0	0	0	0	0	1E-01	0	0	0	0	0	0	0	0	0	0	0.40	50%	2.9	3.3
LPC	0	0.28	0	0	0	0	3E-02	0	0	2E-02	0	0	0	0	0	2E-02	0	0	0.65	28%	4.6	4.9
HPC	0	0	0.72	0	0	0	0	2E-03	0	0	0	0	4E-03	0	0	0	0	7E-04	0.28	72%	2.3	2.3
HPT	0	0	0	0.86	0	0	0	0	0	0	2E-03	0	0	0	0	0	0	9E-04	0.14	86%	1.4	1.4
LPT	0	0	0	0	0.69	0	0	0	0	0	0	0	0	0	0	0	0	9E-03	0.31	69%	2.6	2.6
VSV	0	0	0	0	3E-03	0.89	0	0	2E-02	0	0	0	0	0	9E-04	0	9E-04	0	0.08	89%	0.8	1.0
VBV	0	2E-02	0	0	0	0	0.21	2E-03	0	4E-03	0	0	0	0	0	1E-02	0	0	0.75	21%	5.4	5.4
Nf	7E-04	0	0	0	0	0	0	0.81	0	0	0	0	0	0	0	0	0	0	0.19	81%	1.4	1.4
Nc	0	0	0	0	0	1E-01	0	1E-03	0.18	0	0	0	5E-02	0	0	0	4E-03	0	0.65	18%	4.0	4.7
P24	0	4E-03	0	0	0	0	7E-04	1E-03	0	0.56	0	0	0	0	0	6E-03	0	0	0.43	56%	2.8	2.8
Ps30	0	0	0	0	0	0	0	8E-04	0	0	0.54	0	0	0	0	0	0	0	0.46	54%	3.5	3.5
T24	0	0	0	0	0	0	0	0	0	0	0	0.73	0	0	0	0	2E-03	0	0.26	73%	1.7	1.7
T30	0	0	0	0	0	0	0	2E-03	0	0	0	0	0.66	0	0	0	1E-03	0	0.34	66%	1.9	2.0
T48	4E-03	0	0	9E-04	0	0	0	2E-03	0	0	0	0	0	0.83	0	0	0	0	0.16	83%	1.6	1.5
WF36	3E-03	0	0	0	0	0	0	2E-03	0	0	0	0	0	0	0.78	6E-04	0	6E-04	0.21	78%	1.6	1.7
P2	0	8E-02	0	0	0	0	6E-04	0	0	7E-03	0	0	0	0	0	0.48	6E-04	0	0.43	48%	3.1	3.4
T2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.66	0	0.34	66%	2.1	2.1
Pamb	2E-03	0	0	0	4E-03	0	0	3E-02	0	0	0	0	0	0	0	0	0	0.11	0.86	11%	6.2	6.4
No Fault	8E-06	0	0	0	0	8E-06	0	0	0	0	0	0	0	0	0	4E-06	0	8E-06	0.99997	99.997%	N/A	N/A

Decision Matrix

Predicted State

True State

	Fault	No Fault	Detection Latency
Fault	0.610	0.390	2.5
No Fault	2.7E-05	0.99997	N/A

Kappa Coefficient
0.73

False Alarm Rate		
Once per	37564	flights

Rapid Fault Cases (all)

Confusion Matrix

Predicted State

True State

	Fan	LPC	HPC	HPT	LPT	VSV	VBV	Nf	Nc	P24	Ps30	T24	T30	T48	WF36	P2	T2	Pamb	No Fault	Accuracy	Detect Latency	Classify Latency
Fan	0.31	0	0	0	5E-04	0	0	1E-01	0	0	0	0	0	0	0	5E-04	0	5E-04	0.59	31%	7.7	8.7
LPC	0	0.17	0	0	0	0	2E-02	0	0	8E-03	0	0	0	0	0	4E-03	0	0	0.80	17%	9.6	10.0
HPC	0	0	0.50	0	0	0	0	0	0	0	0	0	8E-04	0	0	0	8E-04	4E-04	0.50	50%	6.5	6.6
HPT	0	0	0	0.60	0	0	0	0	0	0	0	0	0	1E-03	0	0	0	0	0.40	60%	5.7	5.7
LPT	0	0	0	6E-04	0.46	0	0	1E-03	0	0	6E-04	0	0	0	0	0	0	1E-02	0.52	46%	7.0	7.2
VSV	0	0	0	0	0	0.69	0	0	8E-03	5E-04	0	0	0	0	0	0	5E-04	0	0.30	69%	4.4	4.5
VBV	9E-04	6E-03	0	0	0	0	0.12	4E-03	0	6E-03	0	5E-04	0	0	0	1E-02	0	0	0.85	12%	9.5	10.0
Nf	3E-03	0	0	0	0	0	0	0.60	0	0	0	0	0	0	0	0	0	5E-04	0.40	60%	5.0	5.1
Nc	4E-04	0	0	0	8E-04	2E-02	0	0	0.13	0	0	0	2E-02	0	0	0	3E-03	8E-04	0.83	13%	9.7	10.0
P24	0	3E-03	0	0	0	0	1E-03	0	0	0.44	0	0	0	0	0	5E-03	0	1E-03	0.55	44%	6.6	6.8
Ps30	4E-03	0	0	0	0	0	0	1E-03	0	0	0.29	0	0	0	0	0	0	5E-04	0.71	29%	8.6	8.7
T24	0	0	0	0	0	0	0	5E-04	0	0	0	0.52	0	0	0	0	4E-03	0	0.48	52%	5.9	6.0
T30	0	0	0	0	0	0	0	5E-04	0	0	0	0	0.48	0	0	0	0	1E-03	0.52	48%	6.6	6.6
T48	0	0	0	0	0	0	0	3E-03	0	0	0	0	0	0.61	0	1E-03	0	0	0.39	61%	5.4	5.4
WF36	9E-04	0	0	0	0	0	0	6E-03	0	0	0	0	0	0	0.56	2E-03	0	0	0.43	56%	5.7	5.8
P2	0	6E-02	0	0	0	0	4E-04	2E-03	0	7E-03	0	0	0	0	0	0.28	0	0	0.64	28%	8.0	8.2
T2	0	0	1E-03	0	0	0	0	0	0	0	0	1E-03	0	0	0	0	0.44	0	0.56	44%	6.8	6.8
Pamb	1E-03	0	0	0	0	0	0	2E-02	0	0	0	0	0	0	0	0	0	0.07	0.91	7%	10.4	10.6
No Fault	8E-06	0	0	0	0	8E-06	0	0	0	0	0	0	0	0	0	4E-06	0	8E-06	0.99997	99.997%	N/A	N/A

Decision Matrix

Predicted State

True State

	Fault	No Fault	Detection Latency
Fault	0.419	0.581	6.8
No Fault	2.7E-05	0.99997	N/A

Kappa Coefficient
0.56

Figure 5.19. ProDiMES author's guide metrics.

5.4 BLIND TEST RESULTS

Comparison with the diagnostic solutions published in papers

Paper [29] provides the first results of employing ProDiMES. The authors compare four methods. To detect faults, algorithm #1 embraces trend analysis of the measurement deviations filtered by exponential moving average; to identify faults, the method involves a weighted least squares technique. Algorithm #2 differs from method 1 only by the fault identification stage that employs a Probabilistic Neural Network (PNN) instead of the least squares technique. Algorithm #3 includes a constant gain extended Kalman filter for detecting faults and regularized least squares for identifying faults. Algorithm #4 presents a generalized observer used for both detection and identification.

To meet the ProDiMES requirement TNR>0.999 in the blind test, we were forced to increase the number of no-fault engines in the learning data up to 14000 before comparing our results with other solutions. Figure 5.20 and Table 5.9 show the metrics obtained for different numbers of healthy engines. For the aforementioned algorithms, Table 5.10 contains the ProDiMES blind test results averaged for abrupt and rapid faults. The table also includes the corresponding averaged performances of the proposed SVM-based algorithm applied to the same blind test data. As can be seen in the table, the proposed algorithm outperforms all the other algorithms for all the metrics excepting the insignificant worsening of TNR.

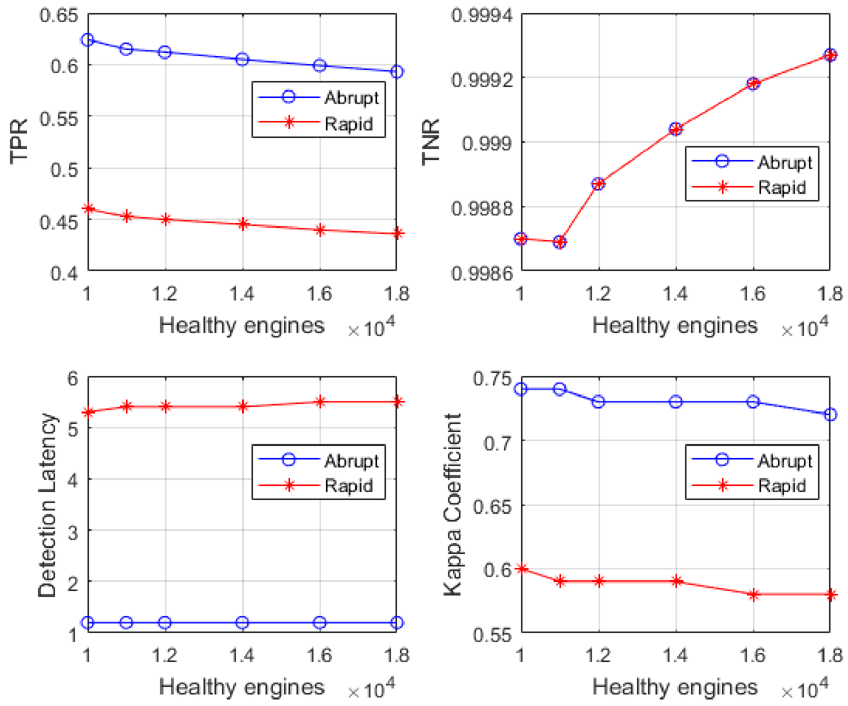


Figure 5.20. Metrics for different numbers of healthy engines using blind test set.

Table 5.9 Metrics for different numbers of healthy engines using blind test set

Number of healthy engines	Faults	TPR	TNR	Latency	Kappa Coefficient
10000	Abrupt	0.6240	0.99870	1.2	0.74
	Rapid	0.4600	0.99870	5.3	0.60
11000	Abrupt	0.6150	0.99869	1.2	0.74
	Rapid	0.4530	0.99869	5.4	0.59
12000	Abrupt	0.6120	0.99887	1.2	0.73
	Rapid	0.4500	0.99887	5.4	0.59
14000	Abrupt	0.6050	**0.99904**	1.2	0.73
	Rapid	0.4450	**0.99904**	5.4	0.59
16000	Abrupt	0.5990	0.99918	1.2	0.73
	Rapid	0.4400	0.99918	5.5	0.58
18000	Abrupt	0.5930	0.99927	1.2	0.72
	Rapid	0.4360	0.99927	5.5	0.58

Table 5.10 Performance metrics of the proposed algorithm and the algorithms analyzed in paper [29] (blind test data averaged for abrupt and rapid faults)

Algorithm	**TPR**	**TNR**	**Detection Latency**	**Kappa**
SVM-based Algorithm	**0.525**	**0.99904**	**3.30**	**0.660**
Algorithm #1	0.447	0.99908	4.86	0.588
Algorithm #2	0.447	0.99908	4.86	0.590
Algorithm #3	0.509	0.99908	4.02	0.627
Algorithm #4	0.519	0.99906	4.24	0.617

Paper [26] deals with five machine learning techniques: Naïve Bayes, Decision Tree (DT), K-Nearest Neighbors (K-NN), linear Support Vector Machine (SVM), and nonlinear SVM. At the first comparison stage, the authors apply all of the techniques to a binary problem (two fault classes at each classification stage) and, using three probabilistic criteria, have selected three best techniques: DT, K-NN, and nonlinear SVM. The comparison at the second stage with a multi-class classification showed that the nonlinear SVM achieved the highest average accuracy. At the third stage, a variation HSVMkSIR of this technique that uses the dimension reduction of a classification space displayed further accuracy enhancement.

To compute the ProDiMES metrics of the mentioned methods, the authors use a reduced fault classification of 10 items for which generate a proper set of simulated measurements. The metrics are determined separately for the cruise and takeoff regimes and are averaged for abrupt and rapid faults. These metrics for the best four methods as well as the corresponding metrics of the SVM-based algorithm are presented in Table 5.11. Comparing all the algorithms, we can see that the proposed diagnostic solution excels the others in all the metrics excepting detection latency. Given that the performances of paper [26] will degrade when changing to the full classification of 19 items, the superiority of the SVM-based algorithm seems to be most probable.

Table 5.11 Performance metrics of the proposed algorithm and the algorithms analyzed in paper [26] (data are obtained on cruise regime measurements and are averaged for abrupt and rapid faults)

Algorithm	**TPR**	**TNR**	**Detection latency**	**TCR**
SVM-based algorithm	**0.601**	**0.945**	**3.9**	**0.699**
DT	0.372	0.924	-	0.388
KNN	0.453	0.961	-	0.449
NSVM	0.705	0.728	-	0.535
HSVMkSIR	-	-	0.7	0.629

The authors of paper [27] propose and validate an algorithm for extracting features and detecting faults using their own set of measurements simulated by ProDiMES. The following techniques constitute the detection algorithm: a) a polynomial baseline model of a fleet-average engine and an individual baseline model to calculate deviations induced by faults; b) exponential moving average applied to measurement "deltas" and recursive median filter of deltas residuals; c) fault detection by tolerance monitoring of the sum of square residuals.

Table 5.12 shows the resulting metrics obtained on the cruise measurements for the fault detection algorithm and the proposed SVM-based one. The comparison results show that, the proposed algorithm is generally superior. This conclusion is again preliminary because the techniques are compared on different sets of measurement data.

Table 5.12 Performance metrics of the proposed algorithm and the detection algorithm analyzed in paper [27] (data are obtained on cruise regime measurements)

Algorithm	**Faults**	**TPR**	**TNR**	**Latency**
SVM-based Algorithm	Abrupt	**0.685**	**0.945**	**1.8**
	Rapid	**0.517**	**0.945**	**6.0**
2-Fault Detection Algorithm	Abrupt	0.455	0.999	1.7
	Rapid	0.271	0.999	6.7

As can be seen from the above comparison, some investigators generate by ProDiMES their own input data for the benchmarking and do not provide the blind test results. For this reason, a part of conclusions on the comparison of proposed diagnostic algorithm with such investigations are considered as preliminary. However, the conjunction of such positive preliminary conclusions and the conclusions made under correct comparison conditions makes sound the general statement about the algorithm's superiority.

Chapter 6: Comparative Study of Physics-Based and Data-Driven Gas Turbine Diagnostic Approaches

6.1 OVERVIEW

This chapter carries out a comparative study of two main gas turbine diagnostic approaches, physics-based and data-driven. They have the principal difference that the first approach performs engine diagnosis in the space of the deviations of measured variables δY while the second approach in the space of health parameters deviations $\delta\theta$. In each space, the classification of engine faults is constructed and the multi-layer perceptron is applied to make a diagnostic decision. To verify whether the $\delta\vec{\theta}$ -space is beneficial to the diagnosis we repeat the diagnosis many times and compute a probability of correct diagnosis for each approach. The approaches are compared using this probability as a criterion. All necessary data to create the fault classification and test the diagnostic approaches come from the thermodynamic model of GT1 and GT2. Six comparison cases for GT1 and three for GT2 with different fault numbers, severity and complexity are considered to better evaluate and compare the approaches.

Figure 6.1 presents in highlighted boxes the influence factors used to realize the diagnostic algorithm. The selection includes: a turboshaft engine for industrial application; gas path diagnostic systems; an integrated algorithm that considers the stages of feature extraction, monitoring and diagnosis; four types of models (physics-based, data-driven, non-linear and linear models); the use of LSM to identify baseline and degraded models; one-point and multi-point diagnostic analysis; MLP for fault recognition; and diagnosis of single and multiple faults.

6.2. METHODOLOGY FOR COMPARING GAS TURBINE DIAGNOSTIC APPROACHES

6.2.1 General description

The methodology for comparing the two approaches is realized as an integrated algorithm shown in Figure 6.2. A description of the methodology steps for the approaches is given as follows. The first step deals with deviation generation. Here, a the non-linear engine model $\vec{Y} = f(\vec{U}, \vec{\theta})$ of GT1, which allows the simulation of healthy and faulty engine scenarios, is transformed into a linear model $\delta\vec{Y} = \mathbf{H} \cdot \delta\vec{\theta}$ to simplify the computation of deviation patterns that describe the actual engine health state and form fault classifications.

Deviations $\delta\vec{Y}^*$ for the data-driven approach are computed first and then used to estimate health parameter deviations $\delta\hat{\vec{\theta}}^*$ for the physics-based approach. In the second step, learning sets are created and used in the third step to feed the networks so they can learn the fault classifications. In the fourth step, validation sets are built and introduced to the fifth step to evaluate if the trained networks can generalize the knowledge by correctly classifying unseen patterns. The network outputs produced in this fifth step are called diagnoses. Based on the diagnoses obtained, the sixth step computes confusion matrices for each approach, and a final diagnosis accuracy indicator is obtained, which is the criterion of approach comparison. The methodology presented allows a fair comparison and the only difference between the two approaches relies in the estimation of health parameter deviations.

6.2.2 Approach 1 (data-driven): diagnostics in $\delta\vec{Y}$ -space

The first approach is performed in the $\delta\vec{Y}$ -space and includes six main steps described below.

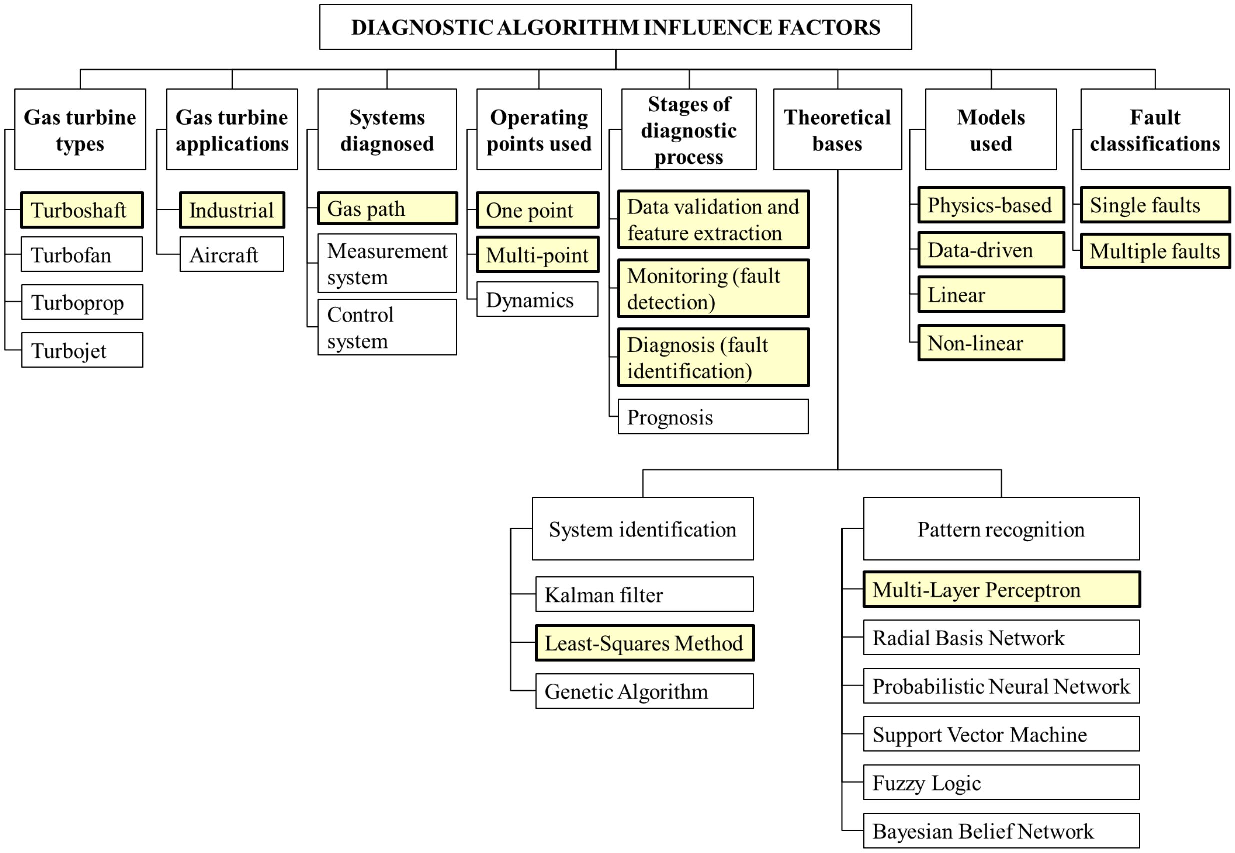


Figure 6.1 Diagnostic algorithm influence factors used throughout Chapter 6.

Step 1: Deviation generation

A gas turbine diagnostic process usually includes a preliminary procedure of computing deviations $\delta Y_i^* = \left(Y^* - Y_0\left(\vec{U}\right)\right)/Y_0\left(\vec{U}\right)$ between actual and baseline values of a monitored variable. Since real engine data to form representative fault classes are not available, simulated deviations with varying fault severities are computed instead through $\delta\vec{Y} = \mathbf{H} \cdot \delta\vec{\theta}$. During the computations, random fault severities are inside predetermined fault limits (e.g. [0, −5%]) and are described by a uniform distribution. Considering deviation errors (noise), simulated deviations take the form:

$$\delta Y^* = \delta Y + \varepsilon_{\delta Y} \qquad (6.1)$$

The deviation noise is simulated using the normal distribution. The deviations computed for all of the monitored variables constitute a $(m \times 1)$ deviation vector $\delta\vec{Y}^*$ (pattern) that is an input to a fault recognition technique and, on the other hand, forms a diagnostic space called $\delta\vec{Y}$ -space.

Step 2: Pattern-based classification (learning data formation)

A classification necessary for pattern recognition techniques can be now constituted by different engine fault classes presented by patterns $\delta\vec{Y}^*$. In the classification, we consider two types of classes; single and multiple. The single type is formed when there is a change of one fault parameter from a healthy state to a maximal severity, while the multiple one corresponds to an independent variation of two or more fault parameters from the same or different engine components. To avoid the problem of exceeding a severity limit, the present investigation applies the Archimedean spiral introduced in [124].

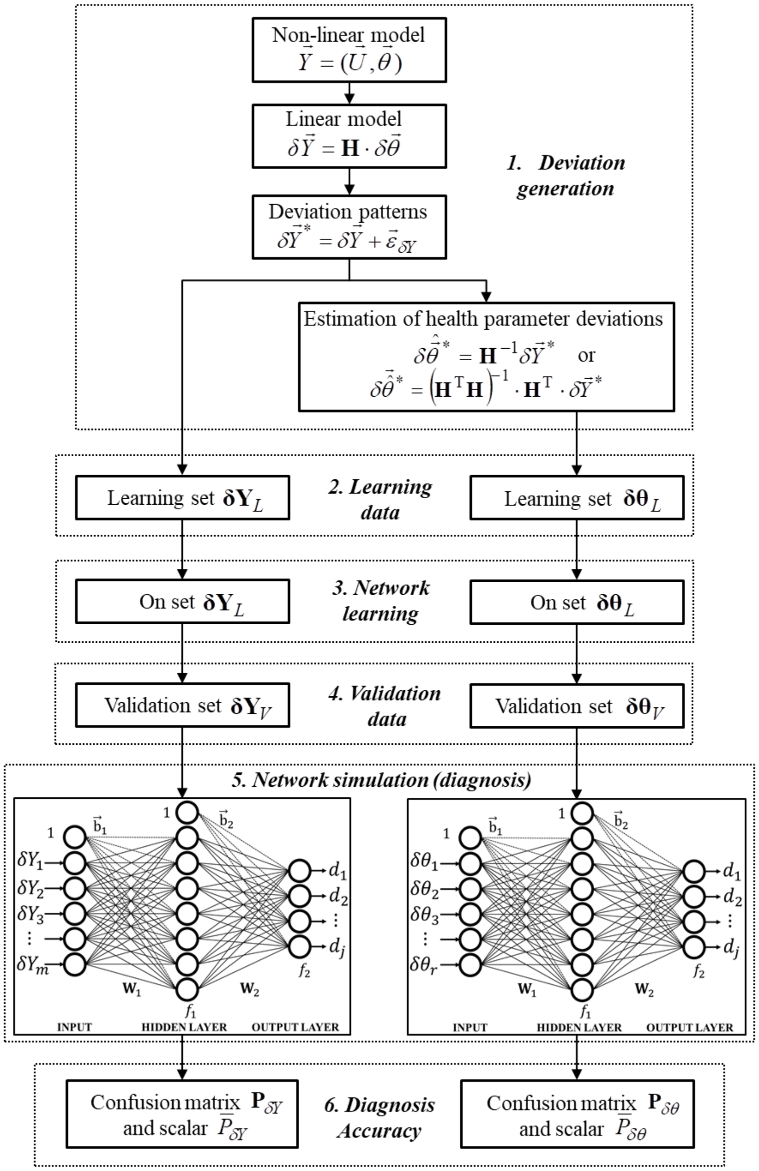

Non-linear model
$\vec{Y} = (\vec{U}, \vec{\theta})$
Linear model
$\delta\vec{Y} = \mathbf{H} \cdot \delta\vec{\theta}$
1. Deviation generation
Deviation patterns
$\delta\vec{Y}^* = \delta\vec{Y} + \vec{\varepsilon}_{\delta Y}$
Estimation of health parameter deviations
$\delta\hat{\vec{\theta}}^* = \mathbf{H}^{-1}\delta\vec{Y}^*$ or
$\delta\hat{\vec{\theta}}^* = (\mathbf{H}^T\mathbf{H})^{-1} \cdot \mathbf{H}^T \cdot \delta\vec{Y}^*$
Learning set $\boldsymbol{\delta}\mathbf{Y}_L$
2. Learning data
Learning set $\boldsymbol{\delta\theta}_L$
On set $\boldsymbol{\delta}\mathbf{Y}_L$
3. Network learning
On set $\boldsymbol{\delta\theta}_L$
Validation set $\boldsymbol{\delta}\mathbf{Y}_V$
4. Validation data
Validation set $\boldsymbol{\delta\theta}_V$
5. Network simulation (diagnosis)
1
$\vec{b}_1$
$\vec{b}_2$
δY_1
δY_2
δY_3
δY_m
d_1
d_2
d_j
f_2
f_1
W_1
W_2
INPUT
HIDDEN LAYER
OUTPUT LAYER
$\delta\theta_1$
$\delta\theta_2$
$\delta\theta_3$
$\delta\theta_r$
Confusion matrix $\mathbf{P}_{\delta Y}$ and scalar $\bar{P}_{\delta Y}$
6. Diagnosis Accuracy
Confusion matrix $\mathbf{P}_{\delta\theta}$ and scalar $\bar{P}_{\delta\theta}$

Figure 6.2 Methodology for comparing the two gas turbine diagnostic approaches.

From the constructed classes, a healthy class can be formed by all incipient faults, i.e. patterns not exceeding a fixed healthy engine boundary (e.g. a sphere of radius *R*=1) from each fault class. In this manner, we can address anomaly detection (motoring) and fault identification (diagnosis) as an integrated process [115] and a unique recognition technique can realize both stages at the same time. A complete fault classification reunites deviation patterns $\delta\vec{Y}^*$ of different classes in a matrix $\boldsymbol{\delta}\mathbf{Y}_L$ called learning set employed to train the selected recognition technique.

<u>Step 3: Network learning</u>

The technique used for gas turbine fault recognition is the MLP. In the learning stage, all patterns of the learning set $\boldsymbol{\delta}\mathbf{Y}_L$ are introduced by turn to the network at each iteration (epoch) and a total error between network outputs and target vectors is computed. Unknown network's weights and biases are corrected in the direction that reduces the error. After multiple epochs, the learning process comes to a minimum, and the network becomes ready to recognize new fault patterns.

<u>Step 4: Validation data formation</u>

To verify whether the trained technique correctly recognizes unseen fault patterns $\delta\vec{Y}^*$, it is necessary the creation of another set $\boldsymbol{\delta}\mathbf{Y}_V$ called a validation set. It is built in the same manner as $\boldsymbol{\delta}\mathbf{Y}_L$ with the exception that a different seed (different series of random numbers) is utilized to determine random fault severity and deviation noise.

<u>Step 5: Network simulation</u>

When a validation pattern enters to the trained network, the network's outputs show how close is this pattern to each fault class. Final diagnostic decision consists in choosing the closest class.

<u>Step 6: Diagnosis accuracy</u>

Each validation pattern belongs to a known fault class. Comparing a diagnosis d_l with a known class D_j for all validation patterns, probabilities $P_{lj} = P(d_l / D_j)$ are computed. These probabilities are elements of a confusion matrix. In this matrix, the diagonal values corresponding to correct class recognition are true positive rates. An average $\bar{P}_{\delta Y}$ of the diagonal represents a probability that characterizes total engine diagnosis reliability.

6.2.3 Approach 2 (physics-based): diagnostics in $\delta\vec{\theta}$ -space

For the physics-based approach, the steps required for diagnostics are the same as those presented in the data-driven approach with the exception of Step 1. In this step, health parameter deviations $\delta\vec{\hat{\theta}}^*$ (with deviation error component $\vec{\varepsilon}_{\delta\Theta}$ included) are additionally estimated with $\delta\vec{\hat{\theta}}^* = \mathbf{H}^{-1}\delta\vec{Y}^*$ if the number of monitored variables equals the number of health parameters, or $\delta\vec{\hat{\theta}}^* = \left(\mathbf{H}^{\mathrm{T}}\mathbf{H}\right)^{-1} \cdot \mathbf{H}^{\mathrm{T}} \cdot \delta\vec{Y}^*$ if the number of monitored variables is more than the number of health parameters [128]. In this way, the fault recognition can be carried out directly using the estimated health parameter deviations in a diagnostic space called $\delta\vec{\theta}$ -space. As before, data samples $\boldsymbol{\delta\theta}_L$ (learning set) and $\boldsymbol{\delta\theta}_V$ (validation set) are created for the network training and network simulation steps, respectively. Finally, the indicator of diagnosis accuracy $\bar{P}_{\delta\theta}$ is computed.

6.3 COMPARISON OF APPROACHES USING GT1

6.3.1 Comparison conditions

Selected variables

The vector $\vec{U}$ of operating conditions is specified in Table 6.1. Table 6.2 presents the six gas path monitored variables used to form deviations for both approaches and to diagnose the engine. Flow capacity (G) and isentropic efficiency (Ef) are the parameters used to describe engine faults in each component and correspond to health parameters commonly employed for gas turbine diagnostics [29,51]. These fault parameters per component are shown in Table 6.3. The fault severity interval for both fault parameters is [0, −5%].

For the purposes of the diagnostic analysis, six operating points are used. They are set by the gas generator rotor speeds (n_C=10700, 10500, 10300, 10100, 9900, and 9700 rpm) and standard atmospheric conditions. To confirm that all these regimes do not vary significantly, Table 6.4 shows maximal and idle regimes.

Table 6.1 Operating conditions (Vector $\vec{U}$) of GT1

No.	Description	Unit	Symbol
1	Ambient pressure	kPa	P_H
2	Inlet temperature	K	T_H
3	Gas generator rotation speed	rpm	n_C

Table 6.2 Monitored variables (Vector $\vec{Y}$) of GT1

No.	Description	Symbol
1	Compressor total pressure	P_C
2	High pressure turbine total pressure	P_{HPT}
3	Compressor total temperature	T_C
4	High pressure turbine total temperature	T_{HPT}
5	Power turbine total temperature	T_{PT}
6	Fuel gas mass flow	G_f

Table 6.3 Simulated fault parameters (Vector $\vec{\theta}$) of GT1

No.	Description	Symbol	Fault parameter interval
1	Compressor air flow capacity	G_C	0 to −5%
2	Compressor efficiency	Ef_C	0 to −5%
3	High pressure turbine flow capacity	G_{HPT}	0 to −5%
4	High pressure turbine efficiency	Ef_{HPT}	0 to −5%
5	Power turbine flow capacity	G_{PT}	0 to −5%
6	Power turbine efficiency	Ef_{PT}	0 to −5%

Table 6.4 Influence matrices for maximal and idle regimes (GT1).

