Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Measurement

Distributed Signal Processing in Wireless Sensor Networks for Diagnostics

DOCTORAL THESIS

Jan Neužil

Supervisor:	doc. Ing. Radislav Šmíd, PhD.
Study programme:	Electrical Engineering and Information Technology
Branch of study:	Measurement and Instrumentation

2016

Acknowledgement

There are a number of people without whom this thesis might not have been written, and to whom I am greatly indebted.

First and foremost, I have to thank my family for their love and great support.

I would like to give my sincerely thanks to my supervisor, Associate Prof. Radislav Šmíd, PhD., for his guidance and support throughout my studies, and especially for his confidence in me.

I declare that I carried out this doctoral thesis independently, and only with the cited sources, literature and other professional sources.

In Prague date .... signature of the author

Název práce: Distribuované zpracování signálů pomocí bezdrátových senzorových sítí pro diagnostiku

Autor: Ing. Jan Neužil

Katedra: Katedra měření

Vedoucí disertační práce: doc. Ing. Radislav Šmíd, Ph.D.

Abstrakt:
Tato práce přináší nový přístup pro monitorování leteckých motorů pomocí distribuovaných bezdrátových senzorových sítí, které zpracovávají vibrační signály.

Navržený přístup umožňuje použití výpočetně náročných metod signálové analýzy ve výpočetně slabých ulzlech bezdrátové senzorové sítě.

Dále přináší metodu detekce vad, která je založená na detekci neobvyklého chování, a zároveň je vykonávána přímo v uzlu senzorové sítě.

Klíčové vlastnosti architektury navrženého systému: adaptivita, rekonfigurabilita a tři fáze operace, umožňují dosáhnout okamžitého odhalení vady a zároveň zajišťují dlouhodobý provoz monitorovacího systému.

Navržené metody a architektura bezdrátového monitorovacího systému byly ověřeny experimentálně s využitím reálného proudového motoru.

Klíčová slova:
Detekce vad, Klasifikace, Sledování stavu strojů, Bezdrátové senzorové sítě

Title: Distributed Signal Processing in Wireless Sensor Networks for Diagnostics

Author: Ing. Jan Neužil

Department: Department of Measurement

Supervisor: doc. Ing. Radislav Šmíd, Ph.D.

Abstract:
This work introduces the on-board vibrational condition monitoring of aircrafts powerplants by its original novel approach of distributed WSN based vibrational condition monitoring.

Proposed approach allows to employ computationally intensive methods of vibrational signal processing and methods of condition monitoring in computationally weak wireless sensor network.

It introduces the fault detection method based on novelty detection which is executed directly in wireless sensor nodes.

The novel framework of WSN condition monitoring system with its key attributes Adaptivity, Reconfigurability and Three phases of operation enables the immediate fault detection capability while providing long-term monitoring.

The proposed methods and framework were evaluated by the means of experiment on the small turbojet engine.

Keywords:
Fault Detection, Classification, Machine Condition Monitoring, Wireless Sensor Networks

List of Figures

List of Tables

Acronyms

ACARS Aircraft Communications Addressing and Reporting System

ADC Analog to Digital Converter

APU Auxiliary Power Unit

BFM Band-Pass Mesh Signal

CBM Condition Based Maintenance

CM Condition Monitoring

CM Condition Monitoring

COTS Commercial of-the-shelf devices

CSMA/CA Carrier Sense Multiple Access with Collision Avoidance

CWD Choi-Williams Distributions

DFSS Direct Sequence Spread Spectrum

DIF Difference signal

DLL Data Link Layer

DSSS Direct Sequence Spread Spectrum

EEPROM Electrically Erasable Programmable Read-Only Memory

EHM Engine Health Management

FD Fault Detection

FDD Fault Detection and Diagnosis

FFT Fast Fourier Transform

HART Highway Addressable Remote Transducer Protocol

HRFT High-Frequency Resonance Technique

HT Hard-Time

HUMS Health and Usage Monitoring System

IoT Internet of Things

ISM	Industrial Scientific and Medical
IWSN	Industrial Wireless Sensor Network
LLC	Logical Link Control
MAC	Medium Access Control
MCM	Machine Condition Monitoring
MCU	Microcontroller Unit
OC	On-Condition
OSI	Open Systems Interconnection
OTAP	Over-the-Air-Programming
PC	Personal Computer
PHY	Physical Layer
PTMD	Predictive Trend Monitoring and Diagnostic
RAM	Random Access Memory
RES	Residual Signal
RFID	Radio Frequency Identification
RoC	Radio-on-the-Chip
Rx	Receive
SoC	System-on-the-chip
STFT	Short-Time Fourier Transform
TSA	Time Synchronous Averaged Signal
Tx	Transmit
UWB	Ultra-Wide Band
WSN	Wireless Sensor Network
WT	Wavelet Transform
WVD	Winger-Ville distributions

Contents

Chapter 1

Introduction

In recent years there is evident effort of aviation industry to reduce costs of aircraft operation and maintenance by employing on-board condition monitoring systems. The main purpose of such systems is to prevent unscheduled maintenance and thus prevent aircraft downtime by transforming unscheduled maintenance to scheduled. Such an example is Health and Usage Monitoring System which is the system monitoring helicopter's drivetrain focusing mainly on its vibrational behavior. Another case is the Engine Health Management system introduced by Rolls Royce which serves to monitor turbofan engines of large passenger aircraft. Although the potential of on-board condition based maintenance for aircraft industry is great there are still only few applications. This is caused mainly by the difficulties connected with introducing on-board monitoring system especially if new sensors, as vibration sensors, are needed.