Regime (rpm)	**Monitored variables**	**Fault parameters**					
		G_C	**Ef_C**	**G_{HPT}**	**Ef_{HPT}**	**G_{PT}**	**Ef_{PT}**
10700	**P_C**	1.2036	-0.5815	-0.7631	-0.6494	-0.3772	0.0126
	P_{HPT}	1.2415	-0.5857	0.2719	-0.8040	-1.5021	0.0492
	T_C	0.3446	-0.7461	-0.2166	-0.1843	-0.1066	0.0035
	T_{HPT}	0.6371	-1.3212	0.4414	-1.7978	-1.0353	0.0334
	T_{PT}	0.4021	-1.2753	0.4050	-1.7263	-0.7482	-0.3133
	G_f	1.8671	-2.0188	0.7294	-2.7443	-1.5769	0.0509
9700	**P_C**	1.1464	-0.5652	-0.7752	-0.6278	-0.3135	0.0013
	P_{HPT}	0.9967	-0.4974	0.2316	-0.6911	-1.2299	0.0051
	T_C	0.2834	-0.6669	-0.1963	-0.1580	-0.0758	0.0003
	T_{HPT}	0.4372	-1.2537	0.4331	-1.7301	-0.8528	0.0035
	T_{PT}	0.1975	-1.1799	0.3882	-1.6193	-0.5735	-0.2301
	G_f	1.6376	-2.0637	0.7618	-2.8411	-1.3990	0.0057

Fault classifications

We propose three fault classifications to be applied in the approach comparison. They are presented in Table 6.5. Classification 1 consists of six single fault classes. Each single class is built by the variation of one independent fault parameter (flow capacity or efficiency) in an engine component. Classification 2 uses the previous six single classes plus six multiple fault classes, resulting in twelve classes. In the first three multiple classes, each class is formed by independently changing both fault parameters (flow capacity and efficiency) in the same engine component. In the other three multiple classes, each class is constructed using independent shifts of four fault parameters from the combination of two components. However, more robust diagnostic algorithms are required that can diagnose faults in the background of more complex scenarios. Classification 3 uses the same structure of Classification 2 with the inclusion of the healthy class (no fault).

For the two first fault classifications, the number of patterns per class is fixed to 1000 in both learning and validation sets because the use of this quantity has shown good recognition performance before [124]. However, for the third classification, the number of patterns is variable and depends on the healthy class boundary as mentioned before.

Table 6.5 Fault classifications

Fault Classif	**Fault classes and fault parameters**												
	D_1 δG_C	**D_2** $\delta\eta_C$	**D_3** δG_{HPT}	**D_4** $\delta\eta_{HPT}$	**D_5** δG_{PT}	**D_6** $\delta\eta_{PT}$	**D_7** δG_C $\delta\eta_C$	**D_8** δG_{HPT} $\delta\eta_{HPT}$	**D_9** δG_{PT} $\delta\eta_{PT}$	**D_{10}** δG_C $\delta\eta_C$ δG_{HPT} $\delta\eta_{HPT}$	**D_{11}** δG_C $\delta\eta_C$ δG_{PT} $\delta\eta_{PT}$	**D_{12}** δG_{HPT} $\delta\eta_{HPT}$ δG_{PT} $\delta\eta_{PT}$	**D_0** No-fault state
1	+	+	+	+	+	+	–	–	–	–	–	–	–
2	+	+	+	+	+	+	+	+	+	+	+	+	–
3	+	+	+	+	+	+	+	+	+	+	+	+	+

Network specifications

Let us briefly present some network specifications employed in the diagnostic approaches using GT1. The two networks contain input, hidden and output layers. In the case of one-point diagnostic analysis, the network input layer size utilized in the $\delta\vec{Y}$ -space correspond to the dimension of a deviation pattern vector $\delta\vec{Y}^*$ (six gas path monitored variables) while for $\delta\vec{\theta}$ -space the number of input layer nodes equals the dimension of an estimated deviation vector $\delta\hat{\vec{\theta}}^*$ (six fault parameters). However, when multi-point diagnostics is used in the $\delta\vec{Y}$ -space, the number of nodes in one-point is multiplied by the number of engine regimes (e.g. 6 basic nodes x 6 regimes=36 input nodes). The output layer size is the same for both networks and depends on the number of fault classes under analysis. The hidden layer nodes can be selected by user's experience or by a trial and error.

Common elements present in the networks are: an additional node in the input and output layers called bias vector with input value of one allowing the classifier to move the decision boundary; a cross-entropy function controlling the network performance by the training error minimization; a hyperbolic tangent sigmoid transfer function f_1 working within an interval of [1, −1] for hidden layer neuron activations; an output layer softmax transfer function f_2 (a generalization of logistic sigmoid function) working within an interval of [0, 1] to assign every input pattern to one of the given fault classes. More detailed information about neural network configurations can be found in [139].

6.3.2 Cases of comparative analysis

Design of numerical experiments

In order to draw solid conclusions about the diagnosis accuracy of both approaches and to make a fair comparison between them, five case studies are performed. They are formed by varying three principal characteristics: the fault classification, the type of diagnostic analysis (one-point or multi-point), and the deviation noise scheme (simulated or real). These cases are specified in Table 6.6.

Table 6.6 Cases studies proposed to evaluate both approaches using GT1.

Case study	Fault classification	Diagnostic analysis	Deviation noise
1	1	One-point*	Simulated
2	2	One-point*	Simulated
3	2	Multi-point	Simulated
4	3	Multi-point	Simulated
5	3	Multi-point	Real

*Maximal engine regime (n_C=10700 rpm)

For an adequate approach comparison in all cases, the optimal network training scheme and the optimal network parameters need to be first determined. The training scheme refers to the use or not of an early stopping option. The network parameters refer to the number of epochs and the number of hidden layer neurons. For each training scheme, the optimal parameters are found. For that purpose, the network with different values of these parameters are trained and then tested on the validation data set. The values that provide the highest $\overline{P}$ are chosen. Once the optimal parameters are determined, we select the training scheme with the best results. With the optimal network configuration, probabilities $\overline{P}$ for 100 seeds are computed and averaged, resulting in a more precise estimation $\overline{P}_{av}$ that will allow better comparison of the approaches.

The determination of optimal network configuration and final diagnosis accuracy of both approaches are presented for each case below.

Case study 1

Case 1 is formed using Classification 1 (6 single fault classes), one-point diagnosis at the maximal regime (n_C=10700 rpm) and simulated deviation noise scheme. Table 6.7 shows the optimal network configuration for both spaces and 1-seed computation. It can be seen that for $\delta\vec{Y}$ -space and $\delta\vec{\theta}$ -space the best training option is “early stopping=0”. However, the differences in $\bar{P}$ between both options remain small (0.25% for $\delta\vec{Y}$ -space and 0.16% for $\delta\vec{\theta}$ -space). Figure 6.3 shows the search of optimal parameters for $\delta\vec{\theta}$ -space and “early stopping=0”. The highest value ($\bar{P}_{\delta\theta}$ =0.8511) is marked with a circle for the optimal parameters (78 hidden neurons and 200 epochs). From the figure one tendency is clear in this case: increasing the number of epochs produces a decrease in the diagnosis accuracy.

Table 6.7 Optimal network configuration for Case 1 (1 seed)

Space	**Early stop=0**		**Early stop=1**		**Diff.**
	Optimal Param.	$\bar{P}$	**Optimal Param.**	$\bar{P}$	
$\delta\vec{Y}$	hn=48 ep=400	0.8558	hn=60	0.8533	0.0025
$\delta\vec{\theta}$	hn=78 ep=200	0.8511	hn=48	0.8495	0.0016

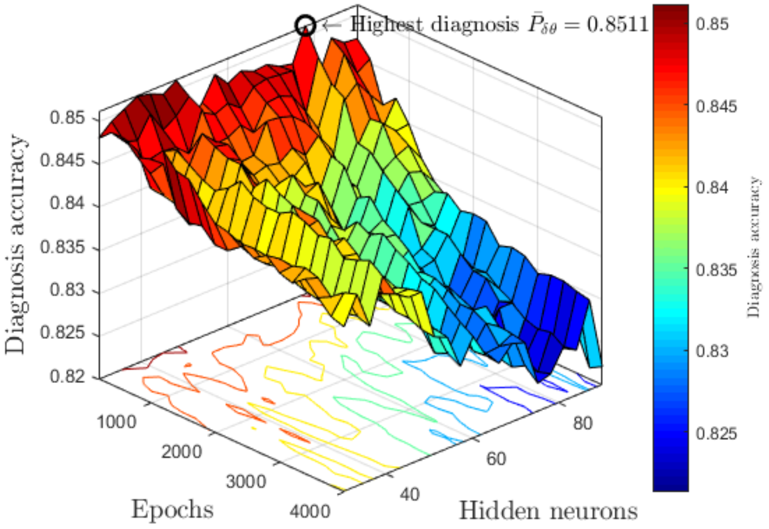


Figure 6.3 Search of optimal parameters for Case 1, $\delta\vec{\theta}$ -space and “early stopping=0”.

Using the optimal network configuration for each approach, the probabilities $\bar{P}$ of 100 seeds are computed and averaged to obtain the final diagnosis accuracy indicator $\bar{P}_{av}$. Table 6.8 presents this indicator as well as averaged true positive rates (vector $\vec{P}_{av}$) corresponding to correct recognition of each fault class. It is visible that the percentage of total correct recognition for both approaches is almost the same. Despite the acceptable level of accuracy achieved in Table 6.8 for $\delta\vec{\theta}$ -space, it is worth analyzing the estimation errors in deviations $\delta\hat{\vec{\theta}}^*$. Figure 6.4 presents patterns $\delta\hat{\vec{\theta}}^*$ of a pair of single fault classes ($\delta\hat{G}_c$ and $\delta\hat{G}_{pt}$) as well as true values (solid lines) of each class that uniformly develop along their corresponding fault axes from 0 to the maximal severity limit (−5%).

As shown, the distribution of estimated deviation noise $\vec{\varepsilon}_{\delta\theta}$ presents an ellipsoid form for both classes due to estimation errors resulting in a scattered pattern distribution and exceeded severity limits. Since the monitored variables in deviation errors $\vec{\varepsilon}_{\delta Y}$ are uncorrelated, the estimation errors are produced by the correlation of some health parameters in deviation errors $\vec{\varepsilon}_{\delta\theta}$ (see Figure 6.5).

Table 6.8 Averaged true classification rates and final accuracy indicator $\overline{P}_{av}$ for Case 1 (100 seeds).

Space	$\vec{P}_{av}$						$\overline{P}_{av}$
	$\mathbf{D_1}$	$\mathbf{D_2}$	$\mathbf{D_3}$	$\mathbf{D_4}$	$\mathbf{D_5}$	$\mathbf{D_6}$	
$\delta\vec{Y}$	0.9115	0.7691	0.8802	0.7481	0.8808	0.8919	0.8469
$\delta\vec{\theta}$	0.9118	0.7756	0.8770	0.7493	0.8756	0.8760	0.8442

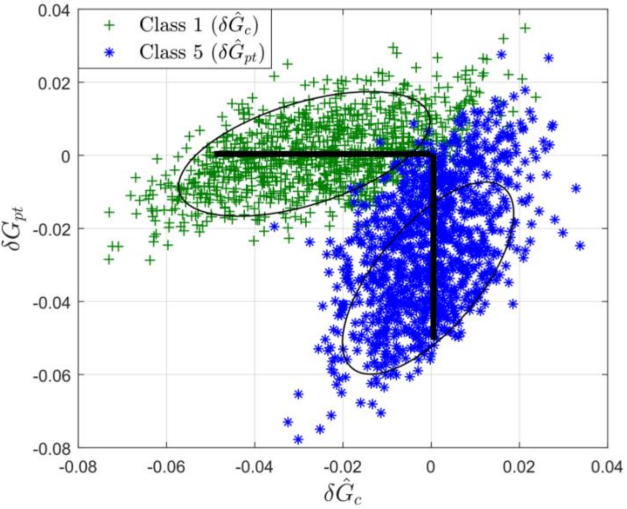


Figure 6.4 Classes of estimated patterns $\delta\hat{\vec{\theta}}^*$ for Case 1.

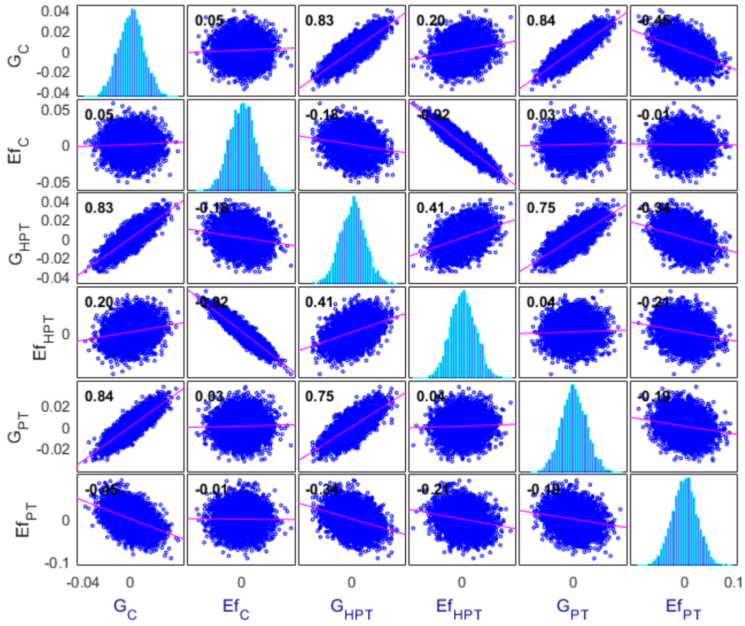


Figure 6.5 Correlation matrix of estimated deviation errors $\vec{\varepsilon}_{\delta\theta}$ for Case 1.

Case study 2

Case 2 corresponds to Classification 2 (twelve fault classes), one-point diagnosis and simulated deviation noise. Table 6.9 contains the optimal network configuration parameters and results for the two approaches. The option "Early stopping=0" shows the best results and the difference between training options is 0.65% for $\delta\vec{Y}$-space and 1.27% for $\delta\vec{\theta}$-space. For the best network configuration, Table 6.10 gives averaged true classification rates and final accuracy indicator for 100 seeds. The final recognition $\overline{P}_{av}$ is around 72.5% for both diagnostic spaces.

Table 6.9 Search of optimal network configuration for Case 2 (1 seed)

Space	**Early stop=0**		**Early stop=1**		**Diff.**
	Optimal Param.	$\overline{P}$	**Optimal Param.**	$\overline{P}$	
$\delta\vec{Y}$	hn=30 ep=1400	0.7285	hn=51	0.7220	0.0065
$\delta\vec{\theta}$	hn=30 ep=3800	0.7268	hn=90	0.7141	0.0127

* hn=hidden neurons; ep=epochs

Table 6.10 True classification rates $\vec{P}_{av}$ and final diagnosis accuracies $\overline{P}_{av}$ for Case 2 (100 seeds).

Space	$\vec{P}_{av}$												$\overline{P}_{av}$
	D$_1$	**D$_2$**	**D$_3$**	**D$_4$**	**D$_5$**	**D$_6$**	**D$_7$**	**D$_8$**	**D$_9$**	**D$_{10}$**	**D$_{11}$**	**D$_{12}$**	
$\delta\vec{Y}$	0.8811	0.6923	0.8630	0.6827	0.6665	0.8818	0.6267	0.6281	0.5595	0.7588	0.7268	0.7434	0.7259
$\delta\vec{\theta}$	0.8710	0.6901	0.8581	0.6858	0.6761	0.8830	0.6297	0.6278	0.5417	0.7564	0.7233	0.7380	0.7234

Case study 3

Case 3 has the same structure as Case 2 (twelve classes from Classification 2 and simulated noise) but instead of one-point, we use a multi-point diagnosis. Table 6.11 shows the search of optimal network configuration. Once again, "Early stopping=0" presents the best results (with a difference between training schemes of 0.46% and 0.48% for $\delta\vec{Y}$ and $\delta\vec{\theta}$ spaces, respectively). Table 6.12 shows averaged correct class recognition values contained in $\vec{P}_{av}$ as well as the total diagnosis accuracy $\overline{P}_{av}$. The level of recognition for the data-driven approach is 86.15% while for the physics-based approach is 87%. It is visible that the probabilities for all classes have considerably improved compared to the previous case.

Table 6.11 Search of optimal network configuration for Case 3 (1 seed)

Space	**Early stop=0**		**Early stop=1**		**Diff.**
	Optimal Param.	$\overline{P}$	**Optimal Param.**	$\overline{P}$	
$\delta\vec{Y}$	hn=36 ep=600	0.8667	hn=78	0.8621	0.0046
$\delta\vec{\theta}$	hn=36 ep=4000	0.8738	hn=57	0.8690	0.0048

* hn=hidden neurons; ep=epochs

Table 6.12 Averaged true classification rates $\vec{P}_{av}$ and final diagnosis accuracies $\overline{P}_{av}$ for Case 3 (100 seeds).

Space	$\vec{P}_{av}$												$\overline{P}_{av}$
	D_1	**D_2**	**D_3**	**D_4**	**D_5**	**D_6**	**D_7**	**D_8**	**D_9**	**D_{10}**	**D_{11}**	**D_{12}**	
$\delta\vec{Y}$	0.9353	0.8559	0.9354	0.8526	0.7973	0.9273	0.8062	0.8161	0.7069	0.9051	0.8959	0.9036	0.8615
$\delta\vec{\theta}$	0.9325	0.8709	0.9397	0.8639	0.8343	0.9426	0.8210	0.8254	0.6880	0.9090	0.9042	0.9091	0.8700

Case study 4

In Case 4, the structure of Case 3 (multi-point and simulated noise) is utilized with the inclusion of a healthy class (Configuration 3). In this way, thirteen classes are considered. Table 6.13 contains the optimal network configuration where “early stopping=0” has the best results (there is a difference in training schemes of 0.71% and 0.64% for $\delta\vec{Y}$ and $\delta\vec{\theta}$ spaces, respectively).

Table 6.13 Search of optimal network configuration for Case 4 (1 seed)

Space	**Early stop=0**		**Early stop=1**		**Diff.**
	Optimal Param.	$\overline{P}$	**Optimal Param.**	$\overline{P}$	
$\delta\vec{Y}$	hn=33 ep=400	0.8720	hn=81	0.8649	0.0071
$\delta\vec{\theta}$	hn=36 ep=2400	0.8800	hn=87	0.8736	0.0064

* hn=hidden neurons; ep=epochs

In Table 6.14, correct class recognition vectors $\vec{P}_{av}$ and probabilities $\overline{P}_{av}$ are shown. For the data-driven approach, the recognition is 86.6% while for the physics-based is 87.4%. Despite a new class was incorporated to the analysis, Case 4 maintains similar recognition level taking Case 3 as a reference.

Table 6.14 Averaged true classification rates $\vec{P}_{av}$ and final diagnosis accuracies $\overline{P}_{av}$ for Case 4 (100 seeds).

Space	$\vec{P}_{av}$													$\overline{P}_{av}$
	D_0	**D_1**	**D_2**	**D_3**	**D_4**	**D_5**	**D_6**	**D_7**	**D_8**	**D_9**	**D_{10}**	**D_{11}**	**D_{12}**	
$\delta\vec{Y}$	0.8708	0.9581	0.8874	0.9633	0.8827	0.8214	0.7784	0.8101	0.8232	0.7277	0.9030	0.8952	0.9027	0.8664
$\delta\vec{\theta}$	0.8778	0.9544	0.8990	0.9661	0.8922	0.8563	0.7966	0.8261	0.8316	0.7080	0.9086	0.9039	0.9082	0.8740

Although the correct class recognition in D_6 is acceptable (77.8% and 79.6% for $\delta\vec{Y}$ and $\delta\vec{\theta}$ spaces, respectively), there exist probabilities up to 20% wrongly assigned to the healthy class D_0 (see Table 6.15 showing the averaged confusion matrix with healthy class for $\delta\vec{Y}$-space). This incorrect classification can be explained as follows. The single class D_6 is formed by the variation of Ef_{PT} fault parameter, which in turn has a small influence on almost all monitored variables as can be seen in the values of the influence matrices. Due to this fact, and in spite of working with a maximal severity of −5%, the fault class D_6 constructed in any diagnostic space is much smaller in the background than the rest of the fault classes. Therefore, patterns from classes D_6 and D_0 are contained in a very close area provoking an incorrect classification for D_6. The risk of this type of problem related to Ef_{PT} is that the algorithm produces a “healthy state” diagnosis when in fact a fault is occurring. The problem related to Ef_{PT} is also visible in class D_9 (with probabilities of 72.7% and 70.8% for $\delta\vec{Y}$ and $\delta\vec{\theta}$ spaces, respectively), which is formed by the union of Ef_{PT} and G_{PT}, because incorrect diagnoses fall into single class D_5 (built using G_{PT}) with misclassification of 21.7%. As commented before, due to the small influence of Ef_{PT} on monitored variables, D_9 only receives the impact of G_{PT}. For that reason, D_5 and D_9 are very similar, thus provoking elevated incorrect diagnoses (Note that Cases 2 and 3 present the same problem with class D_9). For multiple classes D_{11} and D_{12} where Ef_{PT} is also present, the above problem is hidden due to the sum of other fault parameters. Despite the problems commented for Ef_{PT} and considering that Case 5 presents a complex fault classification, both approaches classify with a good level of accuracy this small class.

Table 6.15 Averaged validation confusion matrix (100 seeds) for Case 4 in δY -space

		True class												
		D0	**D1**	**D2**	**D3**	**D4**	**D5**	**D6**	**D7**	**D8**	**D9**	**D10**	**D11**	**D12**
Diagnosis	**d0**	0.8708	0.0228	0.0175	0.0261	0.0141	0.0150	0.2012	0.0045	0.0029	0.0160	0.0001	0.0005	0.0001
	d1	0.0063	0.9581	0	0	0	0	0.0002	0.0638	0	0	0.0036	0.0024	0
	d2	0.0043	0	0.8874	0	0.0564	0.0061	0.0003	0.0657	0.0133	0.0033	0.0111	0.0126	0.0012
	d3	0.0156	0.0001	0	0.9633	0	0	0	0	0.0361	0	0.0024	0	0.0011
	d4	0.0019	0	0.0477	0	0.8827	0.0044	0	0.0048	0.0784	0.0031	0.0057	0.0018	0.0119
	d5	0.0076	0	0.0038	0	0.0037	0.8214	0	0.0001	0.0003	0.2174	0	0.0019	0.0034
	d6	0.0814	0.0004	0.0003	0	0.0003	0	0.7784	0.0004	0.0001	0.0241	0	0.0011	0.0003
	d7	0.0037	0.0178	0.0150	0	0.0027	0	0.0039	0.8101	0.0001	0.0002	0.0421	0.0581	0
	d8	0.0020	0	0.0105	0.0100	0.0262	0.0003	0.0003	0.0002	0.8232	0.0003	0.0275	0	0.0618
	d9	0.0058	0	0.0056	0	0.0062	0.1518	0.0129	0.0003	0.0004	0.7277	0	0.0178	0.0122
	d10	0.0005	0.0005	0.0054	0.0004	0.0017	0	0.0009	0.0216	0.0127	0	0.9030	0.0040	0.0018
	d11	0.0001	0.0003	0.0062	0	0.0015	0.0001	0.0016	0.0284	0.0001	0.0037	0.0036	0.8952	0.0037
	d12	0	0	0.0007	0.0002	0.0045	0.000	0.0001	0	0.0324	0.0042	0.0009	0.0047	0.9027
$\vec{P}_{av}$		0.8708	0.9581	0.8874	0.9633	0.8827	0.8214	0.7784	0.8101	0.8232	0.7277	0.9030	0.8952	0.9027
$\overline{P}_{av}$ =0.8664														

Case study 5

Case 5 uses the same structure as Case 4 (thirteen classes and multi-point), but in order to have a more realistic scenario for both approaches, we included real monitored variable deviation errors $\varepsilon_{\delta Yi}$ in all simulated fault trajectories. Real noise is obtained from deviations based on real data with varying levels of degradation recorded as hourly steady-state snapshots from the same engine under analysis. First, engine deterioration is modeled through the expression $Y_i(\vec{U},\bar{t})$ using the Least-Squares Method (LSM) or any linear or non-linear regression method. Then, relative differences between measured values Y_i^* and degraded engine estimations are computed. Thus, real deviation errors are expressed by $\varepsilon_{\delta Yi} = \left(Y_i^* - Y_i(\vec{U},\bar{t})\right) / Y_i(\vec{U},\bar{t})$. Table 6.16 shows the comparison between training schemes as well as their optimal parameters. Once more, “early stopping=0” is the winner. Table 6.17 presents the true classification rates $\vec{P}_{av}$ and final diagnosis accuracy $\overline{P}_{av}$ for 100 seeds. Taking into account the previous case, probabilities $\overline{P}_{av}$ considerably improved (up to 94.5% and 94.1% for $\delta\vec{Y}$ and $\delta\vec{\theta}$ spaces, respectively).

Table 6.16 Search of optimal network configuration for Case 5 (1 seed)

Space	**Early stop=0**		**Early stop=1**		**Diff.**
	Optimal Param.	$\overline{P}$	**Optimal Param.**	$\overline{P}$	
$\delta\vec{Y}$	hn=36 ep=1000	0.9468	hn=57	0.9415	0.0053
$\delta\vec{\theta}$	hn=33 ep=3200	0.9411	hn=87	0.9366	0.0045

* hn=hidden neurons; ep=epochs

Table 6.17 Averaged true classification rates $\vec{P}_{av}$ and final diagnosis accuracies $\overline{P}_{av}$ for Case 5 (100 seeds).

Space	$\vec{P}_{av}$													$\overline{P}_{av}$
	D$_0$	**D$_1$**	**D$_2$**	**D$_3$**	**D$_4$**	**D$_5$**	**D$_6$**	**D$_7$**	**D$_8$**	**D$_9$**	**D$_{10}$**	**D$_{11}$**	**D$_{12}$**	
$\delta\vec{Y}$	0.9268	0.9578	0.9518	0.9844	0.9815	0.9635	0.9412	0.8653	0.9235	0.8892	0.9684	0.9703	0.9699	0.9455
$\delta\vec{\theta}$	0.9292	0.9551	0.9554	0.9831	0.9791	0.9616	0.9418	0.8567	0.9115	0.8638	0.9669	0.9696	0.9699	0.9414

In accordance with the probabilities shown in Table 6.17 for $\delta\vec{\theta}$ -space, there is a positive change in the behavior of estimated health parameter deviations $\delta\hat{\vec{\theta}}^*$. Figure 6.6 shows patterns $\delta\hat{\vec{\theta}}^*$ of the same pair of single classes (G$_C$ and G$_{PT}$) addressed in Figure 6.4 but with reduced estimation errors. Now the problems found before are reduced: the fault developments are more similar to the noise-free trajectories, there is a clear reduction of error scatter, and the severity limit for the classes is better estimated. Therefore, the combined effect of using real noise distribution and considering more information for the estimation process in the form of multi-point operation lead to a significant reduction in estimation errors.

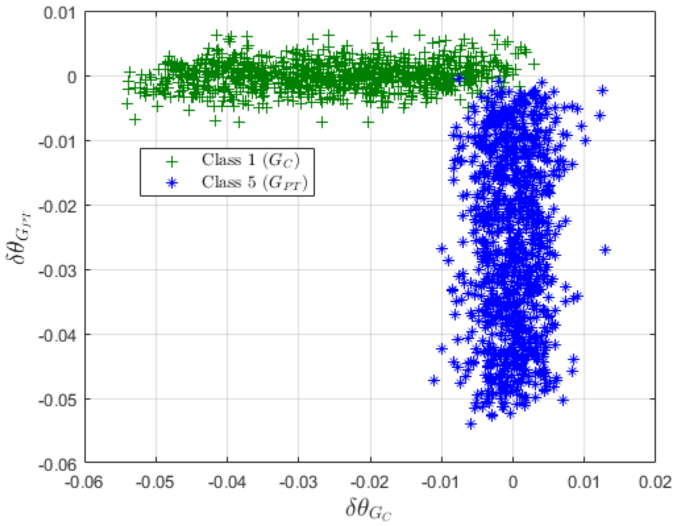


Figure 6.6 Classes of patterns $\delta\hat{\vec{\theta}}^*$ with reduced estimation errors (trajectories along the axes, less scattered noise, and adequate severity limits.

Preliminary conclusions

Let us give some explanations about the level of final accuracy obtained in each case study. Table 6.18 reunites the results of final diagnosis accuracy $\overline{P}_{av}$ as well as the differences between approaches (diff=$\overline{P}_{av\delta\theta}-\overline{P}_{av\delta Y}$) for the five cases. As can be observed, in Case 1, probabilities in $\delta\vec{Y}$ -space and $\delta\vec{\theta}$ -space are around 84.5%. This result is acceptable considering that there are six singular classes intersecting each other in the center. For Case 2, there is visible common reduction of about 12% (probabilities around 72.5%) taking Case 1 as a reference caused by the increase of fault classification complexity, leading to more intersections among all classes and making the fault recognition more difficult as a consequence. However, for Case 3, there exist a diagnosis accuracy increase up to 13.5% for $\delta\vec{Y}$ -space (recognition of 86.15%) and 14.6% for $\delta\vec{\theta}$ -space (recognition of 87%) with respect to Case 2 despite maintaining the same structure of a complex classification. This probability increase that slightly favors more to $\delta\vec{\theta}$ -space in $\overline{P}_{av}$ can be explained by the use of multi-point diagnosis operation by integrating different gas turbine stationary operating points. For Case 4, probabilities $\overline{P}_{av}$ remain very similar to Case 3 (86.6% and 87.4% for $\delta\vec{Y}$ and $\delta\vec{\theta}$ spaces, respectively) in spite of including a healthy scenario. Since the healthy class is built from actual incipient fault trajectories, the reason of this similarity is because the distribution of the entire fault classification is not modified. For Case 5, there is again an increase of 7.9% for $\delta\vec{Y}$ -space (recognition of 94.5%) and 6.7% for $\delta\vec{\theta}$ -space (recognition of 94.1%) with respect to the previous case. Real noise scheme presents less scattered errors than the simulated ones improving probabilities $\overline{P}_{av}$ as a result. Besides, probabilities in Case 5 have an additional value because they give us a better idea about the level of diagnosis accuracy expected in practice in both approaches.

In all cases, the scheme "early stopping=0" resulted to be the most adequate for training the networks. As a conclusion, the two approaches obtain almost the same recognition accuracy in all cases (there is only an averaged difference of 0.14%).

Table 6.18 Final diagnosis accuracy $\overline{P}_{av}$ for both approaches and all case studies (100 seeds)

Case study	**Space**		**Diff**
	$\delta\vec{\theta}$	$\delta\vec{Y}$	
1	0.8442	0.8469	−0.0027
2	0.7234	0.7259	−0.0025
3	0.8700	0.8615	0.0085
4	0.8740	0.8664	0.0076
5	0.9414	0.9455	−0.0041
Averaged diff			**0.0014**

Case study 6: full classification

Case 6 deals with a full classification representing a hypothetical engine fleet with component faults of all kind. It represents a challenge in recognition for both approaches since it considers all possible combinations of classes with one and up to six fault parameters. The base classification contains sixty-three classes and it is formed by:

- 6 single classes of 1 parameter
- 15 multiple classes of 2 parameters
- 20 multiple classes of 3 parameters

- 15 multiple classes of 4 parameters
- 6 multiple classes of 5 parameters
- 1 multiple class of 6 parameters

Figure 6.7 gives an example of the full classification in the $\delta\vec{Y}$ -space. Four variations of the full classification are performed and listed in Table 6.19 with their characteristics. As before, we analyze the influence of the fault classification, the type of diagnostic analysis (one-point or multi-point), and the deviation noise scheme (simulated or real).

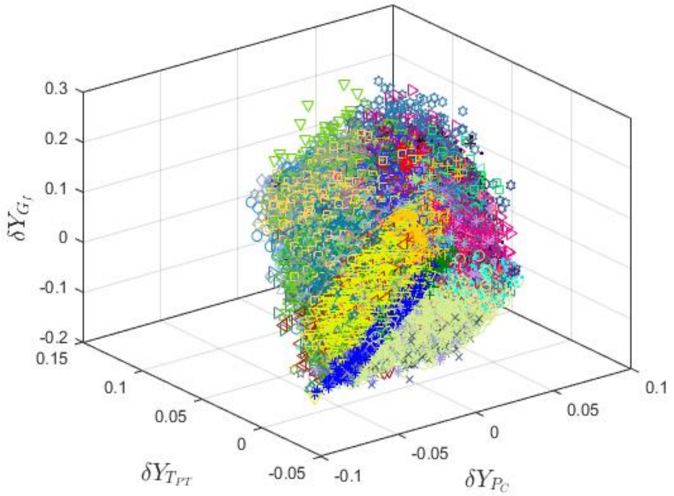


Figure 6.7 Example of full classification (63 fault classes) in $\delta\vec{Y}$ -space.

Table 6.19 Variation of the full classification to evaluate both approaches.

Variation	Number of classes	Healthy Class	Diagnostic analysis	Deviation noise
1	63	No	One-point*	Simulated
2	63	No	Multi-point	Simulated
3	64	Yes	Multi-point	Simulated
4	64	Yes	Multi-point	Real

*Maximal engine regime (n_C=10700)

Table 6.20 contains the results of such variations. Due to the high complexity in the full classification, the training execution time increases significantly. For that reason, probabilities of only ten seeds are computed and averaged ($\overline{P}_{av}$) for all variations. Before computing $\overline{P}_{av}$, the optimal parameters using "early stopping=0" were determined for each network. For Variation 1, the probabilities for both spaces are very similar (around 30%). The level of recognition of this variation is totally normal since the networks cannot correctly classify the patterns because there is a great level of intersection between all classes, especially at the center (see Figure 6.7). For Variation 2, the probabilities increase about 19% for both approaches (a recognition value of 48.7% and 49.6% for $\delta\vec{Y}$ and $\delta\vec{\theta}$ spaces, respectively) as before due to the implementation of multi-point analysis. Variation 3 maintains similar level of accuracy with respect to Variation 2 despite the inclusion of the healthy class (49.4% for δY -space and 50.2% for $\delta\vec{\theta}$ -space). For variation 4, an important increase occurs: 25.2% for $\delta\vec{Y}$ -space (a recognition level of 74.6%) and 21% for $\delta\vec{\theta}$ -

space (a recognition level of 71.3%) with respect to Variation 3. This improvement is produced by the use of the real noise scheme as discussed before.

Both approaches reach an acceptable accuracy despite the high complexity of the full classification. However, it is observable that there is a considerable difference of 3.3% between approaches for this last variation. Despite this fact, the averaged difference in all variations is just about −0.45%.

Table 6.20 Final diagnosis accuracy $\overline{P}_{av}$ for both approaches and all variations of full classification (10 seeds).

Variation	**Space**		**Diff**
	$\delta\vec{\theta}$	$\delta\vec{Y}$	
1	0.3066	0.3091	−0.0025
2	0.4965	0.4870	0.0095
3	0.5028	0.4948	0.0080
4	0.7137	0.7468	−0.0331
Averaged diff			−0.0045

6.4 COMPARISON OF APPROACHES USING GT2

6.4.1 Comparison conditions

Selected variables

The process of comparison of the two approaches is repeated for GT2. The monitored variables and simulated fault parameters employed for diagnostics are listed in

Table 6.21 and Table 6.22, respectively. Three operating points are used and set by the relative gas generator spool speeds (n_{Crel}=1, 0.95 and 0.90) and standard atmospheric conditions. The influence matrices for these regimes are shown in Table 6.23.

Table 6.21 Monitored variables (Vector $\vec{Y}$) of GT2.

ID	**Description**	**Symbol**
1	Compressor exit pressure	P3
2	Compressor exit temperature	T3
3	High pressure turbine exit pressure	P44
4	High pressure turbine exit temperature	T44
5	Power turbine exit pressure	P5
6	Power turbine exit temperature	T5
7	Fuel flow	Gf

Table 6.22 Simulated fault parameters (Vector $\vec{\theta}$) for GT2

ID	**Description**	**Symbol**	**Severity**
1	Compressor flow capacity	FC_C	0 to −5%
2	Compressor efficiency	η_C	0 to −5%
3	HPT flow capacity	FC_{HPT}	0 to −5%
4	HPT efficiency	η_{HPT}	0 to −5%
5	PT flow capacity	FC_{PT}	0 to −5%
6	PT efficiency	η_{PT}	0 to −5%

6.4.2 Cases of comparative analysis

Three case studies are proposed to compare the approaches and are specified in Table 6.24. As can be seen, all the cases include the healthy class, simulated noise scheme and multi-point diagnostic analysis. Figure 6.8 illustrates three fault classes of Case 1 in the $\delta\vec{\theta}$ -space.

Table 6.23 Influence matrices for three regimes of GT2

Relative GG spool speed	**Monitored variables**	**Fault parameters**					
		FC_{C}	**η_{C}**	**FC_{HPT}**	**η_{HPT}**	**FC_{PT}**	**η_{PT}**
1.00 (38422.3 rpm)	**P3**	1.3533	-0.8320	-0.8251	-0.8701	-0.3983	-0.0004
	T3	0.4054	-0.9109	-0.2347	-0.2475	-0.1131	-0.0001
	P44	1.3496	-0.9072	0.2338	-1.1434	-1.4567	-0.0015
	T44	0.7553	-1.8295	0.4119	-2.2975	-1.0499	-0.0012
	P5	0.1274	-0.0940	0.0212	-0.1167	-0.0477	-0.0107
	T5	0.5825	-1.8738	0.4047	-2.3491	-0.8822	-0.3118
	Gf	2.1330	-2.8234	0.6462	-3.5381	-1.6129	-0.0018
0.95 (36501.2 rpm)	**P3**	1.3864	-0.8612	-0.7947	-0.9166	-0.4077	-0.0002
	T3	0.4137	-0.9016	-0.2323	-0.2680	-0.1190	-0.00006
	P44	1.3661	-0.9605	0.2550	-1.2174	-1.4552	-0.0007
	T44	0.8318	-2.0276	0.4049	-2.5487	-1.1269	-0.0005
	P5	0.1065	-0.0800	0.0207	-0.1014	-0.0363	-0.0114
	T5	0.6330	-2.0030	0.3857	-2.5127	-0.9119	-0.2864
	Gf	2.2717	-3.0971	0.6774	-3.8888	-1.7226	-0.0008
0.90 (34580.1 rpm)	**P3**	1.3853	-0.8424	-0.7777	-0.9034	-0.3797	-0.0002
	T3	0.4142	-0.8881	-0.2378	-0.2764	-0.1160	-0.00008
	P44	1.3004	-0.8820	0.2499	-1.1265	-1.3531	-0.0010
	T44	0.8302	-2.0645	0.3607	-2.5995	-1.0889	-0.0008
	P5	0.0871	-0.0643	0.0169	-0.0818	-0.0279	-0.0085
	T5	0.6110	-2.0139	0.3299	-2.5298	-0.8534	-0.2581
	Gf	2.2726	-3.1468	0.6468	-3.9670	-1.6591	-0.0012

Table 6.24 Cases studies proposed to evaluate both approaches using GT1.

Case study	**Diagnostic analysis**	**Deviation noise**	**Fault classes**						
1	Multi-point	Simulated	No-fault class	δG_{C}	$\delta\eta_{C}$	δG_{HPT}	$\delta\eta_{HPT}$	δG_{PT}	$\delta\eta_{PT}$
2	Multi-point	Simulated	No-fault class	δG_{C} $\delta\eta_{C}$	δG_{HPT} $\delta\eta_{HPT}$	δG_{PT} $\delta\eta_{PT}$			
3	Multi-point	Simulated	No-fault class + 63 fault classes						

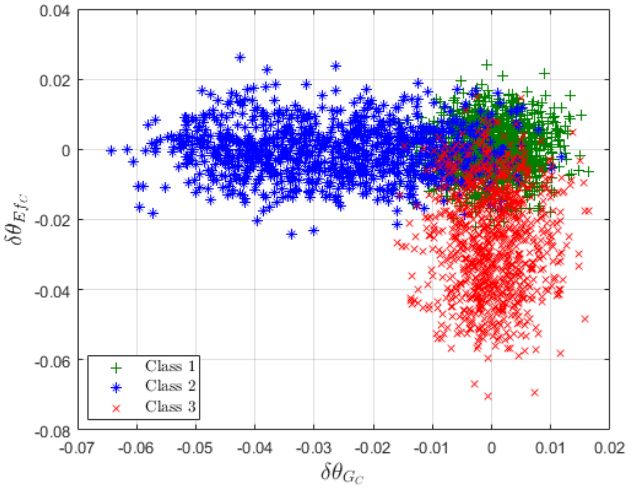


Figure 6.8 Three singular fault classes of Case 1 in $\delta\vec{\theta}$ -space (GT2)

Tuning

The tuning computations to search optimal parameters (hidden neurons and epochs) using the early stopping training option for all cases are shown in Table 6.25. Figure 6.9 and Figure 6.10 display the tuning in $\delta\vec{\theta}$ -space for Case 1 and Case 3, respectively. After finding the optimal parameters, averaged confusion matrices were computed for all cases working with 100 or 10 seeds. Table 6.26, Table 6.27 and Figure 6.11 show the true classification rates (diagonal elements of confusion matrix) for the three cases and both approaches. For Case 1, it is worth noting that the low probability of D6 is related to the small influence on all monitored variables because D6 and the no-fault class are contained in a close region causing an incorrect classification for D6. For Case 3, the true classification rate of the classes with more complexity decreases (see the last classes in Figure 6.11). Table 6.28 shows averaged probabilities $\overline{P}_{av}$ as well as the differences between approaches (diff= $\overline{P}_{av\delta\theta} - \overline{P}_{av\delta Y}$) for the three cases. The high probabilities for Case 1 and 2 are due to the use of simple fault classes and a multiple-point diagnostic analysis. Case 3 presents low recognition probabilities because of the high complexity of the full classification. However, the approaches obtain very similar recognition accuracy (a general difference of 0.38%) as in the case of GT1 confirming that any approach is good option for diagnostics.

Table 6.25 Results of tuning for all case studies (1 seed)

Case study	**Diagnostic space**	**Optimal Parameters**	$\overline{P}$

1	$\delta\vec{Y}$	Neurons=36
		Epochs=200
			0.9213

	$\delta\vec{\theta}$	Neurons=42
		Epochs=200
			0.9260

2	$\delta\vec{Y}$	Neurons=45
		Epochs=200
			0.9733

	$\delta\vec{\theta}$	Neurons=78
		Epochs=200
			0.9766

3	$\delta\vec{Y}$	Neurons=70
		Epochs=6000
			0.4416

	$\delta\vec{\theta}$	Neurons=50
		Epochs=6000
			0.4465

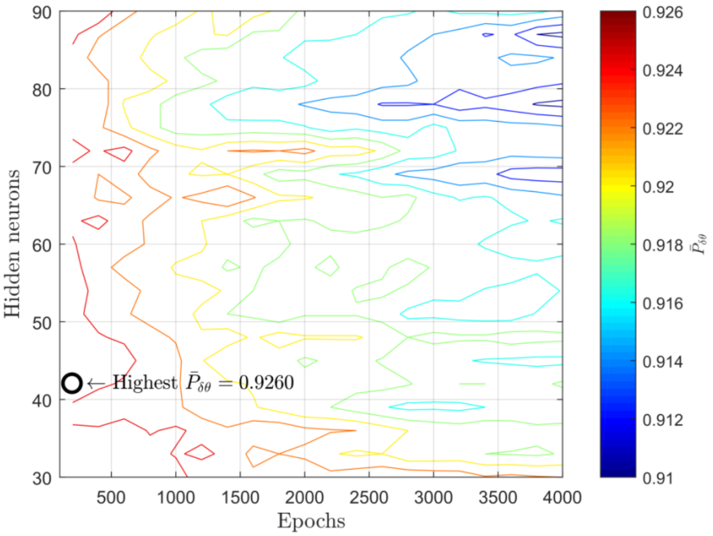


Figure 6.9 Tuning for Case 1 ($\delta\vec{\theta}$ -space)

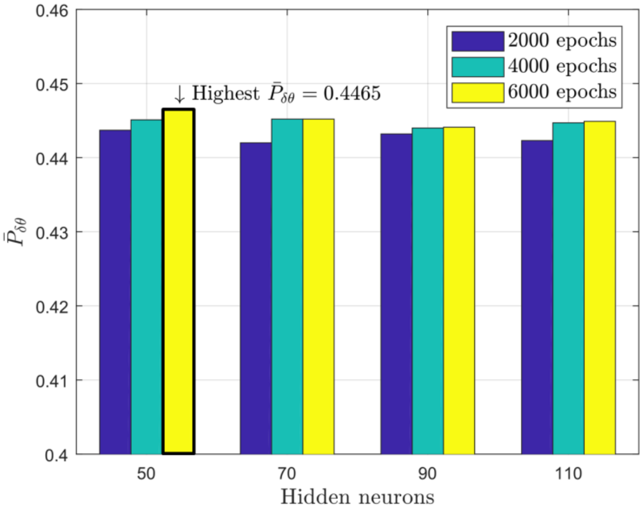


Figure 6.10 Tuning for Case 3 ($\delta\vec{\theta}$ -space)

Table 6.26 True classification rates for Case 1 (100 seeds)

Space	No-fault	D1	D2	D3	D4	D5	D6
$\delta\vec{Y}$	0.9084	0.9857	0.8981	0.9675	0.8929	0.9630	0.4077
$\delta\vec{\theta}$	0.9212	0.9865	0.9058	0.9692	0.8977	0.9643	0.4083

Table 6.27 True classification rates for Case 2 (100 seeds)

Space	No-fault	D1	D2	D3
$\delta\vec{Y}$	0.5458	0.9784	0.9794	0.9737
$\delta\vec{\theta}$	0.5239	0.9803	0.9826	0.9810

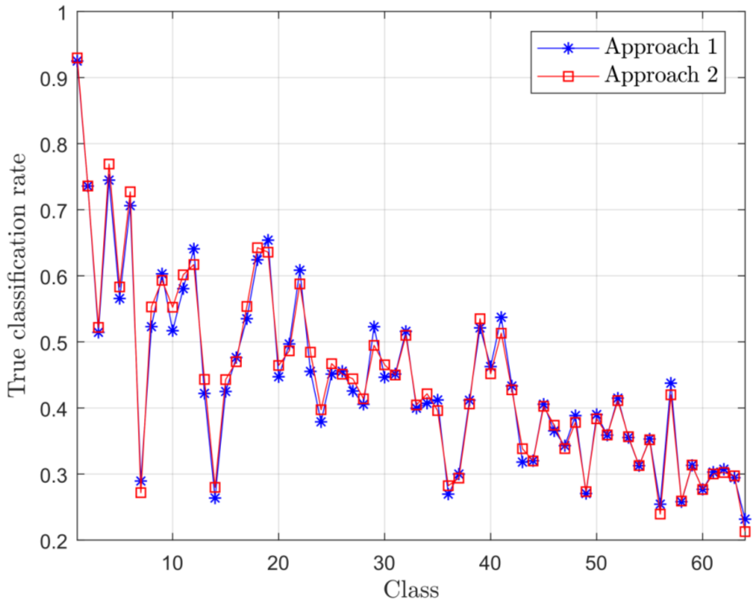


Figure 6.11 True classification rates for Case 3 (10 seeds).

Table 6.28 Final diagnosis accuracy $\overline{P}_{av}$ for both approaches and all case studies using GT2 (10 seeds)

Case study	**Space**		**Diff**
	$\delta\vec{\theta}$	$\delta\vec{Y}$	
1	0.9218	0.9167	0.0051
2	0.9726	0.9690	0.0036
3	0.4480	0.4452	0.0028
Averaged diff			**0.0038**

Chapter 7: Thesis Summary, Contributions and Future Works

7.1 DEVELOPMENT OF AN ONLINE MONITORING ALGORITHM FOR ESTIMATING UNMEASURED GAS TURBINE VARIABLES

Chapter 3 proposed a new method to estimate important unmeasured GTE quantities (engine power or thrust, etc.) using for all of them a common data-driven approach. Each quantity was presented as a function of measured operating conditions and monitored variables. Such a function structure allowed taking into account possible engine performance deterioration and faults. It was also proposed to calculate and monitor the deviations of estimated quantities and to determine the necessary baseline model within the same data-driven approach. Healthy and faulty engine data to create the data-driven models of unmeasured variables were generated using a thermodynamic model at 270 operating points for healthy and 12 faulty engine conditions. The data-driven models were identified using these data and the least-squares method. These data-driven models enabled computing unmeasured quantities and their deviations through measured variables. For such tasks, three algorithms were proposed and tested. As shown in each algorithm, from the point of accuracy there are no limitations for using the estimations of unmeasured engine variables and their deviations for gas turbine diagnosis.

Let us now discuss the issue of prospective use and utility of these quantities for an on-line monitoring system. Gas turbine diagnostics usually relies on physics-based models. However, such models have intrinsic errors, are not always available, or are not easy-to-use in an on-line monitoring system. To avoid these difficulties, a simple on-line monitoring system could be elaborated on the existing information without the physics-based models. For fault detection, for example, tracking the deviations of measured variables can be applied. For fault identification algorithms based on pattern classification, sensor fault classes are easily created without physics-based simulation. However, a representative gas path fault classification cannot be built because only few gas path fault classes can be formed using real data for the deterioration mechanisms that frequently occur in practice, for instance, compressor fouling and inlet filter clogging for stationary power plants. Since the diagnostic capabilities of the on-line system built without the thermodynamic model are limited, it would be beneficial to use the estimates of important unmeasured variables to extend these capabilities.

So far, GTE unmeasured variables are estimated by using physics-based models or other thermodynamics-based approaches that are individual for every variable. In contrast, the technique proposed in this chapter provides a universal and simple data-driven mode to compute these variables. The study involves the thermodynamic model only as a source of data to validate the technique; in future applications it is supposed to use only real data. In addition, the investigation proposes to compute and monitor the deviations of unmeasured variables using a baseline. When a number of real data-based classes and the components diagnosed are small, the use of compressor and turbine efficiency deviations drastically would extend the diagnostic capabilities of the monitoring system because great deviation values indicate faulty components. The deviations of other important unmeasured variables, such as output power or thrust and a combustion chamber temperature could also be useful. They determine an engine capability to perform the required task, overall engine efficiency, and engine integrity.

The main contribution of this investigation is to prove the accuracy of the developed algorithm and its suitability for on-line monitoring systems. The high accuracy of estimations of unmeasured GTE variables and their deviations has been proven for a wide range of engine fault scenarios and numerous differing operating conditions, thus confirming the readiness of the proposed methodology for the implementation in real on-line monitoring systems.

7.2 EVALUATION OF GAS TURBINE DIAGNOSTIC TECHNIQUES UNDER VARIABLE FAULT CONDITIONS

The aim of the methodology presented in Chapter 4 was to evaluate gas path diagnostic techniques using a principle of variable structure classification applied to cover possible fault scenarios in gas turbine maintenance. This principle allowed us to create more versatile and realistic fault conditions relative to existing studies such as complex fault classifications, a new boundary for fault severity and real deviation errors. The techniques analyzed were included into a special procedure that repeated a diagnostic process many times and computed for each fault class a probability of correct diagnosis. Using this probability averaged for all the classes as the evaluation criterion, the techniques were tested under the conditions of four comparative studies. The following explanations summarize the main contributions of Chapter 4:

1) For variable gas turbine fault conditions, any technique presented in this investigation can be an efficient option. Although SVMs produced better results for all the comparative studies analyzed, the difference between all the techniques in terms of $\bar{P}$ is not so great. Furthermore, the similar probabilities prove that the methods are near the theoretical accuracy levels intrinsically related to the engine and the type of fault classification studied and the four techniques are advanced enough to correctly perform the recognition task. Thus, no other technique will significantly increase the probability of correct classification.

Since the evaluation criterion $\bar{P}$ may not be sufficient to select an appropriate technique, there are other important aspects to take into consideration. In the case of MLP, the existence of local minima complicated considerably the training stage. Besides, it has more parameters to tune, for example, the number of hidden neurons, the number of iterations, the training goal and the parameters of the backpropagation method selected. The advantage of MLP is its easiness to implement. As for RBN, it only needs two parameters to be tuned; however, many calculations must be done. Furthermore, it requires much more computational resources and time for training making this technique the least recommended for real training stages. PNN is the simplest gas turbine fault recognition technique and only needs one parameter to tune (σ) resulting in a faster training stage. Also, it has the important advantage of providing confidence estimations for every diagnostic decision which make it a very good option for real monitoring systems. One disadvantage of PNN is the need of more computational resources to store the model when the number of patterns is increased. SVM is the technique that achieves slightly better results and only needs two parameters to be tuned (σ and C). Also, it is not limited by the computational memory supporting more number of patterns. However, one disadvantage of SVM is that its training can be very slow compared to the rest of techniques when the data is increased.

2) Based on the principle of variable classification, the results obtained for all the comparative studies confirm that there is a great influence of the fault classifications on the diagnosis accuracy levels. Thus, the study gives an idea on how the theoretical accuracy levels behave for different gas

turbine fault identification conditions, serving as a help in the decisions of real engine monitoring designers. The principle allows us to create different classifications with necessary totality of fault classes of different type and complexity. The formation of each new classification and change from one classification to another one is simple and do not need to reprogram the algorithm. In general, the diagnosis probabilities generated for all the classifications are acceptable taking into account that the classes are more complex. This complexity can be seen, for example in classification 4, where there are up to 18 classes and most of them intersect in the center. Also, the increasing number of fault parameters to form multiple classes and other characteristics such as the number of patterns per class, the fault development directions and the type of fault class complicate the recognition task for all the techniques.

3) There is an important effect of fault severity boundary on the probabilities of correct diagnosis. With these results, the new boundary makes the simulation more realistic and allows determining more precisely the level of diagnostic accuracy. The fault severity limits of the multiple classes are smoother, which could be the behavior in real faulty conditions. For this reason, the new boundary is advisable for future works.

4) The use of real deviation noise in fault class description provides more accurate simulation of a diagnostic process and provides more reliable level of diagnostic accuracy. This real noise scheme significantly changed the final diagnosis accuracy in all the fault classifications and all the techniques as well.

Some proposed works in the near future may involve: new techniques based on extreme learning and spare representation for fault recognition [143,144]; analysis of measurement inaccuracy and deviation error reduction; novel signal processing approaches for gas turbine diagnostics; different distributions employed to describe random fault severity; and mixed data-driven and model based fault classification.

7.3 DEVELOPMENT OF A DATA-DRIVEN GAS TURBINE DIAGNOSTIC ALGORITHM USING THE PRODIMES SOFTWARE

The objective of Chapter 5 was to develop and examine a gas turbine diagnostic algorithm based on pattern recognition techniques through the ProDiMES software. The data generation using the ProDiMES allowed the simulation of healthy and faulty conditions of an engine fleet in an appropriate environment facilitating the diagnostic process. Before the final comparison with the ProDiMES example solution and other known diagnostic developments, all of the algorithm steps were carefully optimized. In particular, three pattern recognition techniques were tuned and compared. Optimization of the proposed diagnostic algorithm is a complex process not limited by tuning of the pattern recognition techniques employed. For the sake of higher algorithm performances, it was necessary to take into consideration all peculiarities of the ProDiMES methodology and optimize all algorithm steps, namely:

- Choose a proper reference set to determine a baseline model;
- Select the best variation of the baseline model;
- Apply smoothing of measurement deviations from the baseline;
- Optimize the structure of a fault classification constructed from the smoothed deviations (patterns);
- Tune each of the pattern recognition technique;
- Choose the best technique.

As a result, a mean probability of correct classification increased in two times in comparison with other previous results [142], and the other diagnostic performances of the algorithm became higher or comparable to those of known diagnostic solutions making the algorithm a promising tool for real gas turbine monitoring systems. In addition to pretty high accuracy performances, the algorithm has potential advantages of operating without a physics-based model and providing a confidence measure to every diagnostic decision. In general, the presented pattern recognition-based approach seems to be a promising tool for a gas turbine monitoring system. However, the studies have shown that the ProDiMES software allows further optimizing and extending this approach.

As a future work we are going to address the following issues. First, for better algorithms comparison and in accordance with the ProDiMES manual, TNR should be increased up to 0.999 by further algorithm's adjustment. Second, the algorithm under analysis includes three diagnostic stages: feature extraction, fault detection, and fault identification. Working with ProDiMES, we have found that the fourth stage of lifetime prediction can also be included in the algorithm and verified using this software. Third, as shown in Fig. 4, the deviations and their errors grow along with operating time because of normal engine deterioration. Hence, fault diagnosis on the basis of these deviations will worsen. In this way, we need to verify the algorithm during all the lifetime, i.e. develop and examine a "'moving" algorithm. The ProDiMES software allows generating the data for such a complex examination.

7.4 COMPARATIVE STUDY OF PHYSICS-BASED AND DATA-DRIVEN GAS TURBINE DIAGNOSTIC APPROACHES

Chapter 6 proposed a methodology for comparing two diagnostic approaches: one based on the deviations of monitored variables (data-driven) and another one based on fault parameters deltas (physics-based). For comparison purposes, the deviations of both approaches were generated using the same data from two validated engine thermodynamic models. Each approach contained the same steps: network training stage using a learning set, network simulation using a validation set, and the computation of a probability of correct diagnosis for total engine diagnosability. This was a criterion for comparing the two approaches.

The contribution of the study relies on the following points:

- The performance of the two approaches in terms of diagnosis accuracy has not been compared before.
- Different complex fault scenarios and conditions have been not tested yet for both approaches.
- The use of estimated health parameter deviations in gas turbine fault recognition problems is not commonly found in the literature.

The results showed that for the case studies proposed using two test case engines, the diagnostic approaches have very similar diagnosis accuracies despite increasing the fault classification complexity in each case. In this way, the two approaches adjusted to such scenarios and achieved competitive recognition accuracies. The use of multi-mode operation and real deviation errors reduce estimation errors and help to significantly increase the diagnosis accuracy in both approaches.

According to the implemented procedure, there is only one step that is different in both approaches, i.e. the estimation of health parameter deviations. The rest is considered as fixed and common for the two methodologies. This allowed us to have a fair comparison and make more sound conclusions. Also, the results can give an idea on the level of diagnosis accuracy expected in practice.

With the inclusion of a healthy class from current fault classes, the methodology has the advantage of working with both anomaly detection (monitoring) and fault identification (diagnostics) as a joint process. Both stages are independent but the recognition technique employed, i.e. MLP, can compute diagnosis indices for incipient and developed faults at the same time. This characteristic can be exploited by designers of performance monitoring systems.

It was analyzed that the procedure in both diagnostic approaches can also detect small faults that present low influence on monitored variables with a good level of diagnosis accuracy in the background of complex fault classification.

The preliminary analysis of optimal network configurations in all cases ensured a correct and fair comparison of the approaches. The iterations used in the early stopping=1" scheme are not sufficient to completely avoid underfitting when fault classification becomes more complex. It is evident that for "early stopping=0" the training time required for optimizing and validating the network is elevated because two parameters (epochs and neurons) need to be adjusted instead of only one (neurons) when "early stopping=1" is used. However, a significant increase of nearly 1% in diagnosis accuracy can be achieved through "early stopping=0". For that reason, this last option was the most adequate to compute the final recognition task in each case study and each diagnostic space. Although the training stage requires a considerable amount of time, the optimal network is trained once and can be stored to recognize further new fault scenarios. The "early stopping=1" scheme may be helpful only to compute preliminary results when simpler classifications are used and take advantage of its faster training time.

The continuous algorithm improvements of artificial neural network algorithms toolboxes in Matlab are also of great consideration in our final recognition results. For example, the use of the recent implemented cross-entropy error function to control network training performance instead of the well-known mean squared error shows fewer misclassifications and thus an improvement in diagnosis accuracy [139]. For that reason, the employment of such enhanced ANN algorithms allows a more reliable gas turbine diagnostic algorithm development.

Future works involve: the use of other techniques to estimate health parameter deviations such as Artificial Neural Networks and Genetic Algorithm; a testing of the proposed algorithm with ProDiMES by using the influence matrices available for take-off and cruise regimes; the use of recent hybrid advanced machine learning and pattern recognition techniques that can be easily implemented in the algorithms for diagnostics in both approaches.

References

[1] Boyce MP. Gas turbine engineering handbook. 4th ed. Gulf Professional Publishing; 2011.

[2] Stamatis AG. Engine Condition Monitoring and Diagnostics. Progress in Gas Turbine Performance, InTech; 2013, p. 187–212. doi:10.5772/54409.

[3] Hanachi H, Mechefske C, Liu J, Banerjee A, Chen Y. Performance-Based Gas Turbine Health Monitoring, Diagnostics, and Prognostics: A Survey. IEEE Transactions on Reliability 2018;67:1340–63. doi:10.1109/TR.2018.2822702.

[4] Meher-Homji CB, Chaker MA, Motiwala HM. Gas Turbine Performance Deterioration. 30th Turbomachinery Symposium, Texas A&M University, Houston, Texas, USA: 2001, p. 139–76. doi:10.21423/R19Q1P.

[5] Diakunchak IS. Performance Deterioration in Industrial Gas Turbines. ASME 1991 International Gas Turbine and Aeroengine Congress and Exposition, Orlando, Florida, USA: 1991, p. V004T10A012. doi:10.1115/91-GT-228.

[6] Zaita A V., Buley G, Karlsons G. Performance Deterioration Modeling in Aircraft Gas Turbine Engines. ASME 1997 International Gas Turbine and Aeroengine Congress and Exhibition, Orlando, Florida, USA: 1997, p. V004T16A007. doi:10.1115/97-GT-278.

[7] Kurz R, Brun K. Degradation in Gas Turbine Systems. Journal of Engineering for Gas Turbines and Power 2001;123:70–7. doi:10.1115/1.1340629.

[8] Kurz R, Meher-Homji CB, Brun K. Gas Turbine Degradation. 43rd Turbomachinery and 30th Pump Users Symposia (Pump & Turbo 2014), Houston, Texas, USA: 2014, p. 1–36. doi:10.21423/R15W5P.

[9] Meher-Homji CB, Bromley A. Gas Turbine Axial Compressor Fouling and Washing. Proceedings of the Thirty Third Turbomachinery Symposium, Texas A&M University, Houston, Texas, USA: 2004, p. 163–92. doi:10.21423/R1S66R.

[10] Burnes D, Kurz R. Performance Degradation Effects in Modern Industrial Gas Turbines. Proceedings of Zurich 2018 Global Power and Propulsion Forum, Zurich, Switzerland: 2018, p. 1–9.

[11] Marson E. Effect of Manufacturing Deviations on Performance of Axial Flow Compressor Blading. ASME 1992 International Gas Turbine and Aeroengine Congress and Exposition, Cologne, Germany: 1992, p. V001T01A119. doi:10.1115/92-GT-326.

[12] Zhao N, Wen X, Li S. A Review on Gas Turbine Anomaly Detection for Implementing Health Management. ASME Turbo Expo 2016, Seoul, South Korea: 2016, p. V001T22A009. doi:10.1115/GT2016-58135.

[13] Vachtsevanos G, Lewis F, Roemer M, Hess A, Wu B. Intelligent Fault Diagnosis and Prognosis for Engineering Systems. Hoboken, NJ, USA: John Wiley & Sons, Inc.; 2006. doi:10.1002/9780470117842.

[14] Rao BKN. Handbook of Condition Monitoring. 1st ed. Oxford, UK: Elsevier Advanced Technology; 1996.