If the implementation of on-board condition monitoring system were easier and cheaper the next very profitable application could be for example world wide fleets of turboprop engines operated in long-term service agreements.

The above mentioned implementation difficulties may be overcome by taking an advantage of Wireless Sensor Networks (WSNs). WSNs have a broad use in machine condition monitoring and generally in condition based maintenance. Their main benefit is in easy installation in areas where a classical wired system is not applicable or is inconvenient. WSN application can save significant resources that would be spent in heavy-duty wired connection. They usually create a distributed system with more sensing points than in a classical wired centralized system. Thus it brings better resistance to unfavorable effects and better robustness.

Addressing the gap in on-board maintenance of aircraft systems not covered by current applications, the aim of this work is to examine the potential of WSNs for aircraft powerplant on-board vibrational monitoring.

Chapter 2

State of the art

2.1 Condition monitoring

Condition monitoring is the process of determining the monitored object condition while in operation. Condition monitoring is closely tied with the maintenance discipline. In [1] three general maintenance strategies for wide variety of industries are identified:

1. Corrective maintenance – Run to failure
2. Preventive maintenance – Time-Based
3. Predictive maintenance – Condition Based Maintenance (CBM)

In the first kind a machine is run until it breaks down. While it gives the longest time between shutdown, it may result in severe consequential damage or catastrophic scenario. Also the unscheduled repair increases maintenance time.

The second scenario is based on regular planned maintenance intervals which are shorter than expected time between failures. While catastrophic failure is greatly reduced, excessive number of replacement components are consumed.

In the last approach the potential failure is predicted through regular condition monitoring and maintenance is carried out at the optimal time. Condition monitoring typically comprises methods of fault detection, fault diagnosis and fault prediction [2]. In [1] it is recognized that vibration analysis is the most prevalent method for machine condition monitoring.

2.1.1 Maintenance of aircraft systems and powerplant

Although the maintenance of aircraft is just a subset of general machinery maintenance it is very specific branch with its own strict rules, while the similarities with the general maintenance strategies mentioned above still may be found.

Primary maintenance processes for aircraft systems and powerplant defined by Maintenance Steering Group–2nd Task Force (MSG-2) are:

- Hard-Time (HT),
- On-Condition (OC)
- Condition Monitoring (CM)

Hard-time maintenance is a preventive primary process that requires a system, component or appliance to be overhauled periodically at fixed time limits or removed from service (life limit) [3].

On-condition maintenance is also preventive process that requires that system, components, or appliance to be inspected periodically to determine if it can continue in service. The standard of inspection ensures that unit is removed from service before failure during normal operation [3].

Condition Monitoring (CM) introduced by MSG-2 is the maintenance process that have neither HT or OC. The user must control the reliability of systems or equipment based on knowledge gained by the analysis of failures or other indication of deteriorations [3].

Above defined primary maintenance processes are further developed and extended within methodology defined by Maintenance Steering Group–3rd Task Force (MSG-3). MSG-3 methodology enhances and changes MSG-2 approach by rather task-oriented approach than maintenance-process-oriented approach [3]. For this approach the availability of on-board condition monitoring system is extremely beneficial because it broadens the knowledge about the historical and actual system condition. Thus improves better planning within the task-oriented approach.

For that reason there is great effort to introduce on-board health monitoring systems which perform permanent condition monitoring.

2.1.2 Aircraft CM examples

Health and Usage Monitoring System (HUMS) is on-board vibration-monitoring system for continuously monitoring vibrations of helicopter drivetrain, which interfaces to hardwired vibration and tachometer sensors [4]. Several companies like Eurocopter, GE Aviation, Goodrich and Honeywell have their own HUMS products. A study of Honeywell HUMS [5] employed on AH-64 Apaches found 30% reduction in mission aborts, 20% reduction in maintenance test flights and 5–10% reduction in scheduled maintenance.

Rolls Royce uses Engine Health Management (EHM) to track fleet of Trent engines using onboard sensors which monitor engines' temperatures, pressures, speeds and vibrations. Acquired sensor data snapshots and captured summaries are transfered through digital data-link Aircraft Communications Addressing and Reporting System (ACARS). Then the data are on-ground analyzed by Controls and Data services [6].

Honeywell also introduced Predictive Trend Monitoring and Diagnostic (PTMD) service for Auxiliary Power Unit (APU) condition based maintenance. PTMD monitors exhaust gas temperature, oil temperature, inlet pressures and starting trends. Acquired data are transfered via ACARS for on-ground analysis. Customers using PTMD could reduce overall life cycle cost on average by 15% [7].

Main benefits of above mentioned systems are:

- Increase safety of both individual aircraft and whole fleet.
- Improve dispatch availability and thus avoid down time.
- Reduce costs for maintenance like inventory management and logistics.

- Could prolong life time of monitored component.
- Could improve maintenance costs estimation especially in Long Term Service Agreement business model.

2.2 Wireless Sensor Networks

A Wireless Sensor Network consist of nodes which are capable of sensing a certain physical phenomena and also are capable of wireless radio communicating to share gained information.

The term WSN often overlaps with modern trends as Smart Sensors, Internet of Things (IoT), Database of things or Radio Frequency Identification (RFID).