[15] Jardine AKS, Tsang AHC. Maintenance, Replacement, and Reliability: Theory and Applications. 2nd Editio. CRC Press; 2017.

[16] Kelly A. Maintenance and Its Management. Conference Communication; 1989.

[17] Li YG. Performance-analysis-based gas turbine diagnostics: a review. Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy 2002;216:363–77. doi:10.1243/095765002320877856.

[18] Marinai L, Probert D, Singh R. Prospects for Aero Gas-Turbine Diagnostics: a Review. Applied Energy 2004;79:109–26. doi:10.1016/J.APENERGY.2003.10.005.

[19] Loboda I. Gas Turbine Condition Monitoring and Diagnostics. Gas Turbines, InTechOpen; 2010, p. 119–44. doi:10.5772/10210.

[20] Loboda I. Gas Turbine Diagnostics. Efficiency, Performance and Robustness of Gas Turbines, InTech; 2012, p. 191–212. doi:10.5772/38185.

[21] Loboda I, Olivares Robles MA. Gas Turbine Fault Diagnosis Using Probabilistic Neural Networks.

International Journal of Turbo {&} Jet-Engines 2015;32. doi:10.1515/tjj-2014-0019.

[22] Volponi AJ. Foundations of Gas Path Analysis I. Gas Turbine Condition Monitoring & Fault Diagnosis, Von Karman Institute for Fluid Dynamics; 2003.

[23] Volponi AJ, DePold H, Ganguli R, Daguang C. The Use of Kalman Filter and Neural Network Methodologies in Gas Turbine Performance Diagnostics: A Comparative Study. Journal of Engineering for Gas Turbines and Power 2003;125:917–24. doi:10.1115/1.1419016.

[24] Romessis C, Mathioudakis K. Bayesian Network Approach for Gas Path Fault Diagnosis. Journal of Engineering for Gas Turbines and Power 2006;128:64. doi:10.1115/1.1924536.

[25] Kamboukos P, Mathioudakis K. Multipoint Non-Linear Method for Enhanced Component and Sensor Malfunction Diagnosis. ASME Turbo Expo 2006, Barcelona, Spain: 2006, p. 619–27. doi:10.1115/GT2006-90451.

[26] Jaw LC, Lee Y-J. Engine Diagnostics in the Eyes of Machine Learning. ASME Turbo Expo 2014: Turbine Technical Conference and Exposition, Düsseldorf, Germany: 2014, p. 8. doi:10.1115/GT2014-27088.

[27] Borguet S, Léonard O, Dewallef P. Regression-Based Modelling of a Fleet of Gas Turbine Engines for Performance Trending. ASME Turbo Expo 2015, Montreal, Quebec, Canada: 2015, p. V006T05A004. doi:10.1115/GT2015-42330.

[28] Aretakis N, Roumeliotis I, Alexiou A, Romesis C, Mathioudakis K. Turbofan Engine Health Assessment From Flight Data. Journal of Engineering for Gas Turbines and Power 2014;137:8. doi:10.1115/1.4028566.

[29] Simon DL, Borguet S, Léonard O, Zhang X (Frank). Aircraft Engine Gas Path Diagnostic Methods: Public Benchmarking Results. ASME Turbo Expo 2013, San Antonio, Texas, USA: 2013, p. V004T06A014. doi:10.1115/GT2013-95077.

[30] Joly RB, Ogaji SOT, Singh R, Probert SD. Gas-Turbine Diagnostics Using Artificial Neural-Networks for a High Bypass Ratio Military Turbofan Engine. Applied Energy 2004;78:397–418. doi:10.1016/j.apenergy.2003.10.002.

[31] Jiang X, Mendoza E, Lin T. Bayesian Calibration for Power Splitting in Single Shaft Combined Cycle Plant Diagnostics. ASME Turbo Expo 2015, Montreal, Quebec, Canada: 2015, p. V006T05A029. doi:10.1115/GT2015-43878.

[32] DeCastro JA, Frederick DK, Tang L. Engine Parameter Estimation in Test Cells Using Hybrid Physics/Empirical Models. ASME Turbo Expo 2011, Vancouver, British Columbia, Canada: ASME; 2011, p. 169–76. doi:10.1115/GT2011-45633.

[33] Palmer C, Hettler E. Thrust Measurement Model-Based Correction System for Turbine Engine Test Cell Dynamic Data. ASME Turbo Expo 2015, Montreal, Quebec, Canada: 2015, p. V006T05A025. doi:10.1115/GT2015-43720.

[34] Maravilla Herrera C, Yepifanov S, Loboda I. Improved Turbine Blade Lifetime Prediction. ASME Turbo Expo 2015, Montreal, Quebec, Canada: 2015, p. V006T05A019. doi:10.1115/GT2015-43046.

[35] Agrawal RK, MacIsaac BD, Saravanamuttoo HIH. An Analysis Procedure for the Validation of On-Site Performance Measurements of Gas Turbines. Journal of Engineering for Power 1979;101:405–14. doi:10.1115/1.3446593.

[36] Kacprzynski GJ, Gumina M, Roemer MJ, Caguiat DE, Galie TR, McGroarty JJ. A Prognostic Modeling Approach for Predicting Recurring Maintenance for Shipboard Propulsion Systems. ASME Turbo Expo 2001, New Orleans, Louisiana, USA,: ASME; 2001, p. V001T02A003. doi:10.1115/2001-GT-0218.

[37] Cortés O, Urquiza G, Hernández JA. Optimization of operating conditions for compressor performance by means of neural network inverse. Applied Energy 2009;86:2487–93. doi:10.1016/J.APENERGY.2009.03.001.

[38] Sieverding CH, Mathioudakis K. Gas Turbine Condition Monitoring & Fault Diagnosis. VKI Lecture Series, Von Karman Institute for Fluid Dynamics; 2003.

[39] Saravanamuttoo HIH, MacIsaac BD. Thermodynamic Models for Pipeline Gas Turbine Diagnostics. Journal of Engineering for Power 1983;105:875–84. doi:10.1115/1.3227496.

[40] Stamatis A, Mathioudakis K, Papailiou KD. Adaptive Simulation of Gas Turbine Performance. Journal of Engineering for Gas Turbines and Power 1990;112:168–75. doi:10.1115/1.2906157.

[41] Tsalavoutas A, Aretakis N, Mathioudakis K, Stamatis A. Combining Advanced Data Analysis Methods for the Constitution of an Integrated Gas Turbine Condition Monitoring and Diagnostic System. ASME Turbo Expo 2000, Munich, Germany: 2000, p. V004T04A006. doi:10.1115/2000-

GT-0034.

[42] Urban LA. Parameter Selection for Multiple Fault Diagnostics of Gas Turbine Engines. Journal of Engineering for Power 1975;97:225–30. doi:10.1115/1.3445969.

[43] Sugiyama N. System Identification of Jet Engines. Journal of Engineering for Gas Turbines and Power 2000;122:19–26. doi:10.1115/1.483172.

[44] Simon DL, Armstrong JB, Garg S. Application of an Optimal Tuner Selection Approach for On-Board Self-Tuning Engine Models. Journal of Engineering for Gas Turbines and Power 2012;134:11. doi:10.1115/1.4004178.

[45] Miro Zarate LA, Loboda I. Computation and Monitoring of the Deviations of Gas Turbine Unmeasured Parameters. ASME Turbo Expo 2015, Montreal, Quebec, Canada: ASME; 2015, p. V006T05A028. doi:10.1115/GT2015-43862.

[46] Sampath S, Singh R. An Integrated Fault Diagnostics Model Using Genetic Algorithm and Neural Networks. Journal of Engineering for Gas Turbines and Power 2006;128:49–56. doi:10.1115/1.1995771.

[47] Butler S, Pattipati K, Volponi A, Hull J, Rajamani R, Siegel J. An Assessment Methodology for Data-Driven and Model-Based Techniques for Engine Health Monitoring. ASME Turbo Expo 2006: Power for Land, Sea, and Air, vol. 2, Barcelona, Spain: 2006, p. 823–31. doi:10.1115/GT2006-91096.

[48] Roemer MJ, Kacprzynski GJ. Advanced Diagnostic and Prognostic Technologies for Gas Turbine Engine Risk Assessment. Volume 4: Manufacturing Materials and Metallurgy; Ceramics; Structures and Dynamics; Controls, Diagnostics and Instrumentation; Education, ASME; 2000, p. V004T04A002. doi:10.1115/2000-GT-0030.

[49] Loboda I, Yepifanov S, Feldshteyn Y. A Generalized Fault Classification for Gas Turbine Diagnostics at Steady States and Transients. Journal of Engineering for Gas Turbines and Power 2007;129:977–85. doi:10.1115/1.2719261.

[50] Loboda I, Feldshteyn Y, Ponomaryov V. Neural Networks for Gas Turbine Fault Identification: Multilayer Perceptron or Radial Basis Network? Int J Turbo Jet-Engines 2012;29:37–48. doi:10.1515/tjj-2012-0005.

[51] Li Y-G. Diagnostics of power setting sensor fault of gas turbine engines using genetic algorithm. The Aeronautical Journal 2017;121:1109–30. doi:10.1017/aer.2017.49.

[52] Ganguli R. Fuzzy Logic Intelligent System for Gas Turbine Module and System Fault Isolation. Journal of Propulsion and Power 2002;18:440–7. doi:10.2514/2.5953.

[53] Ganguli R. Application of Fuzzy Logic for Fault Isolation of Jet Engines. Journal of Engineering for Gas Turbines and Power 2003;125:617–23. doi:10.1115/1.1470481.

[54] Amare FD, Gilani SI, Aklilu BT, Mojahid A. Two-shaft stationary gas turbine engine gas path diagnostics using fuzzy logic. Journal of Mechanical Science and Technology 2017;31:5593–602. doi:10.1007/s12206-017-1053-9.

[55] Ogaji SOT, Marinai L, Sampath S, Singh R, Prober SD. Gas-turbine fault diagnostics: a fuzzy-logic approach. Applied Energy 2005;82:81–9. doi:10.1016/J.APENERGY.2004.07.004.

[56] Biagioni L, Cinotti R, D’Agostino L. Turboshaft Engine Condition Monitoring by Bayesian Identification. 15th Int. Symposium on Air Breathing Engines (ISABE), Bangalore, India: 2001, p. 1–11.

[57] Consumi M, D’Agostino L. Monitoring and Fault Diagnosis of a Turbojet by Bayesian Inference. Thirteenth International Symposium on Air Breathing Engines (ISABE), Chattanooga, Tennessee, USA: 1997, p. 1082–96.

[58] Romesis C, Mathioudakis K. Setting Up of a Probabilistic Neural Network for Sensor Fault Detection Including Operation With Component Faults. Journal of Engineering for Gas Turbines and Power 2003;125:634–41. doi:10.1115/1.1582493.

[59] Cortes C, Vapnik V. Support-Vector Networks. Machine Learning 1995;20:273–97. doi:10.1023/A:1022627411411.

[60] Boser B, Guyon I, Vapnik V. A Training Algorithm for Optimal Margin Classifiers. Proceedings of the fifth annual workshop on Computational learning theory - COLT ’92, Pittsburgh, PA, USA: ACM Press; 1992, p. 144–52. doi:10.1145/130385.130401.

[61] Widodo A, Yang B-S. Support Vector Machine in Machine Condition Monitoring and Fault Diagnosis. Mechanical Systems and Signal Processing 2007;21:2560–74. doi:10.1016/j.ymssp.2006.12.007.

[62] Yin Z, Hou J. Recent Advances on SVM Based Fault Diagnosis and Process Monitoring in

Complicated Industrial Processes. Neurocomputing 2016;174:643–50. doi:10.1016/j.neucom.2015.09.081.

[63] Xu Q, Shi J. Fault Diagnosis for Aero-engine Applying a New Multi-class Support Vector Algorithm. Chinese Journal of Aeronautics 2006;19:175–82. doi:10.1016/S1000-9361(11)60342-7.

[64] Seo D-H, Roh T-S, Choi D-W. Defect diagnostics of gas turbine engine using hybrid SVM-ANN with module system in off-design condition. Journal of Mechanical Science and Technology 2009;23:677–85. doi:10.1007/s12206-008-1120-3.

[65] Lee S-M, Roh T-S, Choi D-W. Defect diagnostics of SUAV gas turbine engine using hybrid SVM-artificial neural network method. Journal of Mechanical Science and Technology 2009;23:559–68. doi:10.1007/s12206-008-1119-9.

[66] Zhou D, Zhang H, Weng S. A New Gas Path Fault Diagnostic Method of Gas Turbine Based on Support Vector Machine. Journal of Engineering for Gas Turbines and Power 2015;137:6 p. doi:10.1115/1.4030277.

[67] Olson DL, Delen D. Advanced Data Mining Techniques. 1st. Springer-Verlag Berlin Heidelberg; 2008. doi:10.1007/978-3-540-76917-0.

[68] Simon DL, Bird J, Davison C, Volponi A, Iverson RE. Benchmarking Gas Path Diagnostic Methods: A Public Approach. ASME Turbo Expo 2008, Berlin, Germany: 2008, p. 325–36. doi:10.1115/GT2008-51360.

[69] Loboda I, Feldshteyn Y. Polynomials and Neural Networks for Gas Turbine Monitoring: a Comparative Study. International Journal of Turbo and Jet Engines 2011;28:227–236. doi:10.1515/tjj.2011.020.

[70] Koskoletos OA, Aretakis N, Alexiou A, Romesis C, Mathioudakis K. Evaluation of Aircraft Engine Diagnostic Methods Through ProDiMES. ASME Turbo Expo 2018, Oslo, Norway: 2018, p. V006T05A023. doi:10.1115/GT2018-76647.

[71] Jaw LC. Recent Advancements in Aircraft Engine Health Management (EHM) Technologies and Recommendations for the Next Step. ASME Turbo Expo 2005, Reno, Nevada, USA: 2005, p. 683–95. doi:10.1115/GT2005-68625.

[72] Simon DL. Propulsion Diagnostic Method Evaluation Strategy (ProDiMES) User's Guide. 2010.

[73] Davison CR, Bird JW. Review of Metrics and Assignment of Confidence Intervals for Health Management of Gas Turbine Engines. ASME Turbo Expo 2008, Berlin, Germany,: 2008, p. 191–201. doi:10.1115/GT2008-50849.

[74] Kobayashi T, Simon DL. Aircraft Engine On-Line Diagnostics Through Dual-Channel Sensor Measurements: Development of an Enhanced System. ASME Turbo Expo 2008: Power for Land, Sea, and Air, Berlin, Germany: 2008, p. 91–103. doi:10.1115/GT2008-50346.

[75] Li YG. Training Future Engineers on Gas Turbine Gas Path Diagnostics Using PYTHIA. Volume 6: Ceramics; Controls, Diagnostics and Instrumentation; Education; Manufacturing Materials and Metallurgy, ASME; 2014, p. V006T08A003. doi:10.1115/GT2014-25856.

[76] Dewallef P, Romessis C, Léonard O, Mathioudakis K. Combining Classification Techniques With Kalman Filters for Aircraft Engine Diagnostics. Journal of Engineering for Gas Turbines and Power 2006;128:281–7. doi:10.1115/1.2056507.

[77] Hanachi H, Liu J, Banerjee A, Chen Y, Koul A. A Physics-Based Performance Indicator for Gas Turbine Engines Under Variable Operating Conditions. ASME Turbo Expo 2014, Düsseldorf, Germany: 2014, p. V006T06A021. doi:10.1115/GT2014-26367.

[78] Hanachi H, Liu J, Mechefske C. Multi-Mode Diagnosis of a Gas Turbine Engine Using an Adaptive Neuro-Fuzzy System. Chinese Journal of Aeronautics 2018;31:1–9. doi:10.1016/J.CJA.2017.11.017.

[79] Hanachi H, Mechefske C, Liu J, Banerjee A, Chen Y. Enhancement of Prognostic Models for Short-Term Degradation of Gas Turbines. 2017 IEEE International Conference on Prognostics and Health Management (ICPHM), Dallas, TX, USA: 2017, p. 66–9. doi:10.1109/ICPHM.2017.7998307.

[80] Hanachi H, Liu J, Banerjee A, Chen Y. Effects of the Intake Air Humidity on the Gas Turbine Performance Monitoring. ASME Turbo Expo 2015, Montreal, Quebec, Canada: 2015, p. V006T05A018. doi:10.1115/GT2015-43026.

[81] Dewallef P, Léonard O. On-Line Performance Monitoring and Engine Diagnostic Using Robust Kalman Filtering Techniques. ASME Turbo Expo 2003, Atlanta, Georgia, USA: 2003, p. 395–403. doi:10.1115/GT2003-38379.

[82] Dewallef P, Léonard O, Mathioudakis K. On-Line Aircraft Engine Diagnostic Using a Soft-Constrained Kalman Filter. ASME Turbo Expo 2004, Vienna, Austria: 2004, p. 585–94.

doi:10.1115/GT2004-53539.

[83] Borguet S, Léonard O. Assessment of an Anomaly Detector for Jet Engine Health Monitoring. International Journal of Rotating Machinery 2011;2011:1–11. doi:10.1155/2011/942576.

[84] Léonard O, Borguet S, Dewallef P. Adaptive Estimation Algorithm for Aircraft Engine Performance Monitoring. Journal of Propulsion and Power 2008;24:763–9. doi:10.2514/1.34320.

[85] Dewallef P, Borguet S. A Methodology to Improve the Robustness of Gas Turbine Engine Performance Monitoring Against Sensor Faults. Journal of Engineering for Gas Turbines and Power 2013;135:7p. doi:10.1115/1.4007976.

[86] Bird JW, Schwartz HM. Diagnosis of Turbine Engine Transient Performance With Model-Based Parameter Estimation Techniques. ASME 1994 International Gas Turbine and Aeroengine Congress and Exposition, The Hague, Netherlands: 1994, p. 8p. doi:10.1115/94-GT-317.

[87] Chalmers JLY, Bird JW, Gauthier D. Design and Validation of an Experiment for the Detection and Prediction of Stall and Surge in a PT6/T400 Turboshaft Engine. ASME Turbo Expo 2007, Montreal, Canada: 2007, p. 1817–26. doi:10.1115/GT2007-28244.

[88] Davison CR. Determination of Steady State Gas Turbine Operation. ASME Turbo Expo 2012, Copenhagen, Denmark: 2012, p. 12p. doi:10.1115/GT2012-68632.

[89] Vatani A, Khorasani K, Meskin N. Health Monitoring and Degradation Prognostics in Gas Turbine Engines Using Dynamic Neural Networks. ASME Turbo Expo 2015, Montreal, Quebec, Canada: 2015, p. 13p. doi:10.1115/GT2015-44101.

[90] Gholamhossein M, Vatani A, Daroogheh N, Khorasani K. Prediction of the Jet Engine Performance Deterioration. ASME 2012 International Mechanical Engineering Congress and Exposition, Houston, Texas, USA: ASME; 2012, p. 359–66. doi:10.1115/IMECE2012-87936.

[91] Daroogheh N, Baniamerian A, Nayyeri H, Khorasani K. Deterioration Detection and Health Monitoring in Aircraft Jet Engines. ASME 2012 International Mechanical Engineering Congress and Exposition, Houston, Texas, USA: 2012, p. 367–75. doi:10.1115/IMECE2012-87938.

[92] Naderi E, Meskin N, Khorasani K. Nonlinear Fault Diagnosis of Jet Engines by Using a Multiple Model-Based Approach. Journal of Engineering for Gas Turbines and Power 2012;134:10p. doi:10.1115/1.4004152.

[93] Daroogheh N, Vatani A, Gholamhossein M, Khorasani K. Engine Life Evaluation Based on a Probabilistic Approach. ASME 2012 International Mechanical Engineering Congress and Exposition, Houston, Texas, USA: 2012, p. 347–58. doi:10.1115/IMECE2012-87929.

[94] Meskin N, Naderi E, Khorasani K. A Multiple Model-Based Approach for Fault Diagnosis of Jet Engines. IEEE Transactions on Control Systems Technology 2013;21:254–62. doi:10.1109/TCST.2011.2177981.

[95] Tsoutsanis E, Meskin N, Benammar M, Khorasani K. Transient Gas Turbine Performance Diagnostics Through Nonlinear Adaptation of Compressor and Turbine Maps. Journal of Engineering for Gas Turbines and Power 2015;137:12p. doi:10.1115/1.4029710.

[96] Tsoutsanis E, Meskin N, Benammar M, Khorasani K. A Dynamic Prognosis Scheme for Flexible Operation of Gas Turbines. Applied Energy 2016;164:686–701. doi:10.1016/J.APENERGY.2015.11.104.

[97] Tsoutsanis E, Meskin N, Benammar M, Khorasani K. An Efficient Component Map Generation Method for Prediction of Gas Turbine Performance. ASME Turbo Expo 2014: Turbine Technical Conference and Exposition, Düsseldorf, Germany: 2014, p. 12p. doi:10.1115/GT2014-25753.

[98] Tahan M, Tsoutsanis E, Muhammad M, Abdul Karim ZA. Performance-based health monitoring, diagnostics and prognostics for condition-based maintenance of gas turbines: A review. Applied Energy 2017;198:122–44. doi:10.1016/J.APENERGY.2017.04.048.

[99] Herrera CM, Yepifanov S, Loboda I. A Comparative Analysis of Turbine Rotor Inlet Temperature Models. ASME 2011 Turbo Expo, Vancouver, British Columbia, Canada: ASME; 2011, p. 317–27. doi:10.1115/GT2011-46161.

[100] Yepifanov S V., Loboda I. Gas Path Model Identification as an Instrument of Gas Turbine Diagnosing. ASME Turbo Expo 2003, Atlanta, Georgia, USA: ASME; 2003, p. 371–6. doi:10.1115/GT2003-38365.

[101] Shevchenko M, Yepifanov S, Loboda I. Ridge Estimation and Principal Component Analysis to Solve an Ill-Conditioned Problem of Estimating Unmeasured Gas Turbine Parameters. ASME Turbo Expo 2013, San Antonio, Texas, USA: ASME; 2013, p. V004T06A008. doi:10.1115/GT2013-94496.

[102] Loboda I, Yepifanov S. On the Selection of an Optimal Pattern Recognition Technique for Gas

Turbine Diagnosis. ASME Turbo Expo 2013, San Antonio, Texas, USA: 2013, p. 11p. doi:10.1115/GT2013-95198.

[103] Cao Y, Hu P, Yang Q, He Y, Li S, Yu F, et al. Fuzzy Analytic Hierarchy Process Evaluation Method of Gas Turbine Based on Health Degree. ASME Turbo Expo 2018, Oslo, Norway: 2018, p. 10p. doi:10.1115/GT2018-75492.

[104] Yang Q, Cao Y, Yu F, Du J, Li S. Health Estimation of Gas Turbine: A Symbolic Linearization Model Approach. ASME Turbo Expo 2017, Charlotte, North Carolina, USA: 2017, p. 12p. doi:10.1115/GT2017-64071.

[105] Cao Y, Yan D, Yang Q, Li S, Wu M, Chen L. Fault Diagnosis of Gas Turbine Based on Complex Networks Theory. ASME Turbo Expo 2017, Charlotte, North Carolina, USA: 2017, p. 10p. doi:10.1115/GT2017-64373.

[106] Cao Y, He Y, Yu F, Du J, Li S, Yang Q, et al. A Two-Layer Multi-Model Gas Path Fault Diagnosis Method. ASME Turbo Expo 2018, Oslo, Norway: 2018, p. 9p. doi:10.1115/GT2018-75368.

[107] Lu J, Lu F, Huang J, Lu J, Lu F, Huang J. Performance Estimation and Fault Diagnosis Based on Levenberg–Marquardt Algorithm for a Turbofan Engine. Energies 2018;11:18p. doi:10.3390/en11010181.

[108] Lu F, Huang J, Xing Y, Lu F, Huang J, Xing Y. Fault Diagnostics for Turbo-Shaft Engine Sensors Based on a Simplified On-Board Model. Sensors 2012;12:11061–76. doi:10.3390/s120811061.

[109] Lu F, Chen Y, Huang J, Zhang D, Liu N. An Integrated Nonlinear Model-Based Approach to Gas Turbine Engine Sensor Fault Diagnostics. Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering 2014;228:2007–21. doi:10.1177/0954410013511596.

[110] Lu F, Huang J, Ji C, Zhang D, Jiao H. Gas Path On-line Fault Diagnostics Using a Nonlinear Integrated Model for Gas Turbine Engines. International Journal of Turbo & Jet-Engines 2014;31:261–75. doi:10.1515/tjj-2014-0001.

[111] Lu F, Wang Y, Huang J, Wang Q. A Comparison of Hybrid Approaches for Turbofan Engine Gas Path Fault Diagnosis. International Journal of Turbo & Jet-Engines 2016;33:253–64. doi:10.1515/tjj-2015-0029.

[112] Lu F, Zheng W, Huang J, Feng M. Life Cycle Performance Estimation and In-Flight Health Monitoring for Gas Turbine Engine. Journal of Dynamic Systems, Measurement, and Control 2016;138:13p. doi:10.1115/1.4033556.

[113] Lu F, Gao T, Huang J, Qiu X. Nonlinear Kalman filters for aircraft engine gas path health estimation with measurement uncertainty. Aerospace Science and Technology 2018;76:126–40. doi:10.1016/J.AST.2018.01.024.

[114] Loboda I, Yepifanov S, Feldshteyn Y. Diagnostic Analysis of Maintenance Data of a Gas Turbine for Driving an Electric Generator. ASME Turbo Expo 2009, vol. 26, Orlando, Florida, USA: ASME; 2009, p. 745–56. doi:10.1515/TJJ.2009.26.4.235.

[115] Loboda I, Yepifanov S, Feldshteyn Y. An Integrated Approach to Gas Turbine Monitoring and Diagnostics. ASME Turbo Expo 2008, Berlin, Germany: ASME; 2008, p. 359–67. doi:10.1115/GT2008-51449.

[116] Loboda I, Pérez-Ruiz JL, Yepifanov S. A Benchmarking Analysis of a Data-Driven Gas Turbine Diagnostic Approach. ASME Turbo Expo 2018, Oslo, Norway: 2018, p. V006T05A027. doi:10.1115/GT2018-76887.

[117] Frederick DK, DeCastro JA, Litt JS. User's Guide for the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS). 2007.

[118] Ganguli R. Gas turbine diagnostics : signal processing and fault isolation. CRC Press; 2013.

[119] Escher PC. Pythia: An object-orientated gas path analysis computer program for general applications. Cranfield University, 1995.

[120] Sanghi V, Lakshmanan BK, Sundararajan V. Survey of Advancements in Jet-Engine Thermodynamic Simulation. Journal of Propulsion and Power 2000;16:797–807. doi:10.2514/2.5644.

[121] Loboda I. Gas Turbine Diagnostic Model Identification on Maintenance Data of Great Volume. Aerospace Technic and Technology Journal 2007:198–204.

[122] Zedda M, Singh R. Gas Turbine Engine and Sensor Fault Diagnosis Using Optimization Techniques. Journal of Propulsion and Power 2002;18:1019–25. doi:10.2514/2.6050.

[123] Urban LA. Gas Path Analysis Applied to Turbine Engine Condition Monitoring. Journal of Aircraft 1973;10:400–6. doi:10.2514/3.60240.

[124] Pérez-Ruiz JL, Loboda I, Miró-Zárate LA, Toledo-Velázquez M, Polupan G. Evaluation of gas

turbine diagnostic techniques under variable fault conditions. Advances in Mechanical Engineering 2017;9:16. doi:10.1177/1687814017727471.

[125] Pérez-Ruiz JL, Loboda I. A Flexible Fault Classification for Gas Turbine Diagnosis. Aerospace Technic and Technology Journal 2014:94–102.

[126] Yu H, Yuecheng Y, Shiying Z, Zhensheng S. Comparison of linear models for gas turbine performance. Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering 2014;228:1291–301. doi:10.1177/0954410013490090.

[127] Kamboukos P, Mathioudakis K. Comparison of Linear and Nonlinear Gas Turbine Performance Diagnostics. Journal of Engineering for Gas Turbines and Power 2005;127:49–56. doi:10.1115/1.1788688.

[128] Li YG, Nilkitsaranont P. Gas Turbine Performance Prognostic for Condition-Based Maintenance. Applied Energy 2009;86:2152–61. doi:10.1016/J.APENERGY.2009.02.011.

[129] Ogaji SOT, Li YG, Sampath S, Singh R. Gas Path Fault Diagnosis of a Turbofan Engine From Transient Data Using Artificial Neural Networks. ASME Turbo Expo 2003, Atlanta, Georgia, USA: 2003, p. 405–14. doi:10.1115/GT2003-38423.

[130] GasTurb GmbH. GasTurb 13 Design and Off-Design Performance of Gas Turbines. Germany: 2017.

[131] Kurzke J. Correlations Hidden in Compressor Maps. ASME 2011 Turbo Expo, Vancouver, British Columbia, Canada: 2011, p. 161–70. doi:10.1115/GT2011-45519.

[132] Kurzke J. An Enhanced Off-Design Performance Model for Single Stage Fans. ASME Turbo Expo 2014, Düsseldorf, Germany: 2014, p. 9p. doi:10.1115/GT2014-26449.

[133] Saravanamuttoo HIH, Rogers GFC, Cohen H. Gas turbine theory. 5th ed. Prentice Hall; 2001.

[134] Peng S, Sun B, Ying Y, Wu K, He L, Xu Q. Accuracy Research on the Modeling Methods of the Gas Turbine Components Characteristics. ASME Turbo Expo 2014, Düsseldorf, Germany: 2014, p. 9p. doi:10.1115/GT2014-25600.

[135] Loboda I, Yepifanov S, Feldshteyn Y. Deviation Problem in Gas Turbine Health Monitoring. Proceedings of IASTED International Conference on Power and Energy Systems, Clearwater Beach, Florida, USA: 2004, p. 6.

[136] Haykin S. Neural Networks and Learning Machines. 3rd editio. Pearson Prentice Hall; 2009.

[137] Duda RO, Hart PE, Stork DG. Pattern Classification. 2nd. Wiley; 2001.

[138] Cristianini N, Shawe-Taylor J. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. 1st. Cambridge University Press; 2000.

[139] Beale M, Hagan M, Demuth H. Neural Network Toolbox User's Guide. Natick, Massachusetts: MathWorks, Inc.; 2014.

[140] Loboda I, Yepifanov S, Feldshteyn Y. A More Realistic Scheme of Deviation Error Representation for Gas Turbine Diagnostics. Int J Turbo Jet-Engines 2013;30:179–89. doi:10.1515/tjj-2013-0006.

[141] Zacksenhouse M, Braun S, Feldman M, Sidahmed M. Toward Helicopter Gearbox Diagnostics From a Small Number of Examples. Mechanical Systems and Signal Processing 2000;14:523–43. doi:10.1006/mssp.2000.1297.

[142] Salvador F-A, Felipe de Jesús C-A, Igor L, Juan Luis P-R. Gas Turbine Diagnostic Algorithm Testing Using the Software ProDiMES. Ingeniería Investigación y Tecnología 2017;18:75–86.

[143] Cao J, Zhang K, Luo M, Yin C, Lai X. Extreme learning machine and adaptive sparse representation for image classification. Neural Networks 2016;81:91–102. doi:10.1016/j.neunet.2016.06.001.

[144] Cao J, Hao J, Lai X, Vong C-M, Luo M. Ensemble extreme learning machine and sparse representation classification. Journal of the Franklin Institute 2016;353:4526–41. doi:10.1016/j.jfranklin.2016.08.024.

Appendix 1: Permissions to Reproduce Copyrighted Material

From: tcsme@cdnsciencepub.com
Sent: August-22-18 9:27 AM
To: 'Juan Luis Perez Ruiz'
Subject: RE: Request for permission TCSME-2017-0009.R1
Dear Dr. Pérez-Ruiz,

Yes, you may include content of your paper in your PhD thesis, so long as you cite the content as in press with the TCSME.

If you need anything further, please feel free to contact me.

Best regards and good wishes for your thesis.

Laura

Canadian Science Publishing

Laura Duke
Editorial Assistant
Transactions of the Canadian Society for Mechanical Engineering
t 613.656.9846 ext.234 **f** 613.656.9838
cdnsciencepub.com | facebook | twitter

Beth Darchi <DarchiB@asme.org>
Para: Juan Luis Perez Ruiz <perezruiz305@gmail.com>
19 de septiembre de 2018, 13:53

Dear Prof. Pérez-Ruiz,

It is our pleasure to grant you permission to use **all or any part of** the ASME paper "A Benchmarking Analysis of a Data-Driven Gas Turbine Diagnostic Approach," by Igor Loboda, Juan Luis Pérez-Ruiz and Sergiy Yepifanov, Paper No. GT2018-76887, cited in your letter for inclusion in a PhD thesis entitled Non-linear Gas Turbine Thermodynamic Model Applied to Fault Diagnostics* to be published by Instituto Politécnico Nacional.

Permission is granted for the specific use as stated herein and does not permit further use of the materials without proper authorization. Proper attribution must be made to the author(s) of the materials. **Please note:** if any or all of the figures and/or Tables are of another source, permission should be granted from that outside source or include the reference of the original source. ASME does not grant permission for outside source material that may be referenced in the ASME works.

As is customary, we request that you ensure full acknowledgment of this material, the author(s), source and ASME as original publisher. Acknowledgment must be retained on all pages where figure is printed and distributed.

Many thanks for your interest in ASME publications.

Sincerely,

Beth Darchi
Publishing Administrator
ASME
2 Park Avenue
New York, NY 10016-5990
Tel 1.212.591.7700
darchib@asme.org

* The thesis title was changed later to "*Development and Evaluation of Algorithms for Gas Turbine Condition Monitoring and Diagnostics*"

Appendix 2: Subroutines of Thermodynamic Model Software

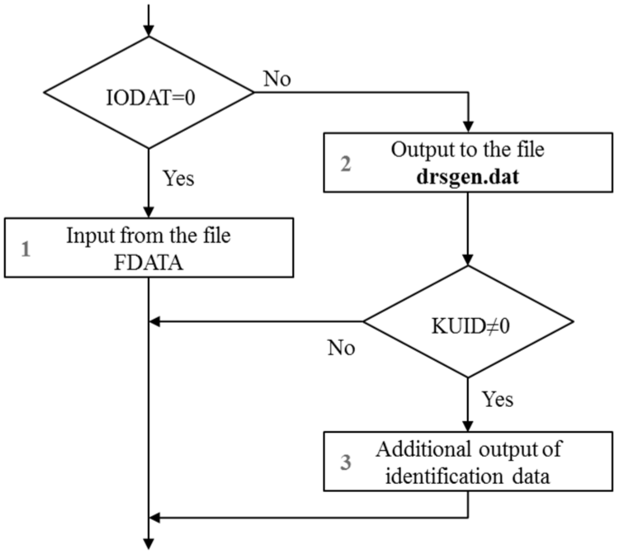

IODAT=0
No
Yes
2 Output to the file drsgen.dat
1 Input from the file FDATA
KUID≠0
No
Yes
3 Additional output of identification data

Figure 0.1 Program “INOUTD” (Input from data.dat, output to data.dat).

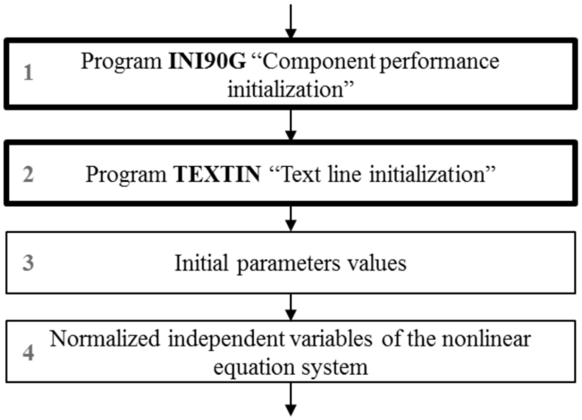

1 Program INI90G “Component performance initialization”
2 Program TEXTIN “Text line initialization”
3 Initial parameters values
4 Normalized independent variables of the nonlinear equation system

Figure 0.2 Program “NACHBL” (Initial block common for different calculations).

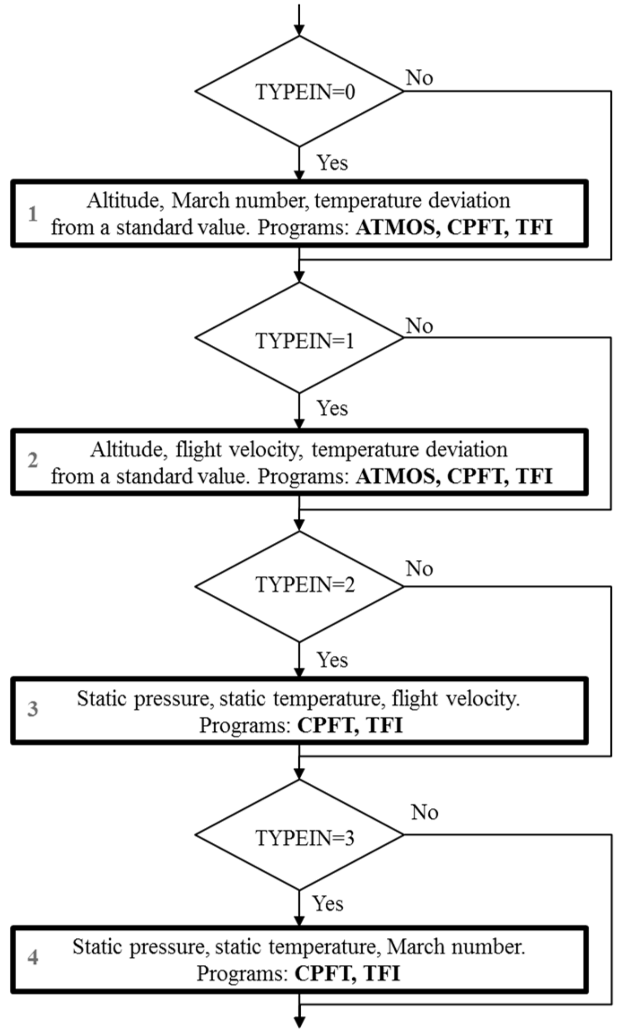

TYPEIN=0
No
Yes
1
Altitude, March number, temperature deviation from a standard value. Programs: ATMOS, CPFT, TFI
TYPEIN=1
No
Yes
2
Altitude, flight velocity, temperature deviation from a standard value. Programs: ATMOS, CPFT, TFI
TYPEIN=2
No
Yes
3
Static pressure, static temperature, flight velocity. Programs: CPFT, TFI
TYPEIN=3
No
Yes
4
Static pressure, static temperature, March number. Programs: CPFT, TFI

Figure 0.3 Program "ATMOSX" (Input parameters definition).

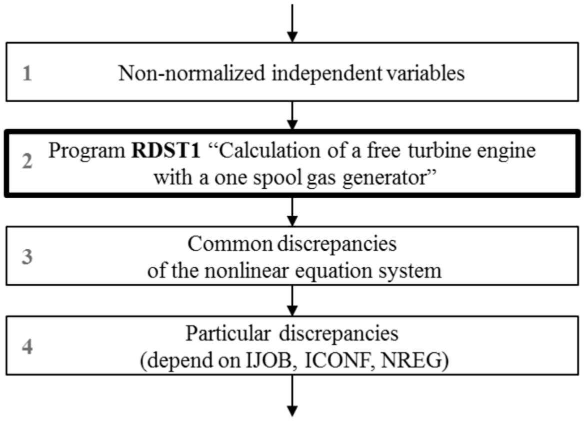

1
Non-normalized independent variables
2
Program RDST1 "Calculation of a free turbine engine with a one spool gas generator"
3
Common discrepancies of the nonlinear equation system
4
Particular discrepancies (depend on IJOB, ICONF, NREG)

Figure 0.4 Program "PDST1" (Left sides of a nonlinear equation system for a free turbine engine with one spool gas generator).

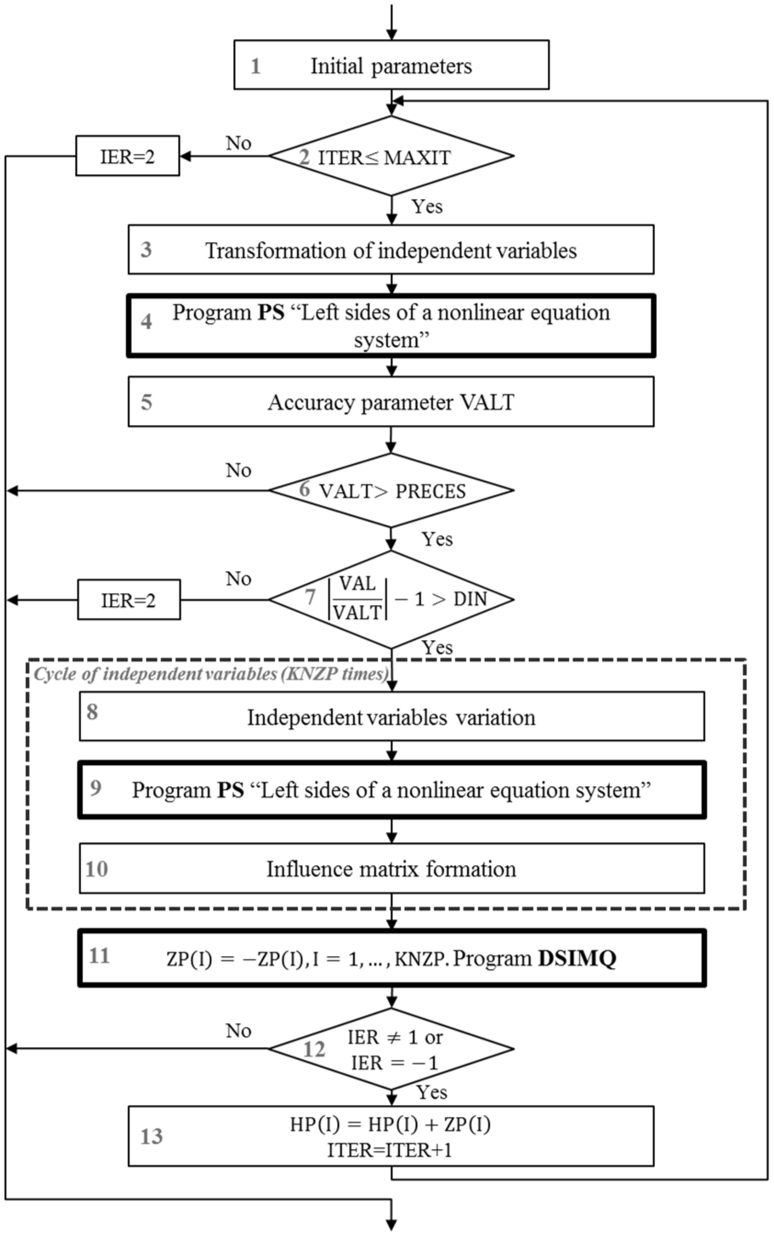

1 Initial parameters
2 ITER≤ MAXIT
No
IER=2
Yes
3 Transformation of independent variables
4 Program PS "Left sides of a nonlinear equation system"
5 Accuracy parameter VALT
6 VALT> PRECES
No
Yes
7 |VAL/VALT| − 1 > DIN
No
IER=2
Yes
Cycle of independent variables (KNZP times)
8 Independent variables variation
9 Program PS "Left sides of a nonlinear equation system"
10 Influence matrix formation
11 ZP(I) = −ZP(I), I = 1, ... , KNZP. Program DSIMQ
12 IER ≠ 1 or IER = −1
No
Yes
13 HP(I) = HP(I) + ZP(I) ITER=ITER+1

Figure 0.5 Program "RSUMN" (Standard program to solve the nonlinear system by the Newton method).

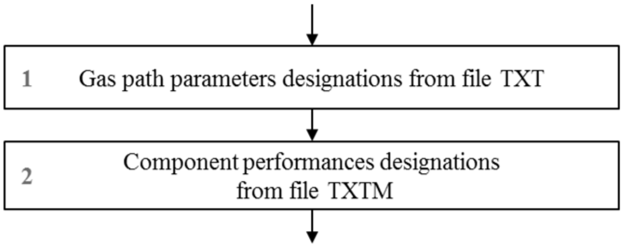

1 Gas path parameters designations from file TXT
2 Component performances designations from file TXTM

Figure 0.6 Program "TEXTIN" (Text line initialization).

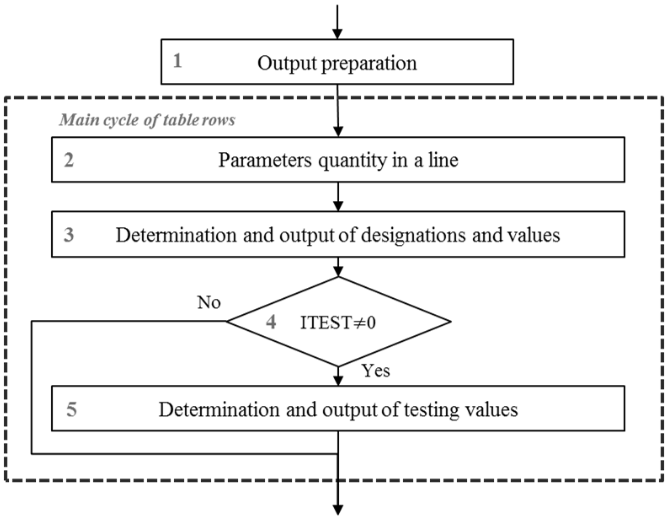

1 Output preparation
Main cycle of table rows
2 Parameters quantity in a line
3 Determination and output of designations and values
No
4 ITEST≠0
Yes
5 Determination and output of testing values

Figure 0.7 Program “PRNTF” (Output of the parameters table to a file).

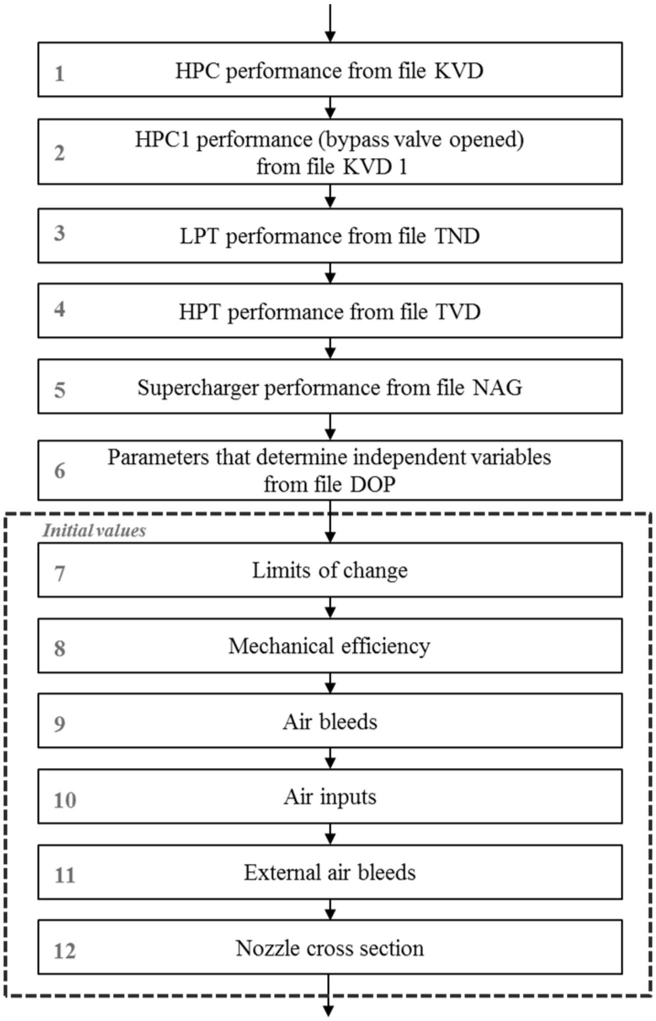

1 HPC performance from file KVD
2 HPC1 performance (bypass valve opened) from file KVD 1
3 LPT performance from file TND
4 HPT performance from file TVD
5 Supercharger performance from file NAG
6 Parameters that determine independent variables from file DOP
Initial values
7 Limits of change
8 Mechanical efficiency
9 Air bleeds
10 Air inputs
11 External air bleeds
12 Nozzle cross section

Figure 0.8 Program “INI90G” (Component performance initialization).

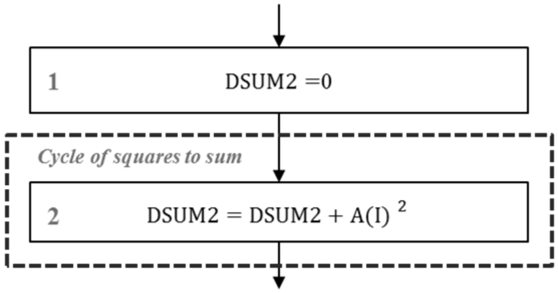


Figure 0.9 Program “DSUM2” (Calculation of a sum of squares).

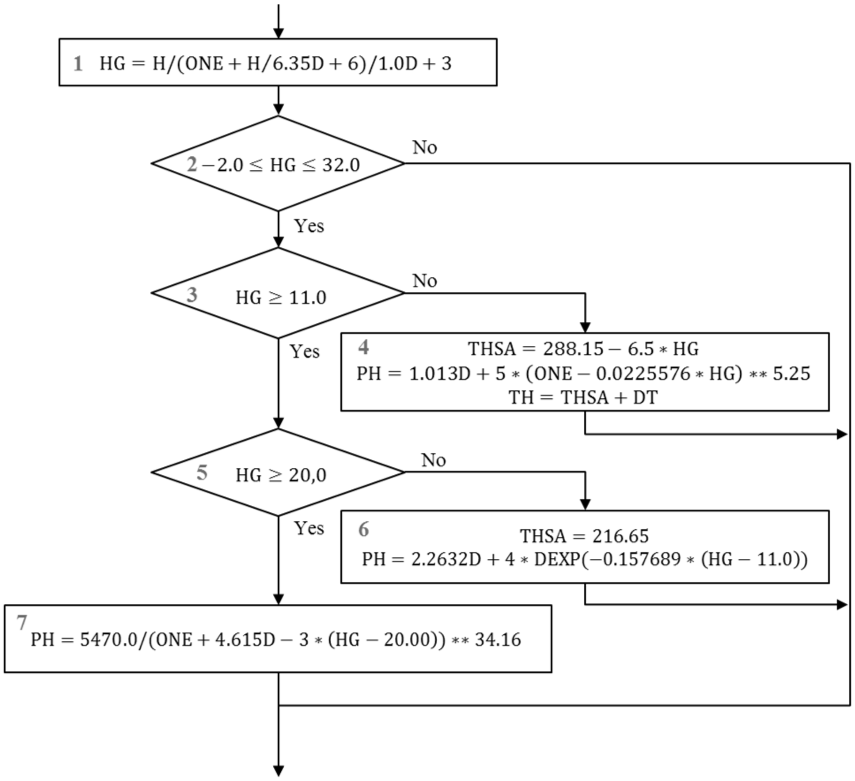

1 HG = H/(ONE + H/6.35D + 6)/1.0D + 3
2 −2.0 ≤ HG ≤ 32.0
No
Yes
3 HG ≥ 11.0
No
Yes
4 THSA = 288.15 − 6.5 ∗ HG
PH = 1.013D + 5 ∗ (ONE − 0.0225576 ∗ HG) ∗∗ 5.25
TH = THSA + DT
5 HG ≥ 20,0
No
Yes
6 THSA = 216.65
PH = 2.2632D + 4 ∗ DEXP(−0.157689 ∗ (HG − 11.0))
7 PH = 5470.0/(ONE + 4.615D − 3 ∗ (HG − 20.00)) ∗∗ 34.16

Figure 0.10 Program “ATMOS” (Standard atmosphere parameters).

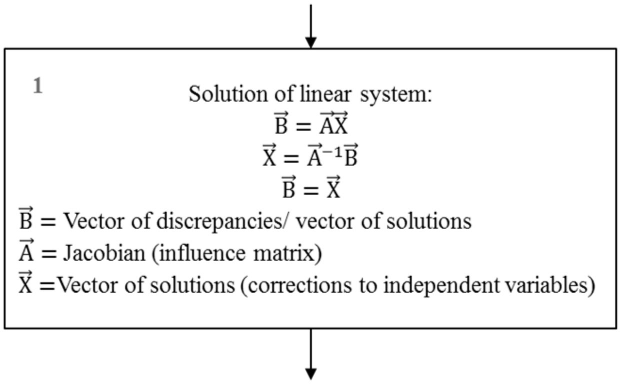

1
Solution of linear system:
B⃗ = A⃗X⃗
X⃗ = A⃗⁻¹B⃗
B⃗ = X⃗
B⃗ = Vector of discrepancies/ vector of solutions
A⃗ = Jacobian (influence matrix)
X⃗ =Vector of solutions (corrections to independent variables)

Figure 0.11 Program “DSIMQ” (Standard program of linear equations system solution by means of exclusion).

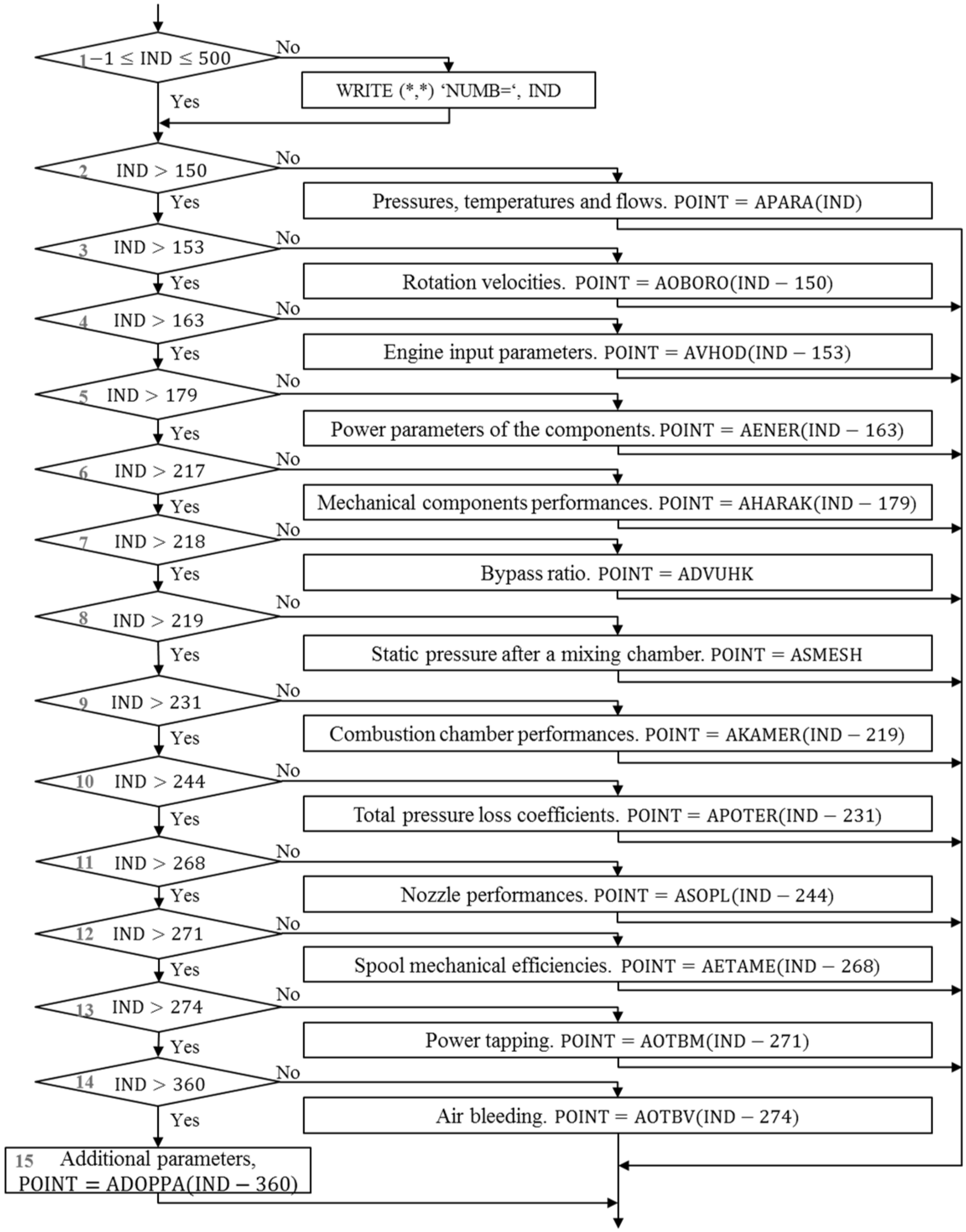

1 −1 ≤ IND ≤ 500
No
WRITE (*,*) 'NUMB=', IND
Yes
2 IND > 150
Pressures, temperatures and flows. POINT = APARA(IND)
3 IND > 153
Rotation velocities. POINT = AOBORO(IND − 150)
4 IND > 163
Engine input parameters. POINT = AVHOD(IND − 153)
5 IND > 179
Power parameters of the components. POINT = AENER(IND − 163)
6 IND > 217
Mechanical components performances. POINT = AHARAK(IND − 179)
7 IND > 218
Bypass ratio. POINT = ADVUHK
8 IND > 219
Static pressure after a mixing chamber. POINT = ASMESH
9 IND > 231
Combustion chamber performances. POINT = AKAMER(IND − 219)
10 IND > 244
Total pressure loss coefficients. POINT = APOTER(IND − 231)
11 IND > 268
Nozzle performances. POINT = ASOPL(IND − 244)
12 IND > 271
Spool mechanical efficiencies. POINT = AETAME(IND − 268)
13 IND > 274
Power tapping. POINT = AOTBM(IND − 271)
14 IND > 360
Air bleeding. POINT = AOTBV(IND − 274)
15 Additional parameters, POINT = ADOPPA(IND − 360)

Figure 0.12 Program "POINT" (Determination of a gas path parameter through its number in a common array).

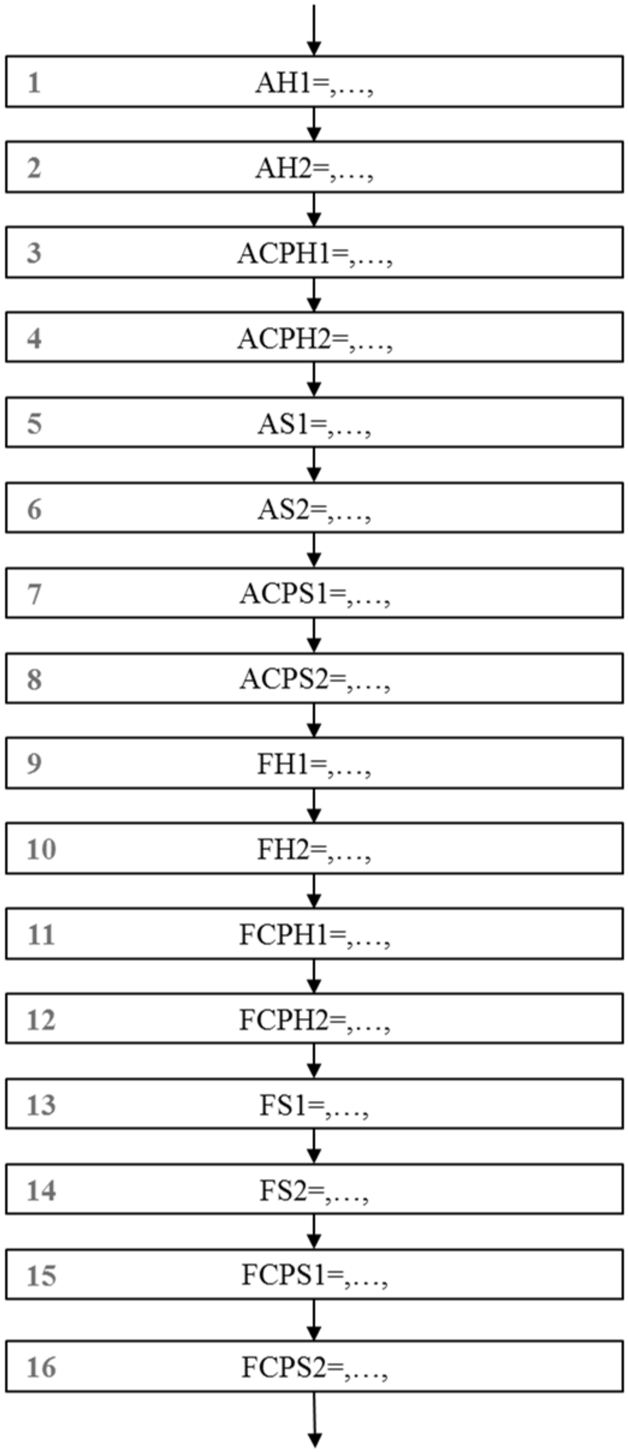


Figure 0.13 Program “STOIL” (Standard hydrocarbon fuel performances).