The initial impulse for WSN evolution came from military background in Cold-War era as systems for submarine surveillance and networks of defense radars [8]. Since 1990s the WSNs have been asserted also in commercial applications.

The rapid progress of both technologies and standards in sensors, microcontrollers, radio communication and power systems gave rise to WSNs expansion.

Gay: "We do not expect new technology to remove these limitations: the benefits of Moore's Law will be applied to reduce size and cost, rather than increase capability." [9].

Also the standardization is a key issue for success of WSN markets while it simplifies the design procedurem reduces costs of production and significantly increases the compatibility of WSN devices from different vedors [10].

Typical features of WSN: restrictions and requirements

- Processing resources
  Computing capability and memory resources of a single node are restricted. A programming paradigm has to adapt to that.
- Energy resources
  A node is generally self-powered and has to usually operate unattended for the whole life-time. A typical source of power are batteries, which can be combined with an energy harvesting system.
- Communication capability
  Nodes of WSN are typical by a low data throughput and short communication range. While a short communication range can be overcame by multi-hop routing, a low data throughput can be surmounted by in-node signal processing, i.e. data compression.
- Cost, size
  A general requirement for a WSN is low cost and small size of a single node, which leads to the use of miniaturized mass produced Commercial off-the-shelf (COTS) devices.
- Application, reliability
  Great advantage of WSNs is ease of deployment due to no constraints caused

by wires. It brings a profit in scalability and thanks to possibly high number of nodes also redundancy. Especially in industrial processes there are strict requirements for reliability, security and data integrity.

The above mentioned features outline the specific approach which is necessary for development and use of Wireless Sensor Networks. This interdisciplinary approach combines skills and knowledge in a wide range of areas like measurement theory, signal processing, embedded systems programming, wireless communication or analog circuity design. In-depth understanding of monitored system is also crucial, because the development of WSN has to be adapted to its special needs. Primarily a trade-off between raw data transmitting and complex in-node computation related to available resources has to be found. Generally in-node computation and thus the data compression brings the great energy-efficient benefit compared to raw data transmission [11].

2.2.1 WSN Hardware

The node is an embedded system composed of a microcontroller, sensors, radio transceiver and power source as it is depicted in Figure 2.1.

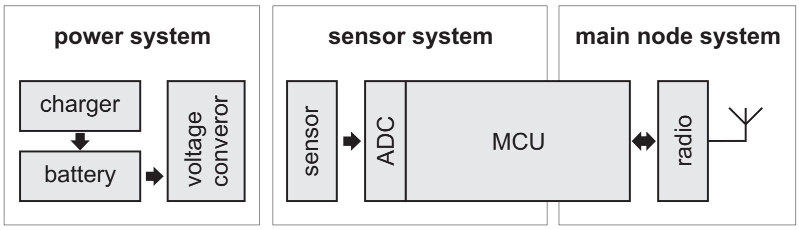

Figure 2.1: Concept of a node

Further see the description of components important for WSN monitoring system implementation.

Microcontroller Unit (MCU)

The core part of a node is a microcontroller unit (MCU). Microcontrollers employed for WSN can be divided into the following classes depending on their processing power and with that related power consumption (see also the table 2.1):

- Low performance class – Usually 8-bit microcontroller with memory (both for program and data) less than 400 KB. This class is represented for example by Microchip PIC16LF which is equipped only by 28 KB FLASH memory, 1 KB RAM and 256 B EEPROM and it consumes about 0.1 µW in sleep mode and 77 µW in operating mode. This microcontroller is capable to handle a basic node's behavior, i.e. to sense an individual value, send and receive message and to control the performance regime (sleep or active).

- Middle performance class – Usually 8-bit or 16-bit microcontrollers operating at 8 MHz equipped with total memory (both for program and data) around 1 MB and power consumption up to 25 mW in active mode and 25 µW in sleep mode. Into this class belong Atmel ATmega 1281 and Texas Instrument

MSP430. In this class it is possible to maintain faster samples collection and signal processing, however the methods for data analysis has to be selected with care due to the restricted memory size.

- High performance class – Into this class fall 32-bit microcontrollers with operating frequency up to hundreds on MHz. In the comparison of the former classes, there is no limitation in memory size which is usually in tens of MB, although a data memory is usually peripheral. The maximum power consumption could be up to 300 mW at the maximum frequency, however these microcontrollers provide a set of power regimes which enable a sophisticated control of MCU processing power and energy consumption. This MCUs enable even very sophisticated signal processing or historical data storing. Typical representatives of this class are NXP LPC175x and STM32L1 both based on Cortex M3 architecture or older Intel (Marvell) Xscale PXA271.

Table 2.1: Performance classes of MCU

Performance class	MCU architecture	Memory (program + data)	Consumption sleep /active	Typical representative
Low	8-bit	400 KB	0.1 μW / 100 μW	PIC16LF
Middle	8-bit, 16-bit	1 MB	25 μW / 25 mW	ATmega1281, MSP430
High	32-bit	tens-hundreds MB (external)	1 μW / 300 mW	STM32L, PXA271

Up to date trend is integration of a node's parts into one single chip, where MCU is a core. This is step forward, which simplifies design and usage of WSNs. Based on the degree of integration there is:

- Radio-on-the-Chip (RoC) – Newly are emerging MCUs which integrated radio module directly in one chip. As an example could be mentioned STM32W which is embedded with a 2.4 GHz IEEE 802.15.4 radio.