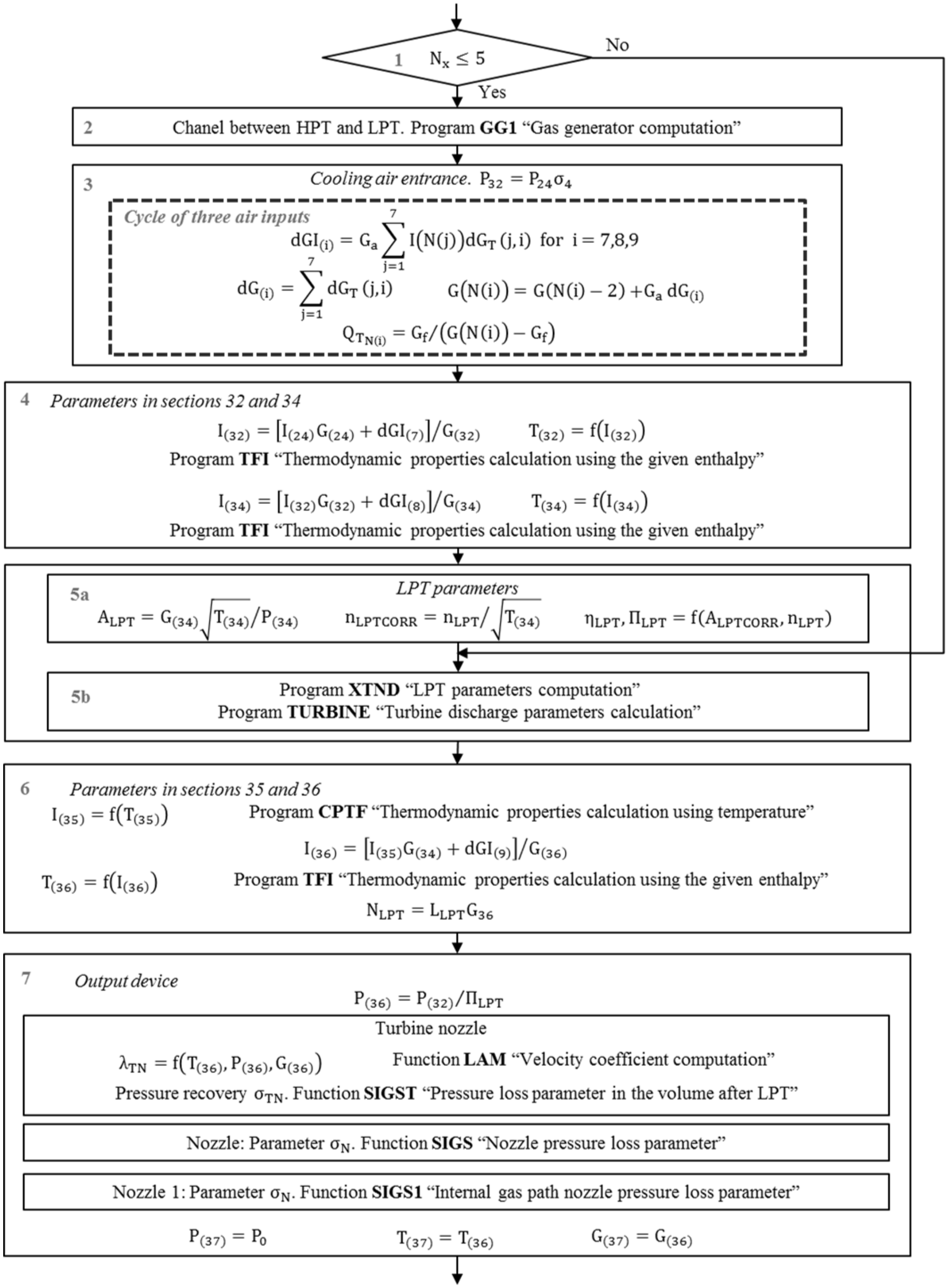

1 $N_x \le 5$
No
Yes
2 Chanel between HPT and LPT. Program **GG1** “Gas generator computation”
3 *Cooling air entrance.* $P_{32} = P_{24}\sigma_4$
Cycle of three air inputs
$dGI_{(i)} = G_a \sum_{j=1}^{7} I(N(j))dG_T(j,i)$ for $i = 7,8,9$
$dG_{(i)} = \sum_{j=1}^{7} dG_T(j,i)$
$G(N(i)) = G(N(i) - 2) + G_a\, dG_{(i)}$
$Q_{T_{N(i)}} = G_f/(G(N(i)) - G_f)$
4 *Parameters in sections 32 and 34*
$I_{(32)} = [I_{(24)}G_{(24)} + dGI_{(7)}]/G_{(32)}$
$T_{(32)} = f(I_{(32)})$
Program **TFI** “Thermodynamic properties calculation using the given enthalpy”
$I_{(34)} = [I_{(32)}G_{(32)} + dGI_{(8)}]/G_{(34)}$
$T_{(34)} = f(I_{(34)})$
Program **TFI** “Thermodynamic properties calculation using the given enthalpy”
5a *LPT parameters*
$A_{LPT} = G_{(34)}\sqrt{T_{(34)}}/P_{(34)}$
$n_{LPTCORR} = n_{LPT}/\sqrt{T_{(34)}}$
$\eta_{LPT}, \Pi_{LPT} = f(A_{LPTCORR}, n_{LPT})$
5b Program **XTND** “LPT parameters computation”
Program **TURBINE** “Turbine discharge parameters calculation”
6 *Parameters in sections 35 and 36*
$I_{(35)} = f(T_{(35)})$
Program **CPTF** “Thermodynamic properties calculation using temperature”
$I_{(36)} = [I_{(35)}G_{(34)} + dGI_{(9)}]/G_{(36)}$
$T_{(36)} = f(I_{(36)})$
Program **TFI** “Thermodynamic properties calculation using the given enthalpy”
$N_{LPT} = L_{LPT}G_{36}$
7 *Output device*
$P_{(36)} = P_{(32)}/\Pi_{LPT}$
Turbine nozzle
$\lambda_{TN} = f(T_{(36)}, P_{(36)}, G_{(36)})$
Function **LAM** “Velocity coefficient computation”
Pressure recovery σ_{TN}. Function **SIGST** “Pressure loss parameter in the volume after LPT”
Nozzle: Parameter σ_N. Function **SIGS** “Nozzle pressure loss parameter”
Nozzle 1: Parameter σ_N. Function **SIGS1** “Internal gas path nozzle pressure loss parameter”
$P_{(37)} = P_0$
$T_{(37)} = T_{(36)}$
$G_{(37)} = G_{(36)}$

Figure 0.14 Program “RDST1” (Calculation of a free turbine engine with one spool gas generator).

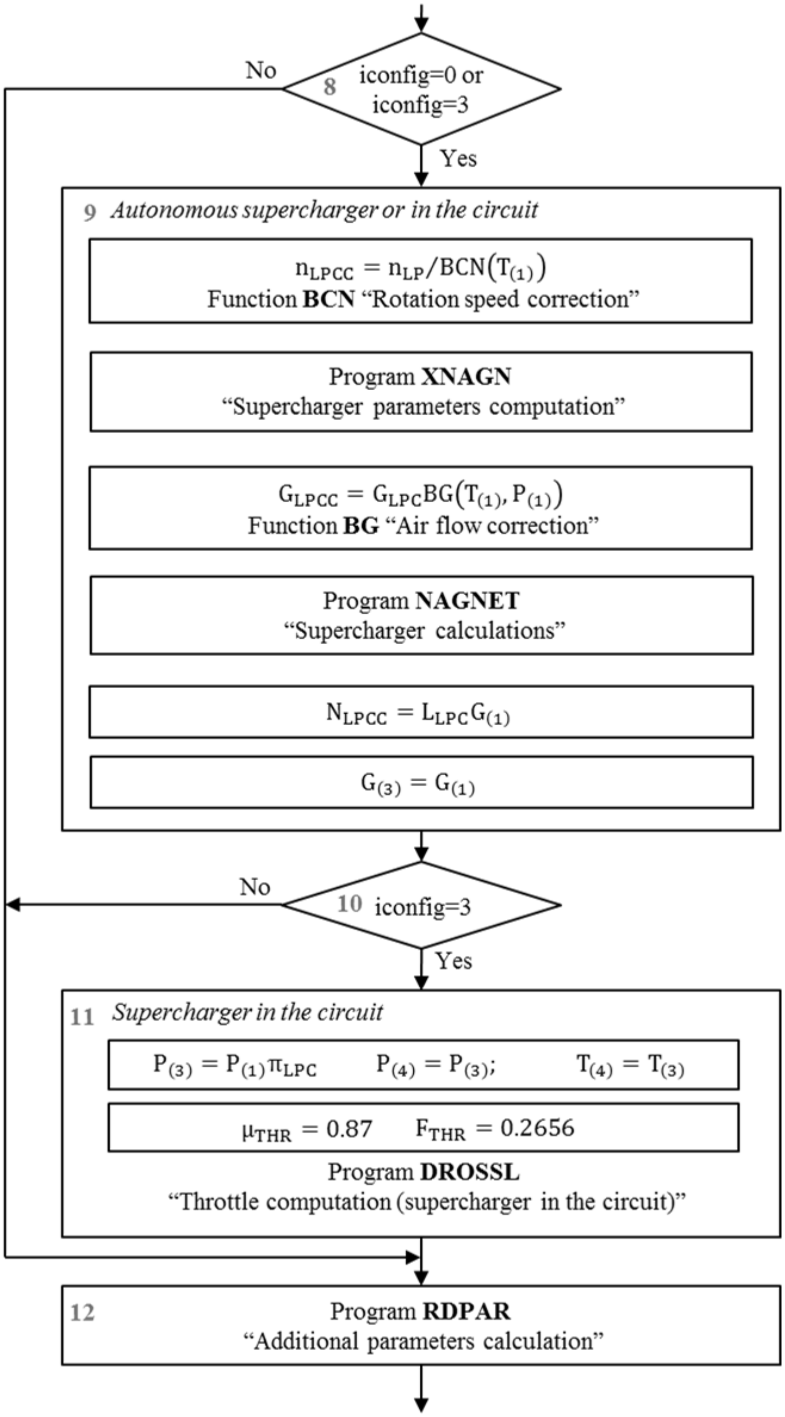

8 iconfig=0 or iconfig=3
No
Yes
9 *Autonomous supercharger or in the circuit*
$n_{LPCC} = n_{LP}/BCN(T_{(1)})$
Function **BCN** “Rotation speed correction”
Program **XNAGN**
“Supercharger parameters computation”
$G_{LPCC} = G_{LPC}BG(T_{(1)}, P_{(1)})$
Function **BG** “Air flow correction”
Program **NAGNET**
“Supercharger calculations”
$N_{LPCC} = L_{LPC}G_{(1)}$
$G_{(3)} = G_{(1)}$
10 iconfig=3
No
Yes
11 *Supercharger in the circuit*
$P_{(3)} = P_{(1)}\pi_{LPC}$ $P_{(4)} = P_{(3)}$; $T_{(4)} = T_{(3)}$
$\mu_{THR} = 0.87$ $F_{THR} = 0.2656$
Program **DROSSL**
“Throttle computation (supercharger in the circuit)”
12 Program **RDPAR**
“Additional parameters calculation”

Figure 0.15 Program “RDST1” (Continuation).

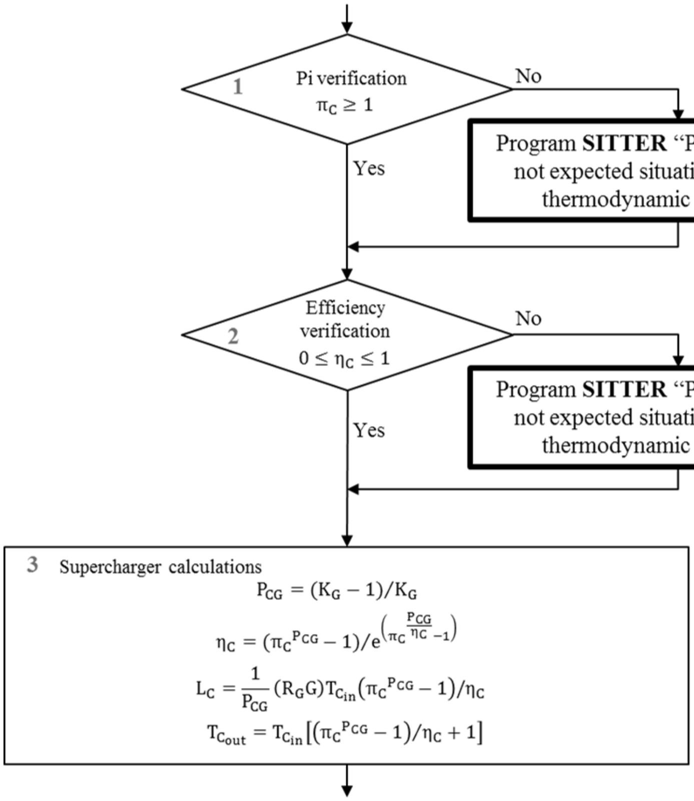

1
Pi verification
$\pi_C \geq 1$
No
Yes
Program **SITTER** "Program to expose not expected situations in common thermodynamic calculations"
2
Efficiency verification
$0 \leq \eta_C \leq 1$
No
Yes
Program **SITTER** "Program to expose not expected situations in common thermodynamic calculations"
3
Supercharger calculations
$P_{CG} = (K_G - 1)/K_G$
$\eta_C = (\pi_C{}^{P_{CG}} - 1)/e^{\left(\pi_C{}^{\frac{P_{CG}}{\eta_C}} - 1\right)}$
$L_C = \frac{1}{P_{CG}}(R_G G)T_{C_{in}}(\pi_C{}^{P_{CG}} - 1)/\eta_C$
$T_{C_{out}} = T_{C_{in}}[(\pi_C{}^{P_{CG}} - 1)/\eta_C + 1]$

Figure 0.16 Program "NAGNET" (Supercharger calculations).

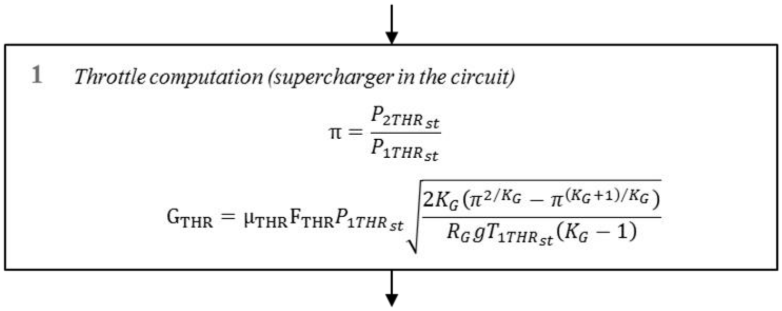

1
Throttle computation (supercharger in the circuit)
$\pi = \frac{P_{2THR_{st}}}{P_{1THR_{st}}}$
$G_{THR} = \mu_{THR} F_{THR} P_{1THR_{st}} \sqrt{\frac{2K_G(\pi^{2/K_G} - \pi^{(K_G+1)/K_G})}{R_G g T_{1THR_{st}}(K_G - 1)}}$

Figure 0.17 Program "DROSSL" (Throttle computation, supercharger in the circuit).

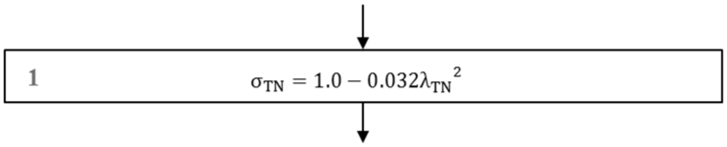


Figure 0.18 Program "SIGST" (Pressure recovery parameter in the volume after PT).

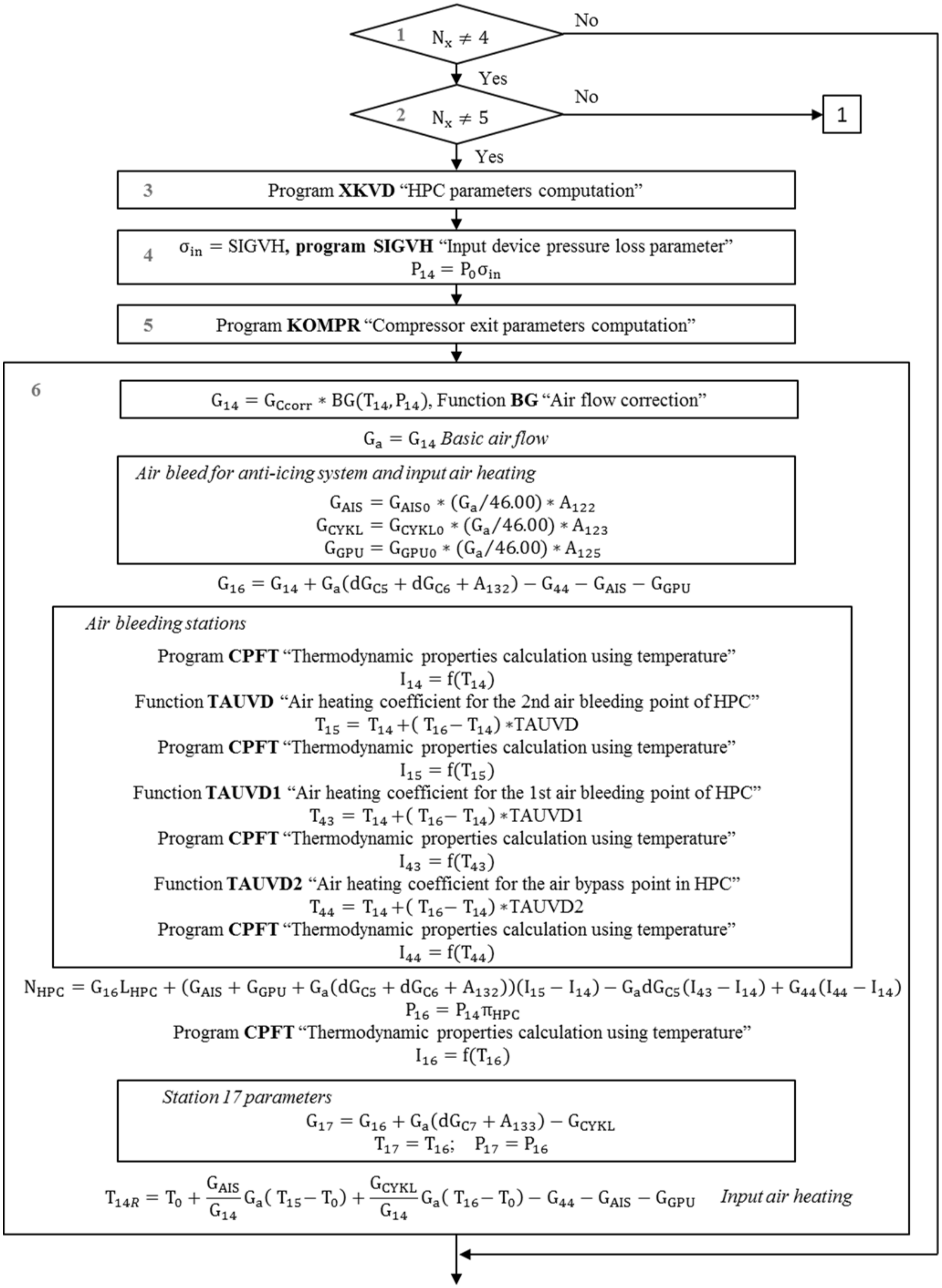
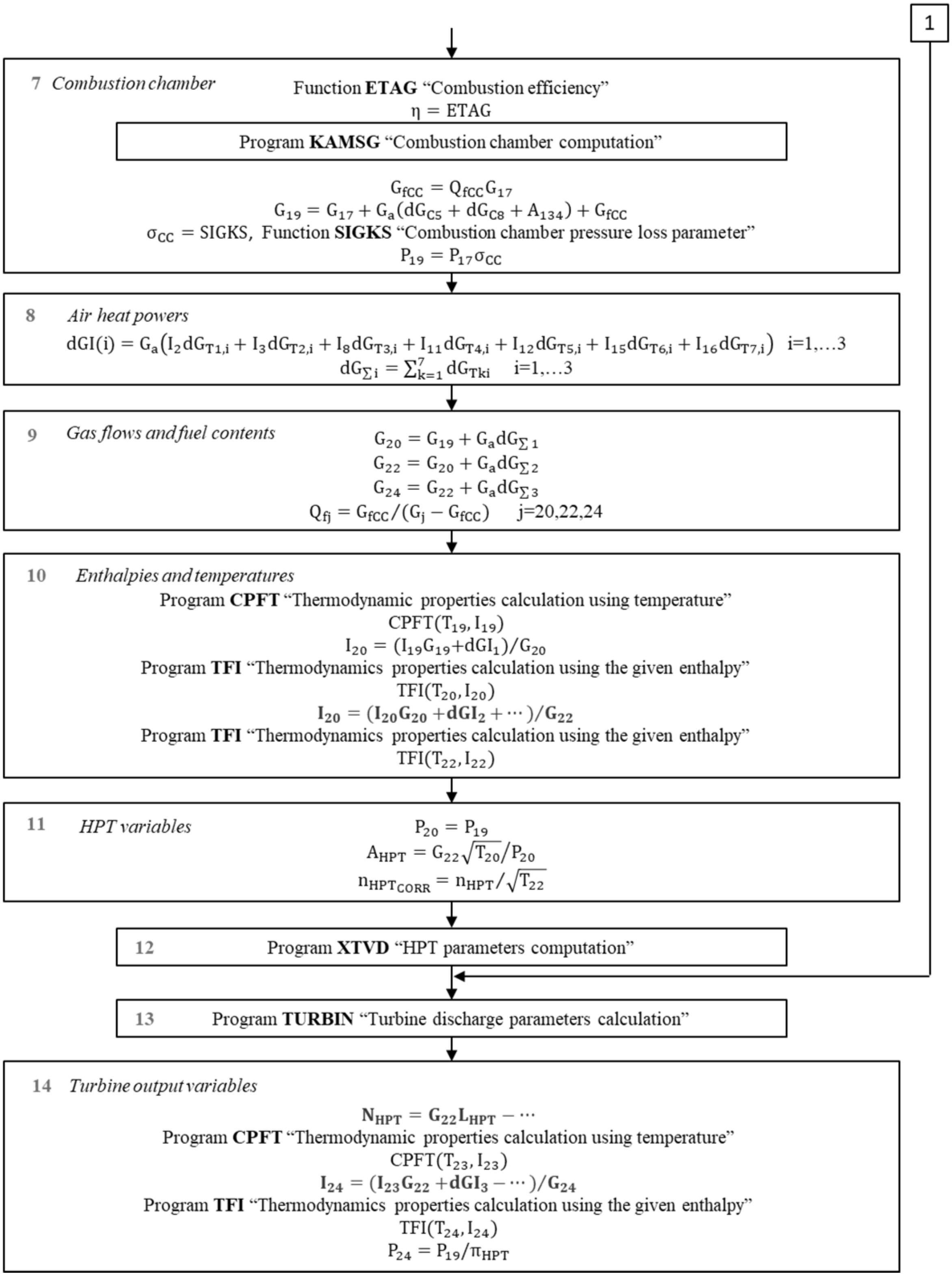

1 $N_x \neq 4$
No
Yes
2 $N_x \neq 5$
No
1
Yes
3 Program **XKVD** “HPC parameters computation”
4 σ_{in} = SIGVH, **program SIGVH** “Input device pressure loss parameter”
$P_{14} = P_0\sigma_{in}$
5 Program **KOMPR** “Compressor exit parameters computation”
6
$G_{14} = G_{Ccorr} * BG(T_{14}, P_{14})$, Function **BG** “Air flow correction”
$G_a = G_{14}$ *Basic air flow*
Air bleed for anti-icing system and input air heating
$G_{AIS} = G_{AIS0} * (G_a/46.00) * A_{122}$
$G_{CYKL} = G_{CYKL0} * (G_a/46.00) * A_{123}$
$G_{GPU} = G_{GPU0} * (G_a/46.00) * A_{125}$
$G_{16} = G_{14} + G_a(dG_{C5} + dG_{C6} + A_{132}) - G_{44} - G_{AIS} - G_{GPU}$
Air bleeding stations
Program **CPFT** “Thermodynamic properties calculation using temperature”
$I_{14} = f(T_{14})$
Function **TAUVD** “Air heating coefficient for the 2nd air bleeding point of HPC”
$T_{15} = T_{14} + (T_{16} - T_{14}) *TAUVD$
Program **CPFT** “Thermodynamic properties calculation using temperature”
$I_{15} = f(T_{15})$
Function **TAUVD1** “Air heating coefficient for the 1st air bleeding point of HPC”
$T_{43} = T_{14} + (T_{16} - T_{14}) *TAUVD1$
Program **CPFT** “Thermodynamic properties calculation using temperature”
$I_{43} = f(T_{43})$
Function **TAUVD2** “Air heating coefficient for the air bypass point in HPC”
$T_{44} = T_{14} + (T_{16} - T_{14}) *TAUVD2$
Program **CPFT** “Thermodynamic properties calculation using temperature”
$I_{44} = f(T_{44})$
$N_{HPC} = G_{16}L_{HPC} + (G_{AIS} + G_{GPU} + G_a(dG_{C5} + dG_{C6} + A_{132}))(I_{15} - I_{14}) - G_a dG_{C5}(I_{43} - I_{14}) + G_{44}(I_{44} - I_{14})$
$P_{16} = P_{14}\pi_{HPC}$
Program **CPFT** “Thermodynamic properties calculation using temperature”
$I_{16} = f(T_{16})$
Station 17 parameters
$G_{17} = G_{16} + G_a(dG_{C7} + A_{133}) - G_{CYKL}$
$T_{17} = T_{16}; \quad P_{17} = P_{16}$
$T_{14R} = T_0 + \frac{G_{AIS}}{G_{14}} G_a(T_{15} - T_0) + \frac{G_{CYKL}}{G_{14}} G_a(T_{16} - T_0) - G_{44} - G_{AIS} - G_{GPU}$ *Input air heating*

Figure 0.19 Program “Gg1” (Gas generator computation).



7 *Combustion chamber*

Function **ETAG** "Combustion efficiency"

$\eta = \mathrm{ETAG}$

Program **KAMSG** "Combustion chamber computation"

$G_{fCC} = Q_{fCC} G_{17}$

$G_{19} = G_{17} + G_a(dG_{C5} + dG_{C8} + A_{134}) + G_{fCC}$

$\sigma_{CC} = \mathrm{SIGKS}$, Function **SIGKS** "Combustion chamber pressure loss parameter"

$P_{19} = P_{17}\sigma_{CC}$

8 *Air heat powers*

$dGI(i) = G_a\left(I_2 dG_{T1,i} + I_3 dG_{T2,i} + I_8 dG_{T3,i} + I_{11} dG_{T4,i} + I_{12} dG_{T5,i} + I_{15} dG_{T6,i} + I_{16} dG_{T7,i}\right)$ i=1,…3

$dG_{\sum i} = \sum_{k=1}^{7} dG_{Tki}$ i=1,…3

9 *Gas flows and fuel contents*

$G_{20} = G_{19} + G_a dG_{\sum 1}$

$G_{22} = G_{20} + G_a dG_{\sum 2}$

$G_{24} = G_{22} + G_a dG_{\sum 3}$

$Q_{fj} = G_{fCC}/(G_j - G_{fCC})$ j=20,22,24

10 *Enthalpies and temperatures*

Program **CPFT** "Thermodynamic properties calculation using temperature"

$\mathrm{CPFT}(T_{19}, I_{19})$

$I_{20} = (I_{19}G_{19} + dGI_1)/G_{20}$

Program **TFI** "Thermodynamics properties calculation using the given enthalpy"

$\mathrm{TFI}(T_{20}, I_{20})$

$\mathbf{I_{20} = (I_{20}G_{20} + dGI_2 + \cdots)/G_{22}}$

Program **TFI** "Thermodynamics properties calculation using the given enthalpy"

$\mathrm{TFI}(T_{22}, I_{22})$

11 *HPT variables*

$P_{20} = P_{19}$

$A_{HPT} = G_{22}\sqrt{T_{20}}/P_{20}$

$n_{HPT_{CORR}} = n_{HPT}/\sqrt{T_{22}}$

12 Program **XTVD** "HPT parameters computation"

13 Program **TURBIN** "Turbine discharge parameters calculation"

14 *Turbine output variables*

$\mathbf{N_{HPT} = G_{22}L_{HPT}} - \cdots$

Program **CPFT** "Thermodynamic properties calculation using temperature"

$\mathrm{CPFT}(T_{23}, I_{23})$

$\mathbf{I_{24} = (I_{23}G_{22} + dGI_3} - \cdots\mathbf{)/G_{24}}$

Program **TFI** "Thermodynamics properties calculation using the given enthalpy"

$\mathrm{TFI}(T_{24}, I_{24})$

$P_{24} = P_{19}/\pi_{HPT}$

Figure 0.20 Program "Gg1" (Continuation).

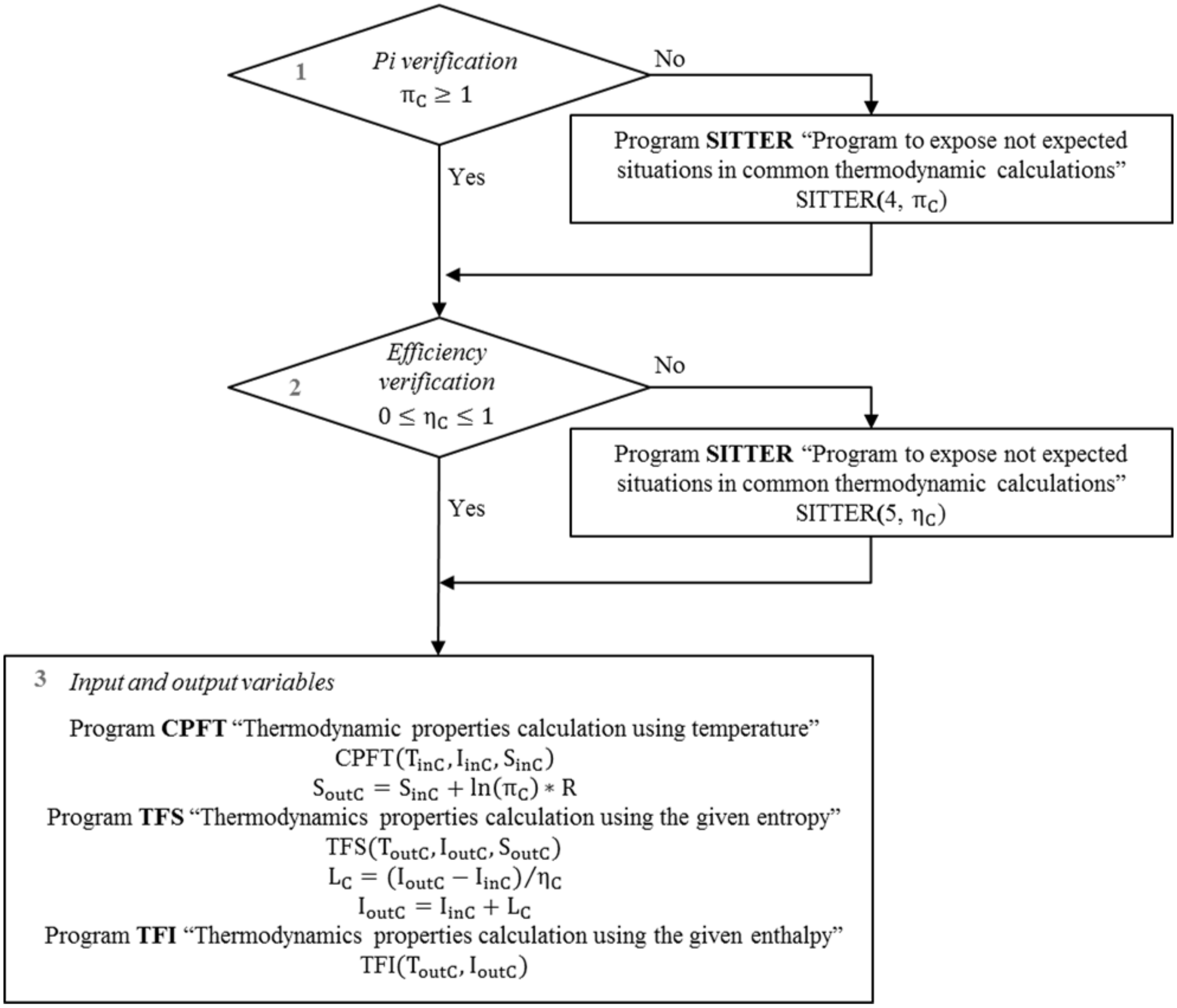

1
Pi verification
$\pi_C \geq 1$
No
Yes
Program **SITTER** “Program to expose not expected situations in common thermodynamic calculations”
SITTER(4, π_C)
2
Efficiency verification
$0 \leq \eta_C \leq 1$
No
Yes
Program **SITTER** “Program to expose not expected situations in common thermodynamic calculations”
SITTER(5, η_C)
3
Input and output variables
Program **CPFT** “Thermodynamic properties calculation using temperature”
$\mathrm{CPFT}(T_{inC}, I_{inC}, S_{inC})$
$S_{outC} = S_{inC} + \ln(\pi_C) * R$
Program **TFS** “Thermodynamics properties calculation using the given entropy”
$\mathrm{TFS}(T_{outC}, I_{outC}, S_{outC})$
$L_C = (I_{outC} - I_{inC})/\eta_C$
$I_{outC} = I_{inC} + L_C$
Program **TFI** “Thermodynamics properties calculation using the given enthalpy”
$\mathrm{TFI}(T_{outC}, I_{outC})$

Figure 0.21 Program “KOMPR” (Compressor exit parameters computation).

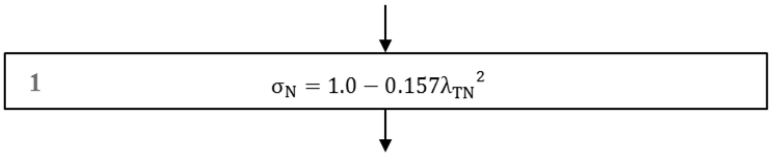


Figure 0.22 Program “SIGS” (Nozzle pressure recovery parameter).