- System-on-the-chip (SoC) – It is a concept where all core parts of a node (battery/energy harvester, sensor, MCU, radio) are integrated on a one single chip. As far as the author knows there is not such commercial product available on the market, but one could expect its arise in a near future.

Energy

Use only batteries in WSNs as a source of energy may be complicated. Periodical replacement of batteries is usually unfeasible. Thus an energy harvesting (energy scavenging) technique which extract the energy from the ambient environment surrounding nodes are needed. Table 2.2 compares some of the most promising techniques. Work [12] brings detailed survey of the state-of-the-art energy harvesting techniques.

For industrial environments the most suitable energy sources are the thermoelectric and vibrations. Energy harvesting form continually vibrating machines (like

Table 2.2: Comparison of energy harvesting techniques [13]

Source	Performance	Necessary dimension
Battery	2880 J/cm^3	–
Light (indoor)	10–100 μW/cm^2	59–590 cm^2
Airflow	0.4–1 mW/cm^3	6–15 cm^3
Vibrations	200–380 μW/cm^3	16–30 cm^3
Thermoelectric	40–60 μW/cm^2	98–148 cm^2
Electromagnetic radiation	0.2–1 mW/cm^2	60-30 cm^2

rotating machines) is the most efficient when the resonant frequency of a harvester matches a significant machine's vibration frequency. Energy from vibrations can be harvested using piezoelectric, electrostatic or electromagnetic conversion [14].

Energy harvesting device can be used as the single source of energy, but usually it is used in combination with a battery. In this combination of sources it is essential to predict a battery remaining life. Paper [15] deals with battery life evaluation.

Current WSN Platforms (Development kits)

A WSN platform is generally a complete WSN node as defined in 2.2.1 (Fig. 2.1) with a toolchain for programming and several examples of networking. Such set of nodes is directly prepared to be used for experiments, for prototype development or even for deployment in a real environment.

There is a large group of general-purpose platforms discussed in depth in [16] and [17]. Among typical representative of present-day platforms, which are suitable for development applications for industrial environment, belong: IRIS and LOTUS from Memsic [18] (IRIS was originally developed by Crossbow), radio-on-the-chip Ember from Silicon Labs or IQRF [19] from Microrisc [20]. See the comparison table 2.3.

For direct use in industrial areas there are also specific-purpose platforms, which are usually very restricted in the ability of individual programming and in the ability of rapid measurements. The typical example is NI WSN-3202 from National Instruments [21].

2.2.2 WSN programming

WSN is a special case of embedded system which is specific by limited resources (memory, computation, energy) and by the nature of processing of immediately incoming concurrent events, e.g. multiple sensor reading and radio message reading. Except the time, when the node is reacting on an event, it tends to be in a mode of low-power consumption (sleep mode).

Because of this characteristic of a WSN system it is necessary to adopt appropriate programming paradigm. The widespread **process-based** programming paradigm is based on concurrent (parallel) execution of multiple processes. This approach suffers in the case of relatively small tasks with respect to interruption routine

Table 2.3: Present-day WSN platforms

Platform	IQRF	IRIS	LOTUS	Ember	NI WSN-3203
Producer	Microrisc	Memsic (Cross-bow)	Memsic	Silicon Labs	National Instru-ments
MCU	8-bit PIC16LF	8-bit AT-mega1281	32-bit ARM Cortex-M3	32-bit ARM Cortex-M3	16-bit MSP430
Performance class (accord-ing to 2.2.1)	Low	Middle	High	High	Middle
Radio	External 433/868/ 916 MHz	External 2.4 GHz	External 2.4 GHz	**Integrated** 2.4 GHz	External 2.4 GHz
Purpose	General/ slow process monitoring	General	General	General	Industrial/ slow process monitoring
Programming	IQRF-OS	**TinyOS**	**TinyOS**	Ember App-Builder	LabVIEW

which handles the switching between them. This is the typical case of WSNs. In addition the interruption routine requires an extra space of memory [11].

The **event-based** programming, which is closer to the reactive WSNs' nature, is the other approach. It adopts following paradigm: The system waits for an event, when an event happens it is detected by an event handler. Event handler performs just a short set of instructions to store necessary information about an event. The actual processing of an incoming information is done outside the event handler in a task procedure. Such a short event handler can interrupt a task, but a task can not interrupt other task or an event handler. Evoked tasks are processed sequentially in the first-in-first-out order.

There is a number of operating systems (OS) which facilitates the development of WSN embedded systems. Such OS brings a programming model, which serves to control and manage resources (memory, processor, input/output), controls scheduling and also supports energy management and directly supports radio communication. Survey [22] examines and compares the state-of-the-art OS for WSNs: TinyOS, MANTIS, Nano-RK and LiteOS.

TinyOS is one of the most widely used OS for WSN's development. It is event-based open-source system, which core requires only 400 bytes of code and data memory. TinyOS was developed by UC Berkley and has been supported by a large community of users.
Among other TinyOS supports Over-the-Air-Programming (OTAP), multihop routing communication, sleep modes and time-synchronization.

Article [23] compares a general-purpose multi-tasking operating system and event-based TinyOS on the same embedded system. The results show that event-based system achieves an 8x improvement in performance, 2x and 30x improvement in instruction and data memory requirement, and a 12x reduction in power consumption.

2.2.3 WSN communication methods

Physical medium for communication – available frequency bands

For wireless communication among WSN are largely used so called ISM (Industrial, Scientific and Medical) frequency bands defined by International Telecommunication Union. Primarily ISM bands were established for ISM-licensed applications, which are usually non-communicating such as microwave ovens, diathermy machines or electric cookers. Secondly ISM bands are shared with license-free communication applications such as WSNs. That implies, that communicating devices has to be error-prone since there is no guarantee of electromagnetic interference. Most often is for WSNs utilized the ISM frequency band from 2.4 GHz to 2.5 GHz (further referred as 2.4 GHz).

Network topology

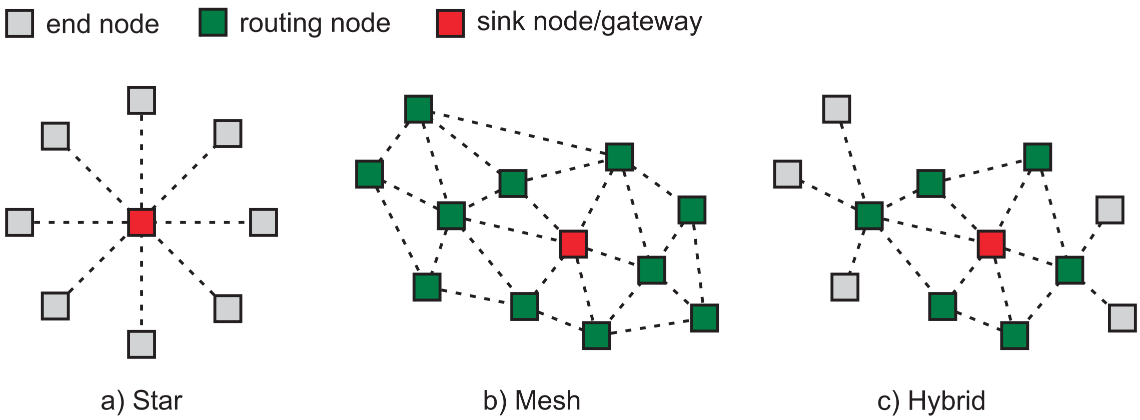

Figure 2.2: Network topology [11]

The simplest topology of a wireless sensor network is the star topology (Fig. 2.2a), where each node is connected directly to the sink node. The end-nodes are not able to pass data or commands between each other, but only with the sink node, which serves as an overall coordinator. The message from the end-node performs just a single-hop. When the connection from the end-node to the sink is broken (e.g. due to obstacle or interference), the communication is lost.

On the other hand in the topology called a mesh network (Fig. 2.2b) every node has the routing capability. The nodes can communicate between each other and they can multi-hop a message from one node to another until it reaches its destination. This brings the ability of self-organizing, self-healing, path redundancy and thus improving robustness of the network. A multi-hop principle allows a message to overcome large distance and obstacles and also improves the energy efficiency of communication, because the attenuation of radio signals is quadratic in relation to increasing distance (in most environments) [11]. However message multi-hopping in mesh networks brings a higher demands for routing algorithms.

An appropriate topology is always strongly depended on the type of application (i.e. size of a network, number of nodes or a kind of monitored object). Usually the WSN's topology can by created as a hybrid network (Fig. 2.2c), where are combined routing and non-routing nodes. The network can by organized in clusters or can be a tree topology with the backbone structure of routing nodes and end-nodes connected to them.

Routing protocols in a mesh network of WSN must optimize requirements especially for energy consumption and latency. These routing algorithms are specific by strong interactions between MAC and NWK layer. Chapter 11 of [24] brings a survey of routing protocols solutions proposed for WSNs deployed in industrial environments.

Standards and protocols for WSN

Since the year 2000 there has been significant development in the field of wireless communication. To ensure consistency and interoperability between developed applications, it is crucial to follow widely respected standards. Institute of Electrical and Electronics Engineers Standards Association (IEEE-SA) brings the family of standards IEEE 802 which specify the lower two layers of OSI reference model: Physical layer (PHY) and Data link layer (DLL), which is further divided in sub-layers called Logical Link Control (LLC) and Media Access Control (MAC).

Standard **IEEE 802.11 (Wi-Fi)**, operating at 2.4 GHz is primarily determined for Wireless Local Area Network (WLAN). Thanks to its wide expansion and high bitrate, it is often used as substitution of wire link also in industrial environments. However due to its high energy consumption and high price of devices using Wi-Fi, this standard is not suitable for Wireless Sensor Network communication, how it is defined in 2.2.3.

The group of standards IEEE 802.15 is specified for Wireless Personal Area Network (WPAN).

IEEE 802.15.1 (Bluetooth) operates also at 2.4 GHz and is intended for short range communication among mobile devices and accessories, for example handsfree set connected to a mobile phone. In comparison with above mentioned Wi-Fi it has much lower consumption, but still it is not very suitable for WSNs.

The most recent version Bluetooth 4.0 introduces the **Bluetooth Low Energy technology (BLE)**, which is aimed for healthcare, fitness, security and home entertainment applications.

International standards based on IEEE 802.15.4

IEEE 802.15.4 is designed for low-power, low data rate, short-range, and low-cost wireless communication. This meets requirements of WSNs. Within this standard only physical (PHY) and medium access control (MAC) layer of the OSI (Open Systems Interconnection) model is specified.