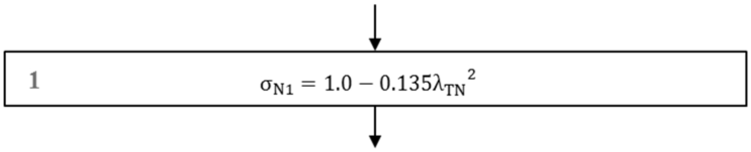


Figure 0.23 Program “SIGS1” (Internal gas path nozzle pressure recovery parameter)

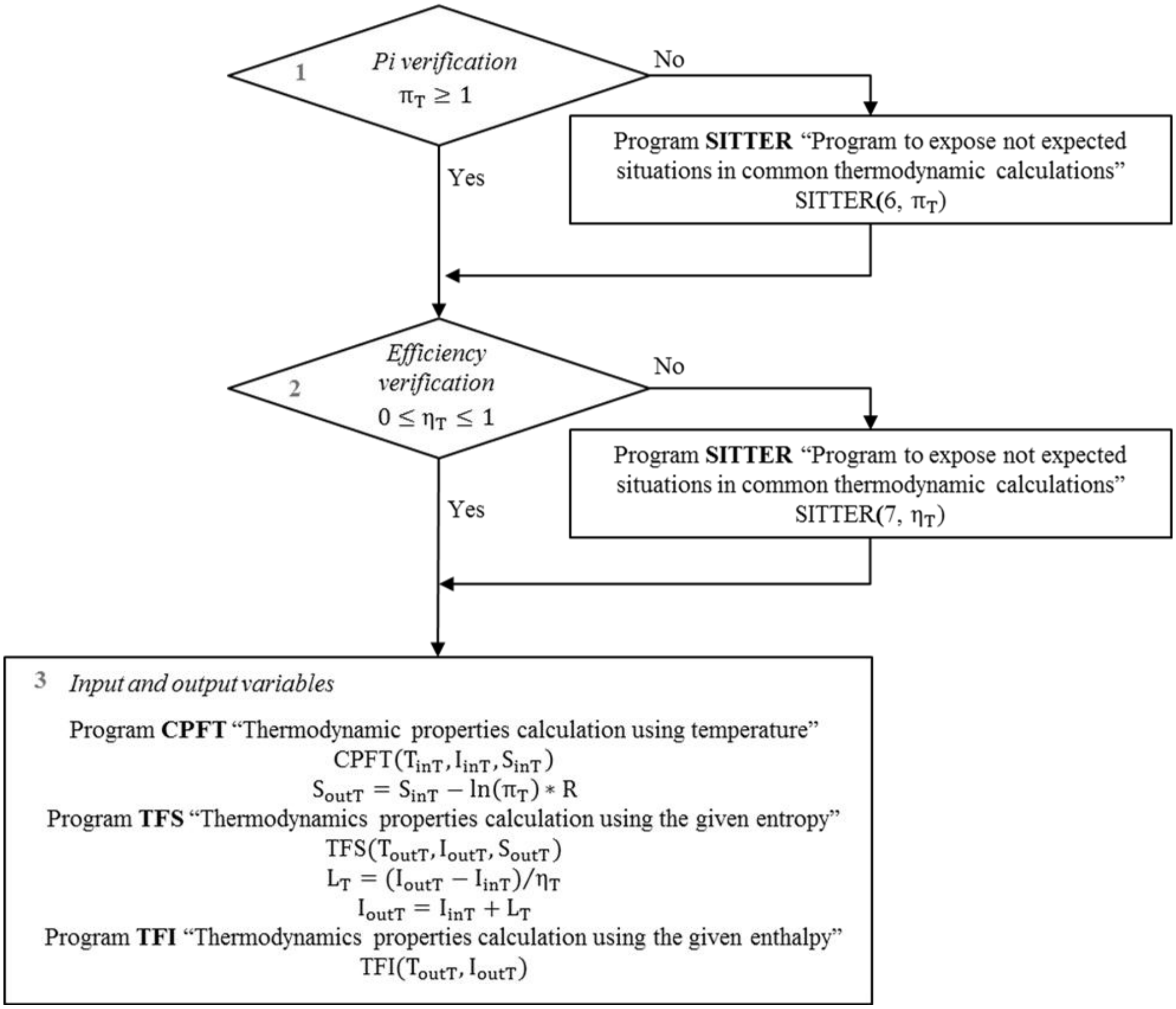

1
Pi verification
$\pi_T \geq 1$
No
Yes
Program **SITTER** "Program to expose not expected situations in common thermodynamic calculations"
SITTER(6, π_T)
2
Efficiency verification
$0 \leq \eta_T \leq 1$
No
Yes
Program **SITTER** "Program to expose not expected situations in common thermodynamic calculations"
SITTER(7, η_T)
3
Input and output variables
Program **CPFT** "Thermodynamic properties calculation using temperature"
CPFT($T_{inT}, I_{inT}, S_{inT}$)
$S_{outT} = S_{inT} - \ln(\pi_T) * R$
Program **TFS** "Thermodynamics properties calculation using the given entropy"
TFS($T_{outT}, I_{outT}, S_{outT}$)
$L_T = (I_{outT} - I_{inT})/\eta_T$
$I_{outT} = I_{inT} + L_T$
Program **TFI** "Thermodynamics properties calculation using the given enthalpy"
TFI(T_{outT}, I_{outT})

Figure 0.24 Program "TURBIN" (Turbine discharge parameters calculation)

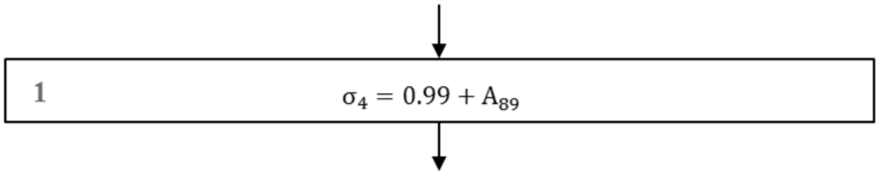


Figure 0.25 Program "SIGP4" (Pressure recovery parameter in the channel between HPT and PT).

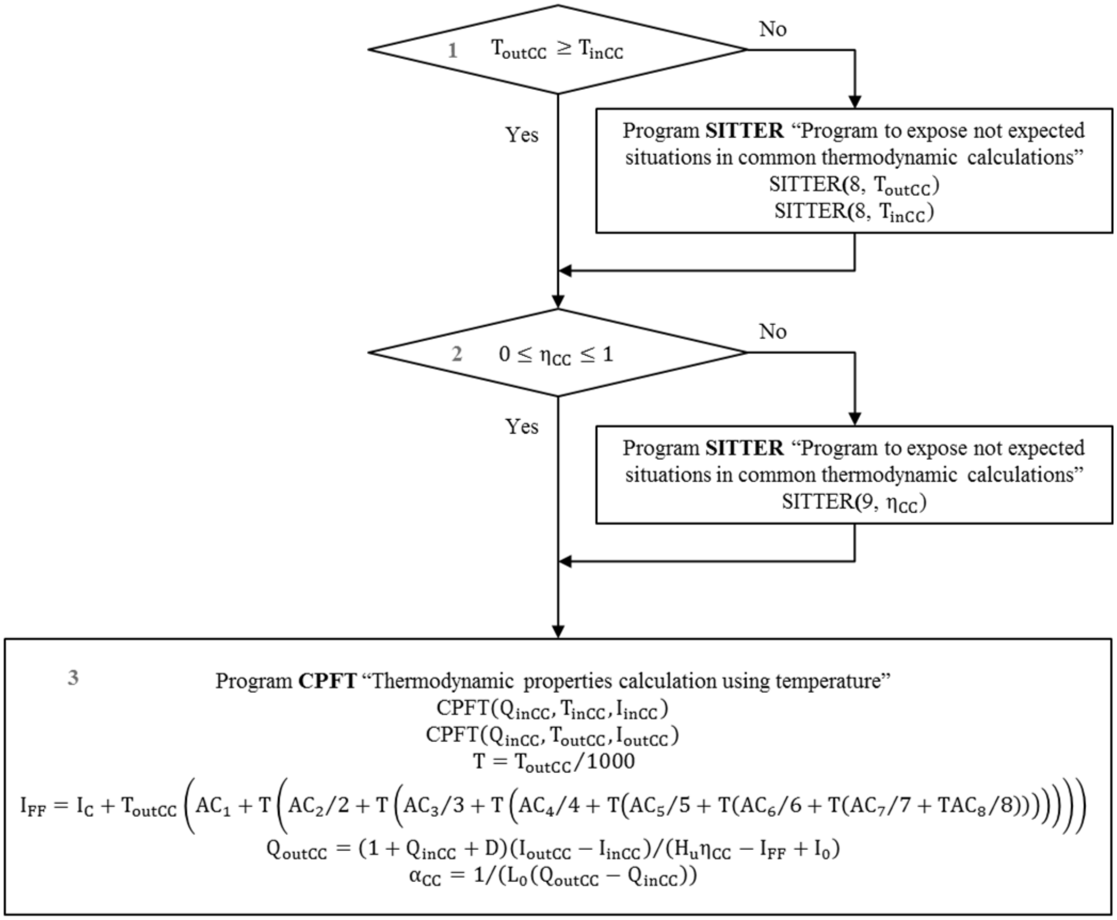

1 $T_{outCC} \ge T_{inCC}$
No
Yes
Program **SITTER** "Program to expose not expected situations in common thermodynamic calculations"
$SITTER(8, T_{outCC})$
$SITTER(8, T_{inCC})$
2 $0 \le \eta_{CC} \le 1$
No
Yes
Program **SITTER** "Program to expose not expected situations in common thermodynamic calculations"
$SITTER(9, \eta_{CC})$
3
Program **CPFT** "Thermodynamic properties calculation using temperature"
$CPFT(Q_{inCC}, T_{inCC}, I_{inCC})$
$CPFT(Q_{inCC}, T_{outCC}, I_{outCC})$
$T = T_{outCC}/1000$
$I_{FF} = I_C + T_{outCC}\left(AC_1 + T\left(AC_2/2 + T\left(AC_3/3 + T\left(AC_4/4 + T\left(AC_5/5 + T(AC_6/6 + T(AC_7/7 + TAC_8/8))\right)\right)\right)\right)\right)$
$Q_{outCC} = (1 + Q_{inCC} + D)(I_{outCC} - I_{inCC})/(H_u\eta_{CC} - I_{FF} + I_0)$
$\alpha_{CC} = 1/(L_0(Q_{outCC} - Q_{inCC}))$

Figure 0.26 Program "KAMSG" (Combustion chamber computation).

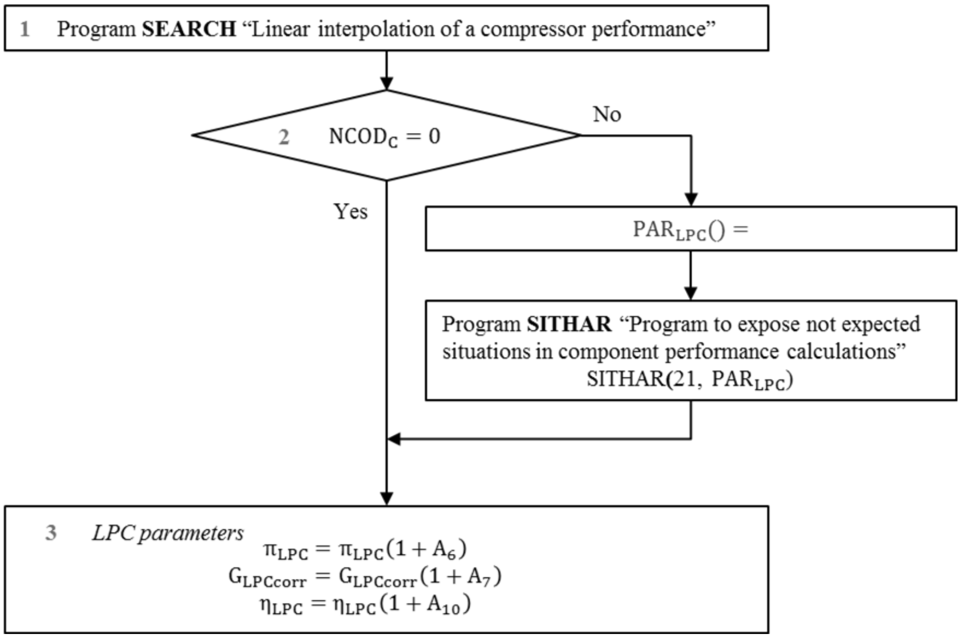

1 Program **SEARCH** "Linear interpolation of a compressor performance"
2 $NCOD_C = 0$
No
Yes
$PAR_{LPC}() =$
Program **SITHAR** "Program to expose not expected situations in component performance calculations"
$SITHAR(21, PAR_{LPC})$
3 *LPC parameters*
$\pi_{LPC} = \pi_{LPC}(1 + A_6)$
$G_{LPCcorr} = G_{LPCcorr}(1 + A_7)$
$\eta_{LPC} = \eta_{LPC}(1 + A_{10})$

Figure 0.27 Program "XNAGN" (Supercharger parameters computation).

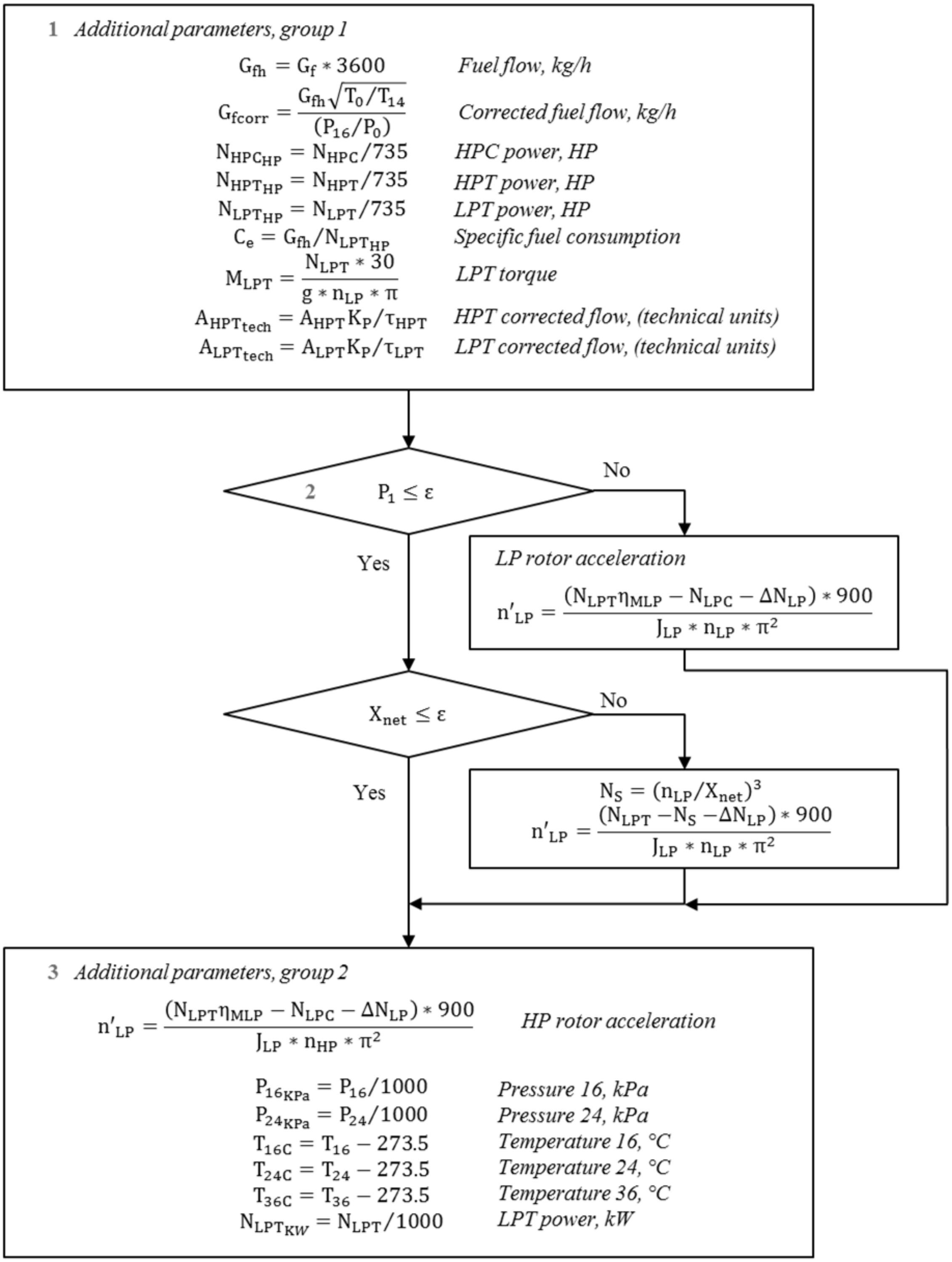

1 Additional parameters, group 1
$G_{fh} = G_f * 3600$
Fuel flow, kg/h
$G_{fcorr} = \frac{G_{fh}\sqrt{T_0/T_{14}}}{(P_{16}/P_0)}$
Corrected fuel flow, kg/h
$N_{HPC_{HP}} = N_{HPC}/735$
HPC power, HP
$N_{HPT_{HP}} = N_{HPT}/735$
HPT power, HP
$N_{LPT_{HP}} = N_{LPT}/735$
LPT power, HP
$C_e = G_{fh}/N_{LPT_{HP}}$
Specific fuel consumption
$M_{LPT} = \frac{N_{LPT} * 30}{g * n_{LP} * \pi}$
LPT torque
$A_{HPT_{tech}} = A_{HPT}K_P/\tau_{HPT}$
HPT corrected flow, (technical units)
$A_{LPT_{tech}} = A_{LPT}K_P/\tau_{LPT}$
LPT corrected flow, (technical units)
2 $P_1 \leq \varepsilon$
No
Yes
LP rotor acceleration
$n'_{LP} = \frac{(N_{LPT}\eta_{MLP} - N_{LPC} - \Delta N_{LP}) * 900}{J_{LP} * n_{LP} * \pi^2}$
$X_{net} \leq \varepsilon$
No
Yes
$N_S = (n_{LP}/X_{net})^3$
$n'_{LP} = \frac{(N_{LPT} - N_S - \Delta N_{LP}) * 900}{J_{LP} * n_{LP} * \pi^2}$
3 Additional parameters, group 2
$n'_{LP} = \frac{(N_{LPT}\eta_{MLP} - N_{LPC} - \Delta N_{LP}) * 900}{J_{LP} * n_{HP} * \pi^2}$
HP rotor acceleration
$P_{16_{KPa}} = P_{16}/1000$
Pressure 16, kPa
$P_{24_{KPa}} = P_{24}/1000$
Pressure 24, kPa
$T_{16C} = T_{16} - 273.5$
Temperature 16, °C
$T_{24C} = T_{24} - 273.5$
Temperature 24, °C
$T_{36C} = T_{36} - 273.5$
Temperature 36, °C
$N_{LPT_{KW}} = N_{LPT}/1000$
LPT power, kW

Figure 0.28 Program “RDPAR” (Additional parameters calculation).

```
$\lambda_{CC} = f(T_{19}, P_{19}, G_{19})$
Function **LAM** "Velocity coefficient computation"
$\sigma_{CC} = 1 - 0.88\lambda_{CC}{}^{2} + A_{66}$
```

Figure 0.29 Program "SIGKS" (Combustion chamber pressure recovery parameter).

```
$\tau_{HP} = 0.55 + A_{124}$
```

Figure 0.30 Program "TAUVD" (Air heating coefficient for the 2nd air bleeding point of HPC).

```
$\tau_{HP1} = 0.3$
```

Figure 0.31 Program "TAUVD1" (Air heating coefficient for the 1st air bleeding point of HPC).

```
$\tau_{HP2} = 0.32$
```

Figure 0.32 Program "TAUVD2" (Air heating coefficient for the air bypass point in HPC).

```
$\lambda_{in} = f(T_0, P_0, G_{HPCcorr})$
$\sigma_{in} = 1 - 0.0355\lambda_{in} + A_3$
$PP = P_0\sigma_{in}$
$GG = G_{HPCcorr}BG(T_{14}, PP)$
Function **BG** "Air flow correction"
$\lambda_{in} = f(T_0, P_0, GG)$
$\sigma_{in} = 1 - 0.0355\lambda_{in} + A_3$
```

Figure 0.33 Program "SIGVH" (Input device pressure recovery parameter).

```
$n_{AS} = R_{AS}N_{HP}$
$M_{AStech} = f(n_{AS})$ –Program **LINTX** "Interpolation of the function of one argument"
$M_{AS} = M_{AStech} * 9.8065$
$N_{AS} = \frac{\pi}{30}M_{AS} * n_{AS} * \eta_{AS}$
```

Figure 0.34 Program "PARVTS" (Air starter parameters).

```
$SUM = \sum_{i=1}^{N} A_i$
```

Figure 0.35 Program “DSUM” (Sum calculation).

1

$$\mathrm{BCN} = \sqrt{\mathrm{T}/\mathrm{T}_0}$$

Figure 0.36 Program “BCN” (Rotation speed correction).

1

$$\eta_{\mathrm{CC}} = 0.995(1 + \mathrm{A}_{69})$$

Figure 0.37 Program “ETAG” (Combustion efficiency).

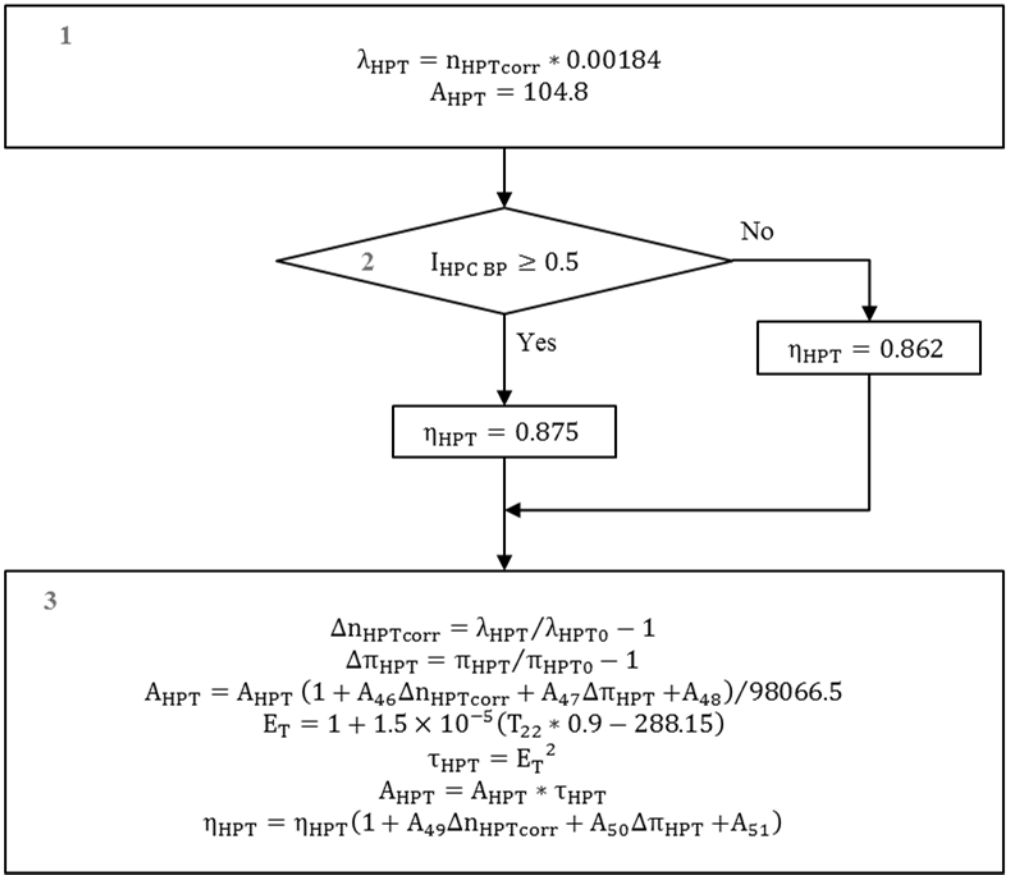

1
$\lambda_{\mathrm{HPT}} = \mathrm{n}_{\mathrm{HPTcorr}} * 0.00184$
$\mathrm{A}_{\mathrm{HPT}} = 104.8$
2
$\mathrm{I}_{\mathrm{HPC\,BP}} \geq 0.5$
No
Yes
$\eta_{\mathrm{HPT}} = 0.862$
$\eta_{\mathrm{HPT}} = 0.875$
3
$\Delta\mathrm{n}_{\mathrm{HPTcorr}} = \lambda_{\mathrm{HPT}}/\lambda_{\mathrm{HPT0}} - 1$
$\Delta\pi_{\mathrm{HPT}} = \pi_{\mathrm{HPT}}/\pi_{\mathrm{HPT0}} - 1$
$\mathrm{A}_{\mathrm{HPT}} = \mathrm{A}_{\mathrm{HPT}}\ (1 + \mathrm{A}_{46}\Delta\mathrm{n}_{\mathrm{HPTcorr}} + \mathrm{A}_{47}\Delta\pi_{\mathrm{HPT}} + \mathrm{A}_{48})/98066.5$
$\mathrm{E}_{\mathrm{T}} = 1 + 1.5 \times 10^{-5}(\mathrm{T}_{22} * 0.9 - 288.15)$
$\tau_{\mathrm{HPT}} = {\mathrm{E}_{\mathrm{T}}}^{2}$
$\mathrm{A}_{\mathrm{HPT}} = \mathrm{A}_{\mathrm{HPT}} * \tau_{\mathrm{HPT}}$
$\eta_{\mathrm{HPT}} = \eta_{\mathrm{HPT}}(1 + \mathrm{A}_{49}\Delta\mathrm{n}_{\mathrm{HPTcorr}} + \mathrm{A}_{50}\Delta\pi_{\mathrm{HPT}} + \mathrm{A}_{51})$

Figure 0.38 Program “XTVD” (HPT parameters computation).

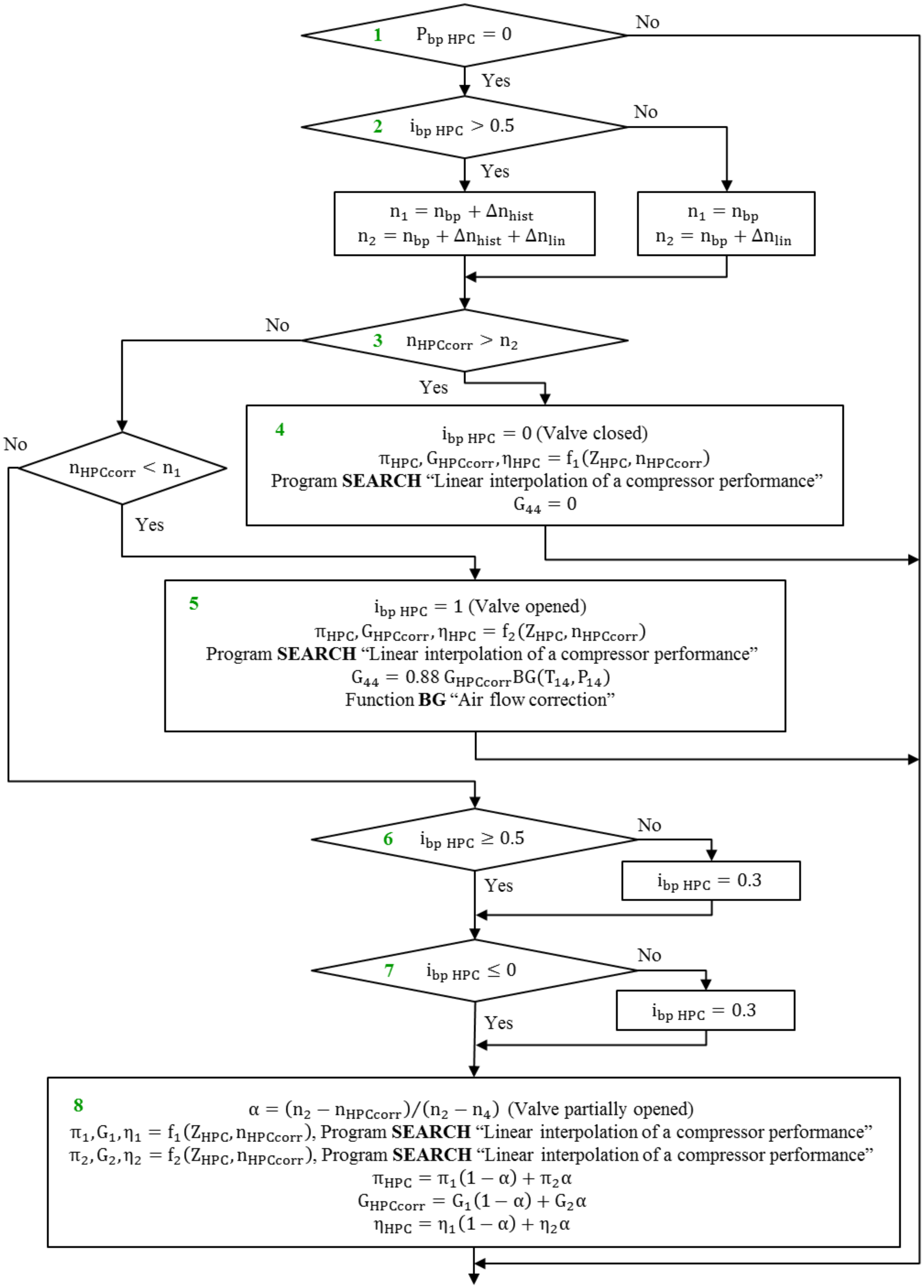

1 $P_{bp\ HPC} = 0$
No
Yes
2 $i_{bp\ HPC} > 0.5$
No
Yes
$n_1 = n_{bp} + \Delta n_{hist}$
$n_2 = n_{bp} + \Delta n_{hist} + \Delta n_{lin}$
$n_1 = n_{bp}$
$n_2 = n_{bp} + \Delta n_{lin}$
No
3 $n_{HPCcorr} > n_2$
Yes
4
$i_{bp\ HPC} = 0$ (Valve closed)
$\pi_{HPC}, G_{HPCcorr}, \eta_{HPC} = f_1(Z_{HPC}, n_{HPCcorr})$
Program **SEARCH** "Linear interpolation of a compressor performance"
$G_{44} = 0$
No
$n_{HPCcorr} < n_1$
Yes
5
$i_{bp\ HPC} = 1$ (Valve opened)
$\pi_{HPC}, G_{HPCcorr}, \eta_{HPC} = f_2(Z_{HPC}, n_{HPCcorr})$
Program **SEARCH** "Linear interpolation of a compressor performance"
$G_{44} = 0.88\ G_{HPCcorr} BG(T_{14}, P_{14})$
Function **BG** "Air flow correction"
6 $i_{bp\ HPC} \geq 0.5$
No
$i_{bp\ HPC} = 0.3$
Yes
7 $i_{bp\ HPC} \leq 0$
No
$i_{bp\ HPC} = 0.3$
Yes
8
$\alpha = (n_2 - n_{HPCcorr})/(n_2 - n_4)$ (Valve partially opened)
$\pi_1, G_1, \eta_1 = f_1(Z_{HPC}, n_{HPCcorr})$, Program **SEARCH** "Linear interpolation of a compressor performance"
$\pi_2, G_2, \eta_2 = f_2(Z_{HPC}, n_{HPCcorr})$, Program **SEARCH** "Linear interpolation of a compressor performance"
$\pi_{HPC} = \pi_1(1 - \alpha) + \pi_2\alpha$
$G_{HPCcorr} = G_1(1 - \alpha) + G_2\alpha$
$\eta_{HPC} = \eta_1(1 - \alpha) + \eta_2\alpha$

Figure 0.39 Program "XKVD" (HPC parameters computation).

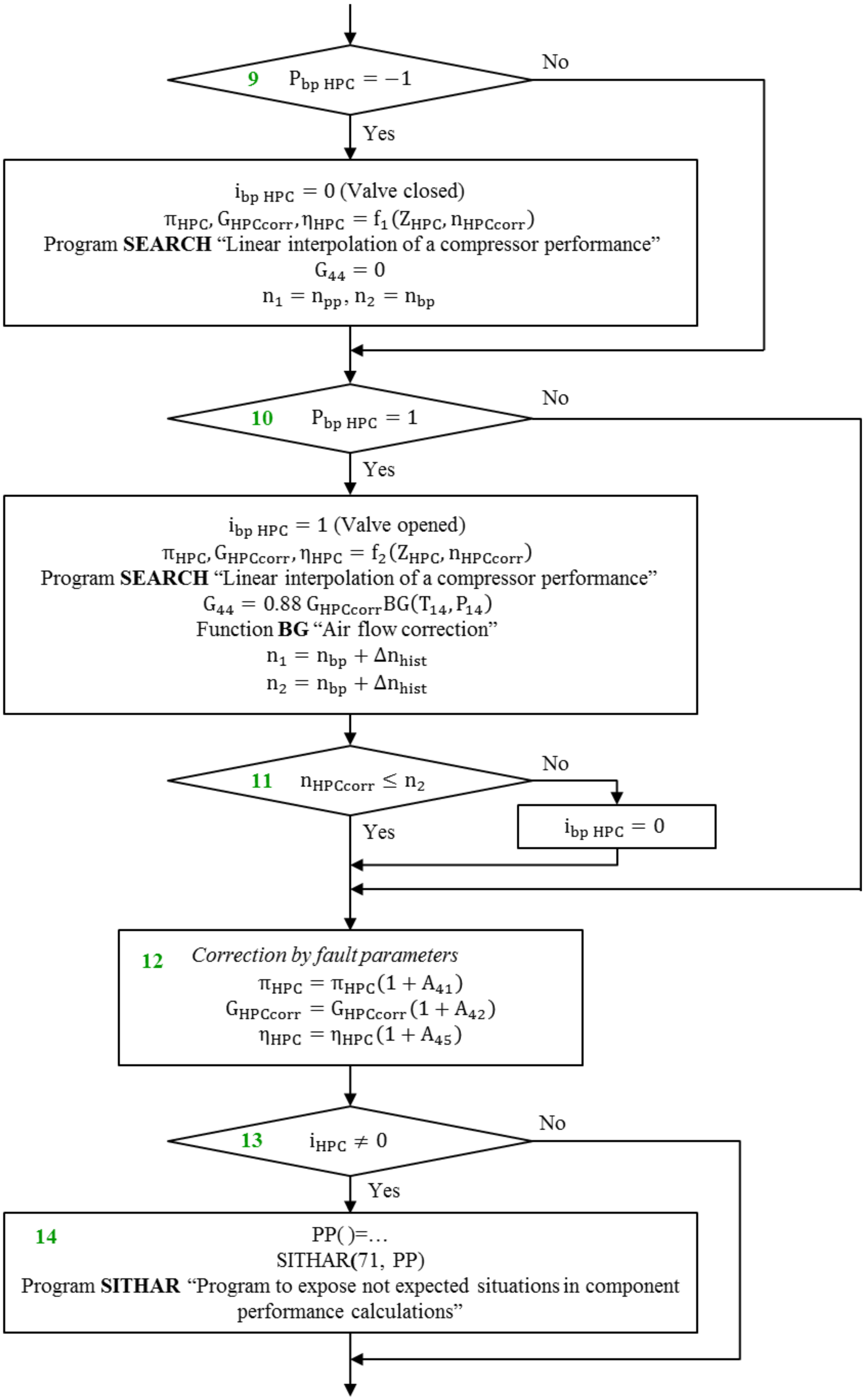

9 $P_{bp\ HPC} = -1$
No
Yes
$i_{bp\ HPC} = 0$ (Valve closed)
$\pi_{HPC}, G_{HPCcorr}, \eta_{HPC} = f_1(Z_{HPC}, n_{HPCcorr})$
Program **SEARCH** "Linear interpolation of a compressor performance"
$G_{44} = 0$
$n_1 = n_{pp}, n_2 = n_{bp}$
10 $P_{bp\ HPC} = 1$
No
Yes
$i_{bp\ HPC} = 1$ (Valve opened)
$\pi_{HPC}, G_{HPCcorr}, \eta_{HPC} = f_2(Z_{HPC}, n_{HPCcorr})$
Program **SEARCH** "Linear interpolation of a compressor performance"
$G_{44} = 0.88\ G_{HPCcorr} BG(T_{14}, P_{14})$
Function **BG** "Air flow correction"
$n_1 = n_{bp} + \Delta n_{hist}$
$n_2 = n_{bp} + \Delta n_{hist}$
11 $n_{HPCcorr} \leq n_2$
No
$i_{bp\ HPC} = 0$
Yes
12 *Correction by fault parameters*
$\pi_{HPC} = \pi_{HPC}(1 + A_{41})$
$G_{HPCcorr} = G_{HPCcorr}(1 + A_{42})$
$\eta_{HPC} = \eta_{HPC}(1 + A_{45})$
13 $i_{HPC} \neq 0$
No
Yes
14 PP( )=…
SITHAR(71, PP)
Program **SITHAR** "Program to expose not expected situations in component performance calculations"

Figure 0.40 Program "XKVD" (Continuation).

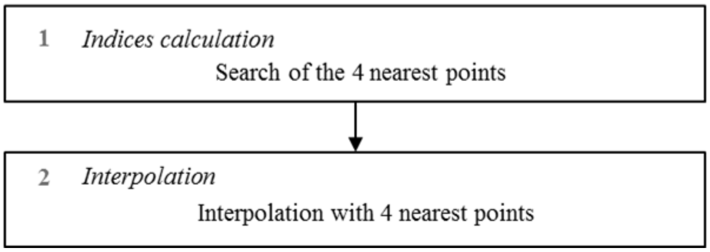

1 *Indices calculation*
Search of the 4 nearest points
2 *Interpolation*
Interpolation with 4 nearest points

Figure 0.41 Program “LINXYT” (Turbine performance interpolation).

41. LAM─ Velocity coefficient computation

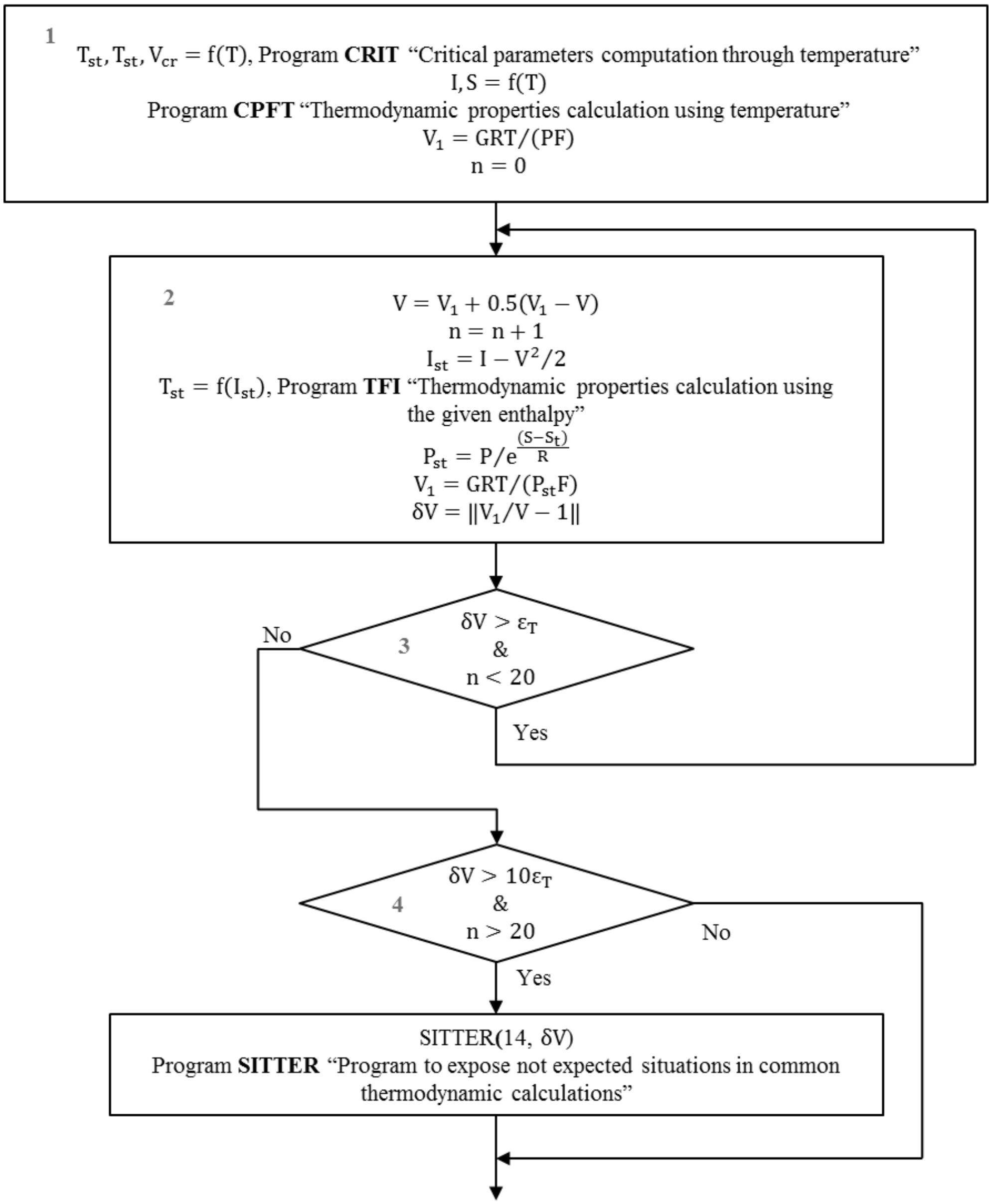

1
$T_{st}, T_{st}, V_{cr} = f(T)$, Program **CRIT** “Critical parameters computation through temperature”
$I, S = f(T)$
Program **CPFT** “Thermodynamic properties calculation using temperature”
$V_1 = GRT/(PF)$
$n = 0$
2
$V = V_1 + 0.5(V_1 - V)$
$n = n + 1$
$I_{st} = I - V^2/2$
$T_{st} = f(I_{st})$, Program **TFI** “Thermodynamic properties calculation using the given enthalpy”
$P_{st} = P/e^{\frac{(S-S_t)}{R}}$
$V_1 = GRT/(P_{st}F)$
$\delta V = \|V_1/V - 1\|$
No
3
$\delta V > \varepsilon_T$
&
$n < 20$
Yes
$\delta V > 10\varepsilon_T$
4
&
$n > 20$
No
Yes
SITTER(14, δV)
Program **SITTER** “Program to expose not expected situations in common thermodynamic calculations”

Figure 0.42 Program “LAM” (Velocity coefficient computation).

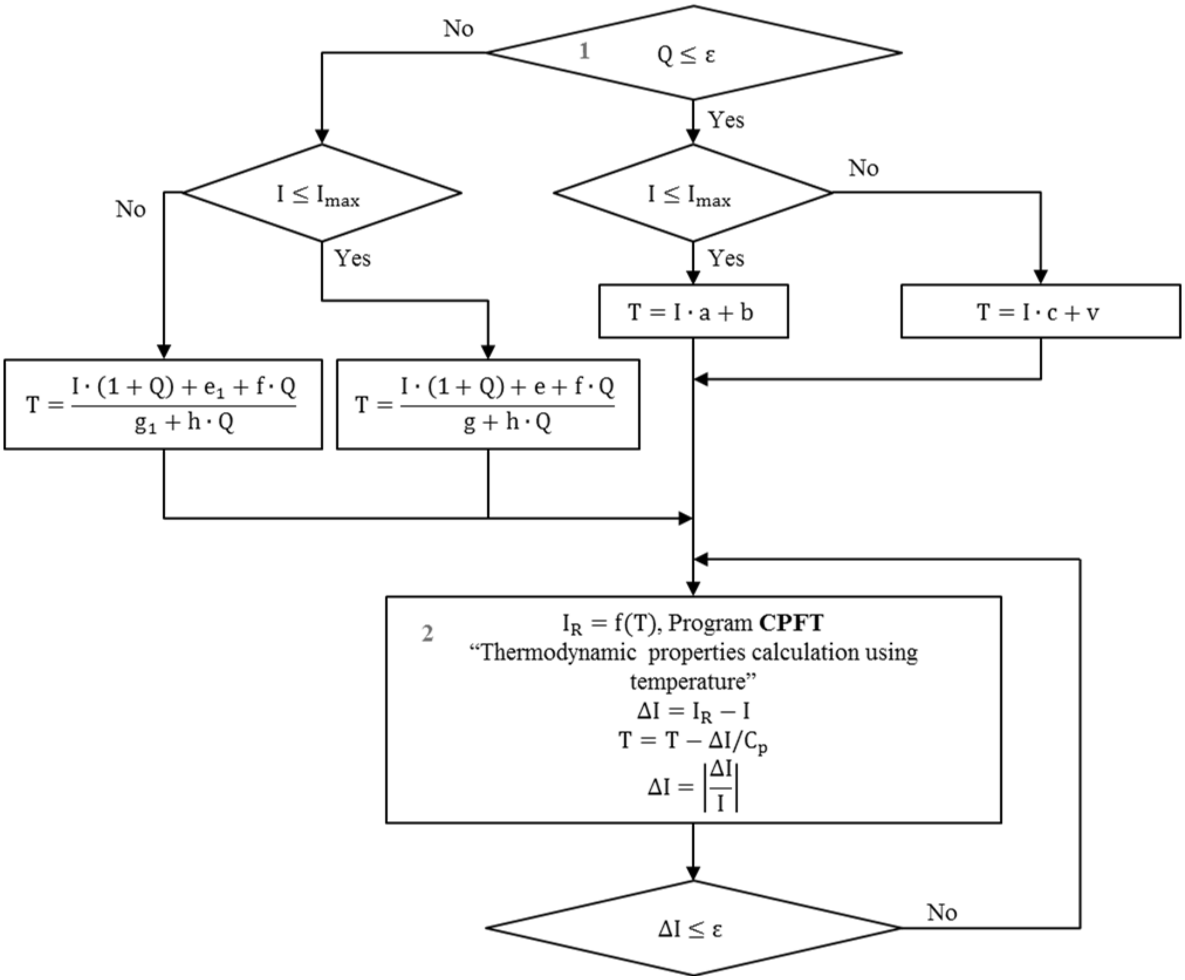

1
$Q \leq \varepsilon$
No
Yes
$I \leq I_{max}$
No
Yes
$I \leq I_{max}$
Yes
No
$T = I \cdot a + b$
$T = I \cdot c + v$
$T = \frac{I \cdot (1+Q) + e_1 + f \cdot Q}{g_1 + h \cdot Q}$
$T = \frac{I \cdot (1+Q) + e + f \cdot Q}{g + h \cdot Q}$
2
$I_R = f(T)$, Program **CPFT**
"Thermodynamic properties calculation using temperature"
$\Delta I = I_R - I$
$T = T - \Delta I / C_p$
$\Delta I = \left|\frac{\Delta I}{I}\right|$
$\Delta I \leq \varepsilon$
No

Figure 0.43 Program "TFI" (Thermodynamic properties calculation using the given enthalpy).

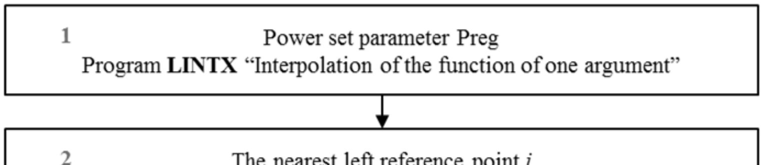

1
Power set parameter Preg
Program **LINTX** "Interpolation of the function of one argument"
2
The nearest left reference point j
3
Power set parameter time derivative
$Preg' = \frac{P_{reg} - P_{reg_j}}{t - t_j}$

Figure 0.44 Program "PARREG" (Power set parameter calculation in dynamics).

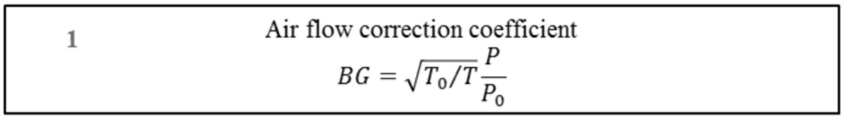

1
Air flow correction coefficient
$BG = \sqrt{T_0/T}\frac{P}{P_0}$

Figure 0.45 Program "BG" (Air flow correction).

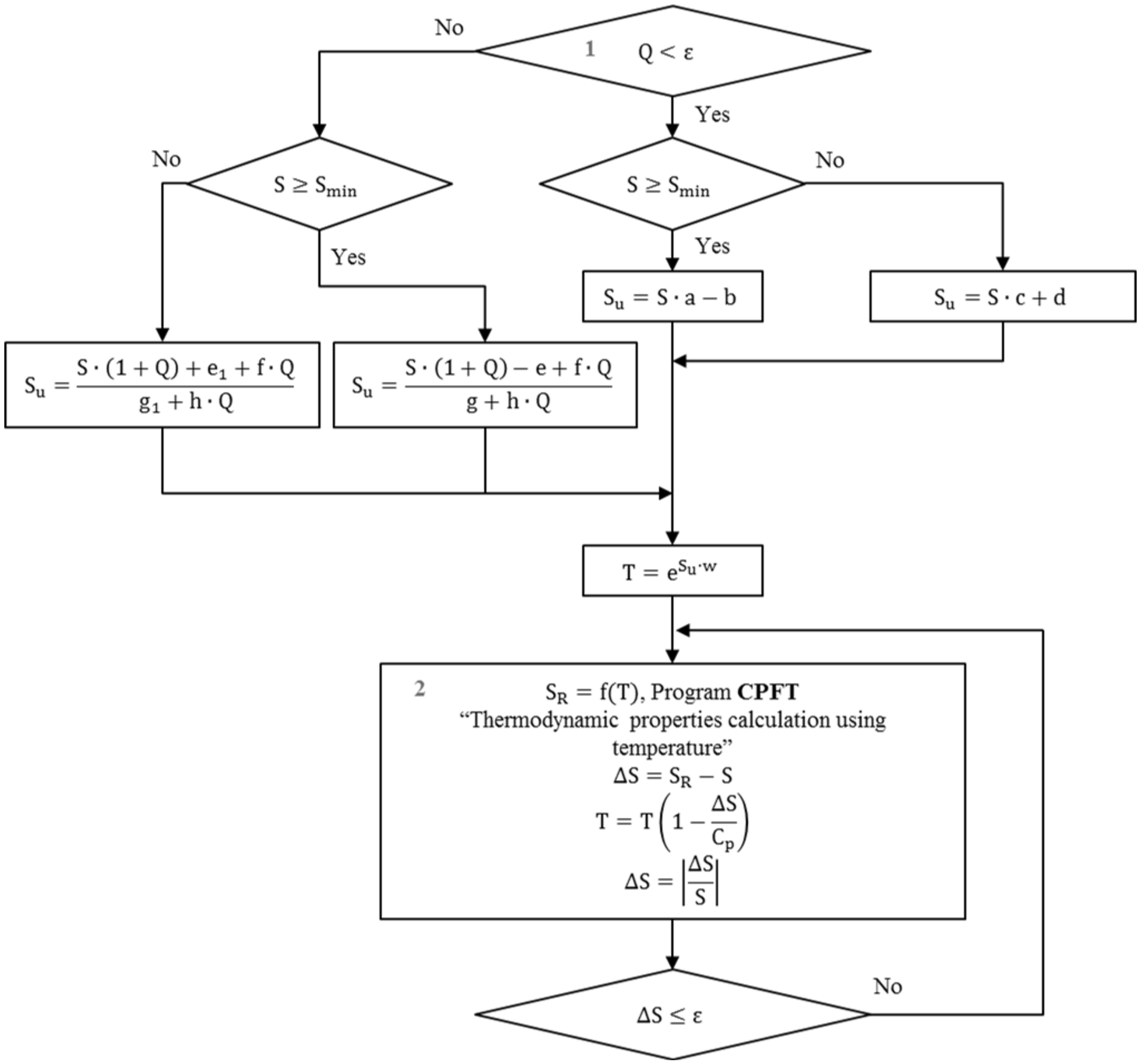

No
1 $Q < \varepsilon$
Yes
No
$S \geq S_{min}$
Yes
$S \geq S_{min}$
No
Yes
$S_u = S \cdot a - b$
$S_u = S \cdot c + d$
$S_u = \frac{S \cdot (1+Q) + e_1 + f \cdot Q}{g_1 + h \cdot Q}$
$S_u = \frac{S \cdot (1+Q) - e + f \cdot Q}{g + h \cdot Q}$
$T = e^{S_u \cdot w}$
2 $S_R = f(T)$, Program **CPFT**
“Thermodynamic properties calculation using temperature”
$\Delta S = S_R - S$
$T = T\left(1 - \frac{\Delta S}{C_p}\right)$
$\Delta S = \left|\frac{\Delta S}{S}\right|$
No
$\Delta S \leq \varepsilon$

Figure 0.46 Program “TFS” (Thermodynamic properties calculation using the given entropy).

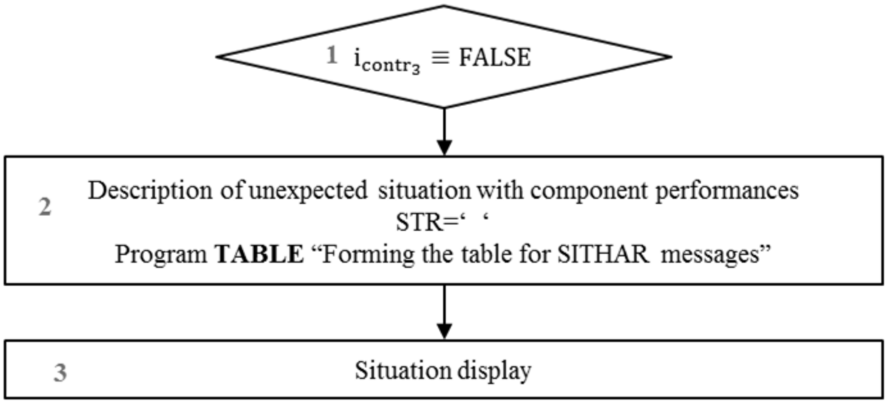

1 $i_{contr_3} \equiv$ FALSE
2 Description of unexpected situation with component performances
STR=‘ ‘
Program **TABLE** “Forming the table for SITHAR messages”
3 Situation display

Figure 0.47 Program “SITHAR” (Program to expose not expected situations).

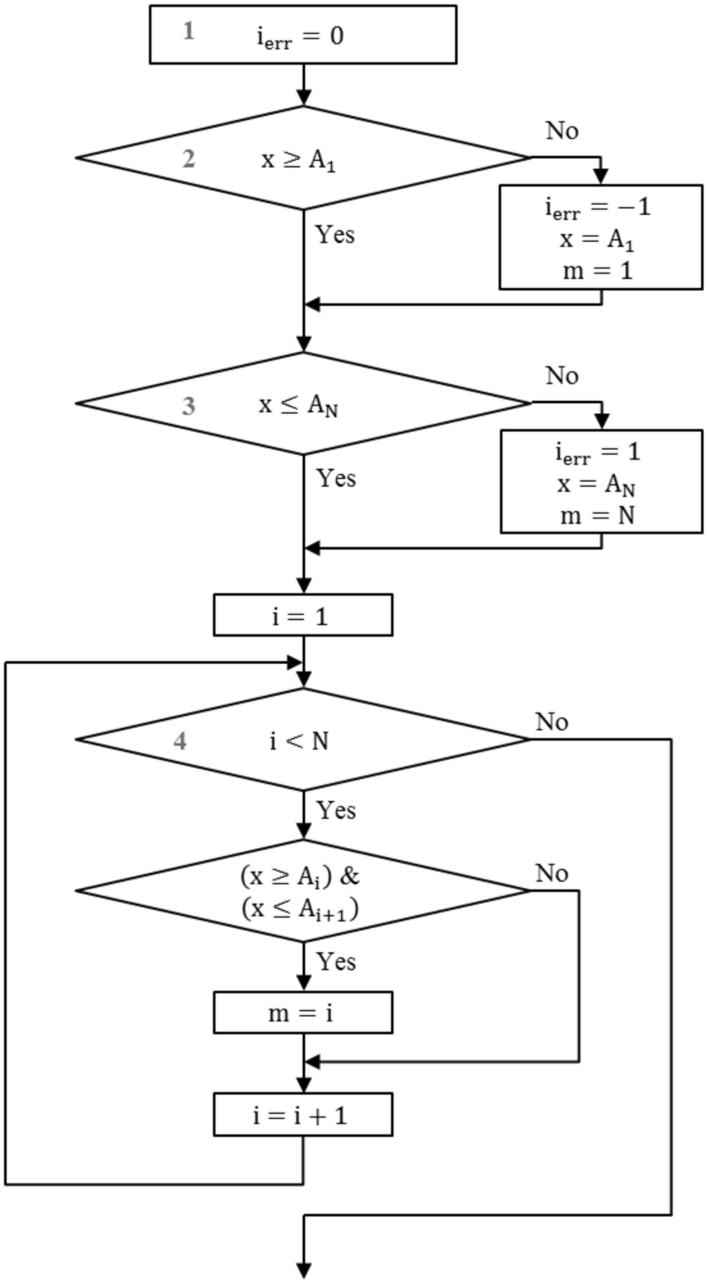


Figure 0.48 Program “SEAR” (Searching the table interval where the given value is situated).

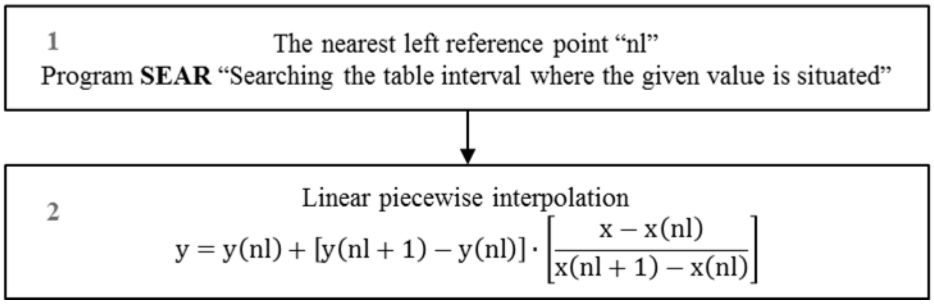

1
The nearest left reference point “nl”
Program **SEAR** “Searching the table interval where the given value is situated”
2
Linear piecewise interpolation
$y = y(nl) + [y(nl+1) - y(nl)] \cdot \left[\frac{x - x(nl)}{x(nl+1) - x(nl)}\right]$

Figure 0.49 Program “LINTX” (Interpolation of the function of one argument).

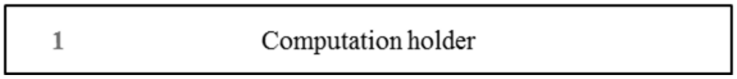


Figure 0.50 Program “HOLDER” (Computation holder).

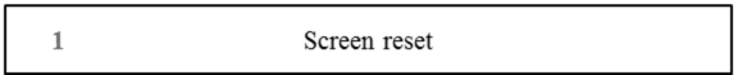


Figure 0.51 Program “RESET” (Screen reset).

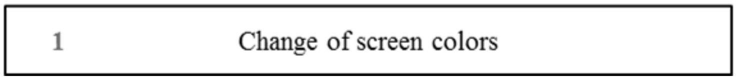


Figure 0.52 Program “WHIRED” (White symbols, red background).

52. CLS– Screen cleaning

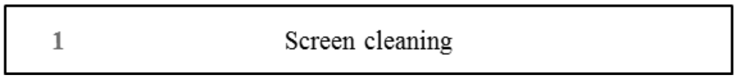


Figure 0.53 Program “CLS” (Screen cleaning).

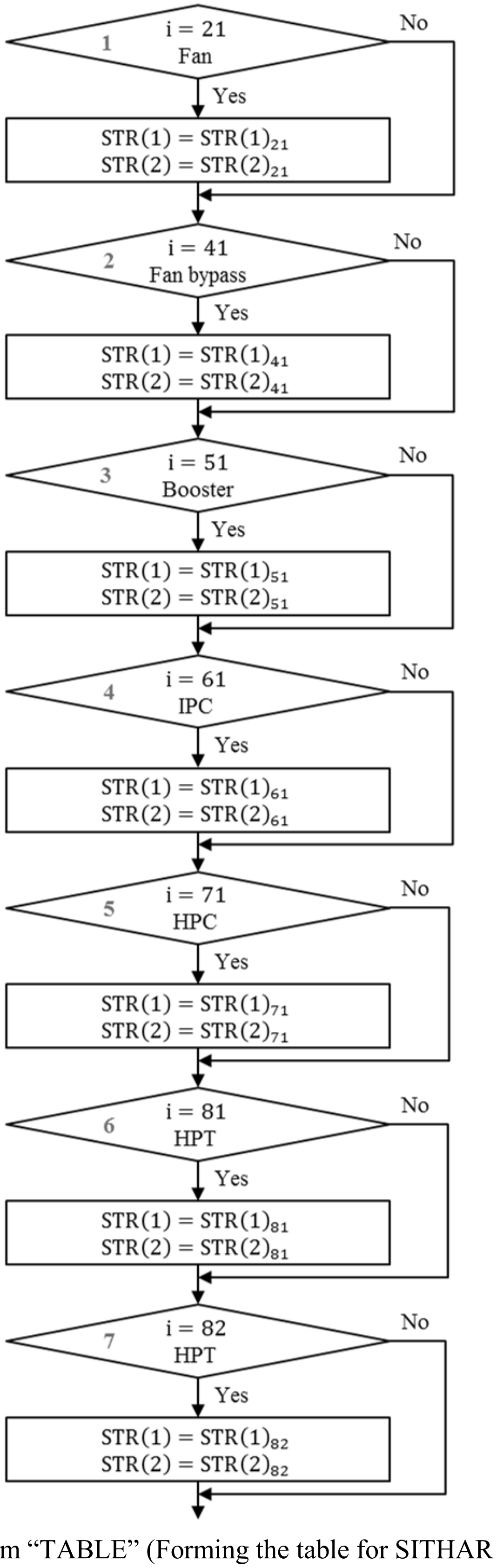


Figure 0.54 Program “TABLE” (Forming the table for SITHAR messages).

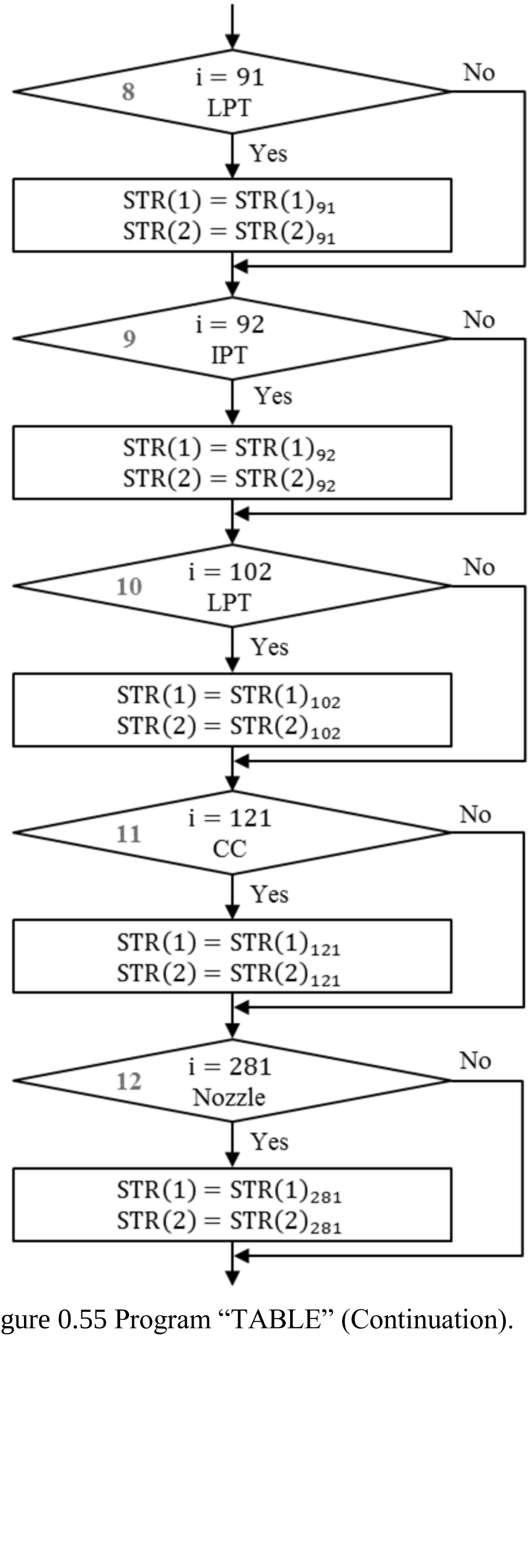


Figure 0.55 Program “TABLE” (Continuation).