Physical layer of IEEE 802.15.4 uses Direct Sequence Spread Spectrum (DSSS) working at 868 MHz (only Europe), 915 MHz (only North America) or 2.4 GHz (worldwide) with a maximal data rate 250 kb/s. In MAC layer is employed Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). Network topology can be built as star or peer-to-peer, which enables mesh topologies implemented at higher layers of OSI model. More details may be found in [25].

Amendment IEEE 802.15.4a from 2007 extends the specification of physical layer of this standard. It specifies the PHY layer using **ultra-wide band (UWB)** technology. It brings advantages in spectral efficiency, ability of high data rate with low-power or device location ability, however it is not suitable for unsafe zones due to the high peak energy of pulses.

The PHY and MAC layer of standard 802.15.4 gives a basis for the whole range of standardized protocols like ZigBee, WirelessHART, ISA100.11a, 6LoWPAN. They differs in their implementations of higher layers of OSI model.

Zigbee was introduced by Zigbee Alliance in 2004. Its objective is to be a cost-effective, low-rate, low-power communication technology for embedded devices in home automation, monitoring and control applications.

It is built on IEEE 802.15.4 and further specifies network (NWK) and application layer (APL). The MAC layer employs only CSMA/CA technique and supports mesh routing. At startup it scans the channel with the least amount of interference, but there is no frequency hopping. Although there is support for security, including authentication, integrity and encryption in NWK and APL layer, the security is not mandatory.

Paper [26] compares the performance of ZigBee Pro and ISA100.11a in an aerospace environment. They concluded that even simpler and cheaper protocol ZigBee Pro performs relatively well under moderate levels of interference, but to ensure reliable data delivery under heavier interference, the more complex and costly protocol ISA100.11a is needed.

Due to the lack of industrial-grade robustness, coexistence and security, ZigBee is not considered suitable for the use in most industrial applications. This was a motivation for developing new standards such as WirelessHART or ISA 100.11a which meets requirements of communication in harsh industrial process environments. [27]

WirelessHART gives the Highway Addressable Remote Transducer Protocol (HART) a wireless ability. It was introduced in 2007 by a large group of industrial companies and since 2010 it was approved as the international standard IEC 62591.

The protocol creates a mesh topology, which aims to be time synchronized, self-organizing and self-healing.

WirelessHART is fully compliant with IEEE 802.15.4 MAC layer and further employs a Time Division Multiple Access (TDMA). All devices are time synchronized and communicates in pre-scheduled fixed length time-slots. To avoid interference it uses a a Frequency Hopping Spread Spectrum (FHSS) across 16 channels.

Security is mandatory and there is a number of security measures implemented in all layers of protocol stack.

WirelesHART uses strictly the HART protocol and defines only one NWK layer, which simplifies implementation for the end user. WirelesHART has wider variety of available devices manufactured by multiple manufacturers with guaranteed interoperability between them.

Since 2009 **ISA100.11a**, is the standard approved by the International Society for Automation (ISA). It is intended to provide reliable and secure wireless connection for non-critical monitoring and control industrial applications.

MAC layer combines TDMA and CSMA/CA and similar to WirelessHART it uses also frequency hopping. TDMA provides deterministic features of communicatiom, while CSMA/CA improves ability of retransmission on failure links.

ISA100.11a provides a number of options that must be specified by the end user, e.g. it defines several network layer header formats supporting IPv6. Due to the higher variability in ISA100.11a devices, the end user should be more careful to ensure interoperability between devices from different suppliers.

WirelessHART and ISA100.11a are two independent and competing standards specifically designed for industrial wireless applications. Papers [28] and [29] bring the detailed description and comparison of both industrial WSN protocols. Table 2.4 further summarizes main features of ZigBee, WirelessHART and ISA100.11a

Table 2.4: Comparison of IEEE 802.15.4 based industrial standardized protocols

Attribute	ZigBee	WirelessHART	ISA100.11a
PHY layer	IEEE 802.15.4 (2.4 GHz)	IEEE 802.15.4 (2.4 GHz)	IEEE 802.15.4 (2.4 GHz)
MAC layer	IEEE 802.15.4 (CSMA/CA)	IEEE 802.15.4 + TDMA + frequency hopping	IEEE 802.15.4 + TDMA + CSMA/CA + frequency hopping
Max. datarate	250 kb/s	250 kb/s	250 kb/s
Robustness in industrial environments	low	high	high
Security	middle	high	high
Implementation	easy	moderate	challenging
Compatibility	–	HART	Implements tunneling to encapsulate foreign protocols
Number of suppliers supplying products	high	high	low

Custom-defined standards and protocols supported by local groups

Besides the internationally standardized protocols based on IEEE 802.15.4 described in 2.2.3, there is a number of protocols developed by individual researchers or industrial alliances.

For example **6LoWPAN** integrates the IPv6 network protocol to low-power WPANs. It is based on PHY and MAC layer of IEEE 802.15.4 and it defines encapsulation and header compression mechanism, that allow IPv6 packets to be sent and received. The contribution of this standard is mainly in the interoperability with other IP devices. Paper [30] brings an experimental evaluation of 6LoWPAN in industrial applications.

Another protocols suitable for industrial applications are **ANT, Dash7, EnOcean, Z-Wave** and many others, which are discussed in [31].

2.3 WSN applications

This section brings state-of-the art studies where WSNs are used in industrial environment for machinery condition monitoring. The industrial environment is a term used to describe working conditions that may be outside of optimal, which is very similar to the environment of aircraft applications.

The industrial environment is covered by the WSN studies for condition monitoring especially due to the fact that these general areas are easily accessible than the aviation tightened by strict rules.

Nevertheless the studies for general industrial WSNs application are pioneers giving good basis even for WSNs applied within aircraft industry.

The WSN applications may be found for example for water supply pump diagnosis [32], for condition monitoring in end-milling [33], for monitoring oil and gas processes in refineries [34], for monitoring manufacturing processes like metal cutting by a CNC machine [35], for pipelines infrastructure monitoring [36] and for many others.

Recent papers deal with the application of IWSN in performance monitoring of electrical machines [37], induction motors [38], manufacturing machines [39], pump and pipeline diagnosis [40], auxiliaries in power plants [41], Smart Grids [42] and structural health monitoring [43].

Reference [41] introduces a concept for vibration data acquisition and an online decision IWSN system for monitoring rotating auxiliaries at power plants. This paper introduces a data-level fusion algorithm for similarity judgment of time series, and a task-level fusion algorithm which decides about the sending data strategy – merge and piggyback similar data and thus reduce the total bandwidth and power needs.

Papers [44] and [45] propose IWSN-based induction motor condition monitoring and fault diagnosis. The system monitors the motor stator current and the vibrational signature from two nodes. Data acquisition, feature extraction and classification by the Neural Network Classifier are implemented in the node. Decision level fusion using Dempster-Shafer theory is further executed in the center. The training phase is performed off-line in supervised manner. To complete the training, the four states have to be introduced manually to an experimental machine by adding load and imbalance. The system brings a power-effective approach with distributed signal processing. However, supervised classifier training limits the use to cases where it is possible to introduce a faulty behavior to a tested machine.

2.4 Summary

This chapter brings an overview of the key areas needed for this work. First it describes current trends in condition monitoring of aircraft. Then it brings extensive overview of Wireless Sensor Networks and at last it summarizes the WSN state-of-the art studies and applications in industrial environment.

Chapter 3

Aims of the doctoral thesis

Based on the research of state-of-the art applications summarized in Chapter 2 there emerges a need to increase aircraft components on-board condition monitoring capabilities. Special focus is held on powerplant, systems and components which manifest its health in vibrational behavior. To broaden current condition monitoring systems there are requirements especially to:

- Provide on-line, long-term on-board vibration monitoring capabilities, while energy resource is constrained.
- Perform fault detection while no example of faulty behavior is available.
- Deploy system with no wires, which allows easy installation at multiple spots and gives possibility of retrofit into legacy engines.

The aims of the presented thesis are therefore as follows:

Introduce a novel approach based on WSN

The main objective of this thesis is **to introduce a novel approach** for aircraft components condition monitoring employing computationally intensive methods of vibrational signal processing and methods of condition monitoring in computationally weak wireless sensor network.

Propose a fault detection method based on novelty detection

Propose a fault detection method detecting a novelty in vibrational signatures relative to the baseline signature obtained during condition monitoring system installation phase. The character of process of vibration sensing requires the fault **detection method to be created individually** for each sensor node.

Propose a novel distributed WSN framework

Propose a novel WSN framework allowing distributed signal processing to maximize the immediate fault detection capability while provide long-term monitoring. And further enable the system to react to ambiguous machine states by temporarily changing the diagnostic focus.

Evaluate on jet engine use case

Design an experiment to evaluate proposed methods and framework on aircraft jet engine use case.

Chapter 4

WSN based vibration condition monitoring system

4.1 Analysis of requirements

Based on the high level requirements summarized in Chapter 3 the requirements for vibration condition monitoring system based on WSN are further identified and elaborated in this Chapter.

The requirement for no wires, easy to install and possibility of retrofit was fulfilled by decision to employ Wireless Senor Network. This decision introduces strong boundaries to proposed system architecture by limits and capabilities of WSNs described in Section 2.2.

The requirement for easy installation is also translated as very limited possibility for further physical action to the system as its service, repair or battery replacement.

The need for long-term monitoring capability, while the energy resources are constrained with the requirement for vibrational signal sampling rate at least at units of kHz brings a challenge because these requirements are contradictory.

The approach to fulfill above mentioned requirements is based on:

- Distributed decentralized in-node signal (pre)processing, which allows to radically reduce the communication which is the major energy consumer (see Section 4.2.1). Streaming a raw vibrational signal is not only extremely energy consuming but also it is not possible to stream simultaneously large amount of data from multiple sensor nodes due to low data throughput of low-power wireless communication protocols.

- Event-based programming scheme and software architecture, where the nodes spend the majority of time in extremely low-power sleep and wake up just for short periods where the all necessary work is done.

- Low-power hardware design and potentially energy harvesting. However this is over the scope of this thesis.

The requirement for on-line capability which means the ability of the individual node to be available for real-time interaction upon the system's request is achieved within the system's topology, software and communication protocol scheme.

The last but very crucial is the requirement for the fault detection where no example of a fault is available. At the moment when the monitoring system is installed on a machine, there is no information how the potential fault could manifest. At the same time a monitored object is considered as in health state. For that reason a special kind of pattern recognition design for novelty detection called also one-class classification or outlier detection was employed, see details in Section 5. One-class fault detection together with learning and testing phases of operation described in 4.4 allow to run the fault detection algorithm directly in-node.

To further maximize detection capability and long-term monitoring the monitoring system has ability of:

- Adaptive behavior
- Reconfigurability, see Section 4.3

Table 4.1: Analysis of requirements and proposed solutions

Requirement	Consequence (Restriction)	Proposed solution
No wires, easy to install.	Employ WSN.	Propose WSN based monitoring system (4.2).
Long-term monitoring Limited energy resource	Low-energy design	Distributed decentralized in-node signal (pre)processing (4.2.1). Event-based software design (2.2.2).
Vibration monitoring: Sampling rate at least at units of kHz	Not possible to stream raw samples.	Distributed decentralized in-node signal (pre)processing (4.2.1).
On-line capability	Real-time interaction upon request	System architecture and topology (4.2).
Fault detection where, no example of fault is available.	Track a change of behavior rather than an absolute measurement.	Develop a Fault Detection algorithm (5).
Maximize a detection capability and long-term monitoring	Requirements are contrary to each other	System's adaptability and reconfigurability (4.3).

Section 4.1 provides analysis of the key requirements emerging from the state of the art in machine condition monitoring. In addition to that it also provides the proposal of solutions of individual requirements which are summarized in the Table 4.1 and thus serves as the guide for further thesis subsections.

4.2 System architecture

Following sections are describing proposed framework which is introducing a novel approach for condition monitoring. This approach enables to employ computationally intensive methods of vibrational signal processing and methods of condition monitoring in computationally weak nodes of WSN.

The architecture of proposed framework creates a core part of author's contribution. The framework is innovative primarily by its approach of distributed signal processing, and by its adaptivity and reconfigurability allowing to change the diagnostic focus of proposed monitoring system.

As stated before the proposed solution is based on Wireless Sensor Networks. Features determining the machine behavior are processed directly in node. Then a machine condition is classified locally in each sensor node. Further the trade off between the precise analysis of machine condition and nodes' energy consumption is achieved thanks to Adaptive behavior. The roles are divided between sensor nodes and central node. Methods of fault detection are carried out in sensor nodes while methods of fault localization and trend watching are processed in the central node.

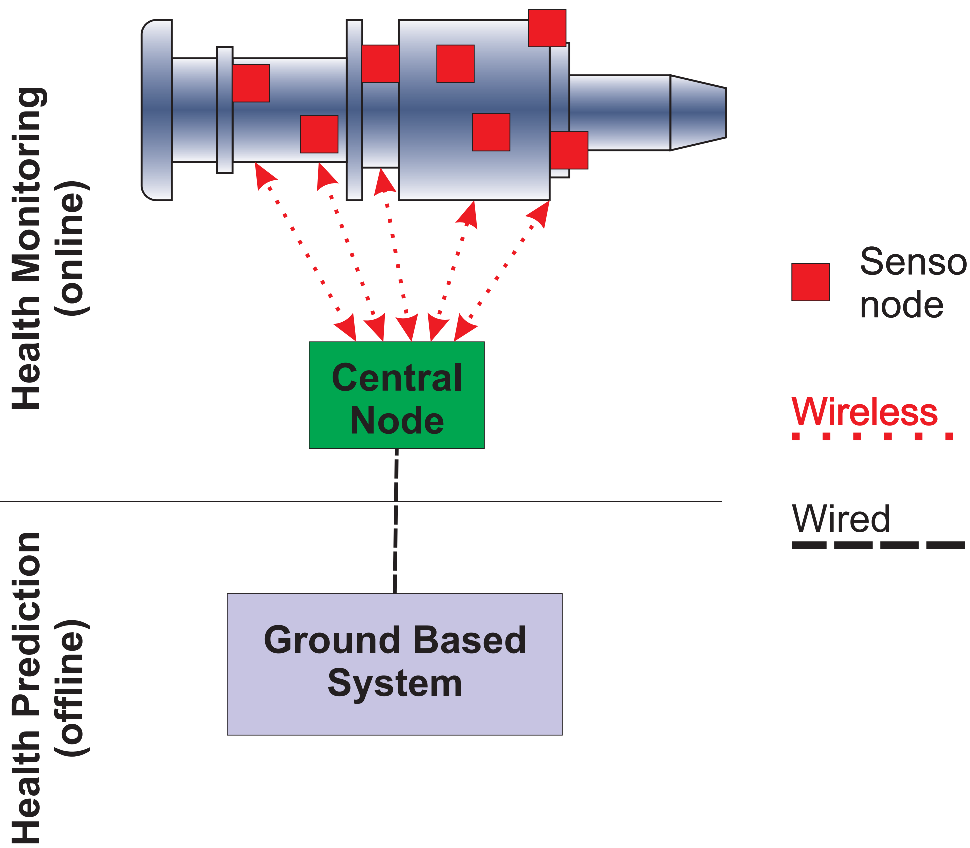

Figure 4.1: Top Level Architecture

This work is aimed above all to the methods of fault detection which are processed directly in the sensor nodes. However the proposed framework allows further fault diagnosis and prognosis in the central node or ground bases system as depicted in the Table 4.2 and Fig. 4.1

Table 4.2: Top level monitoring system architecture

Task	Location	Manner
Fault Detection	In-node	On-line, immediatelly
Fault Diagnostics	Central node	On-line
Prediction	Central node/Ground based system	Off-line

4.2.1 In-node signal processing

Due to the restricted energy resources and limited data throughput it is not possible to create a centralized system where the role of the nodes is just to sample vibrations and stream them to the central node for further processing (see case A in Fig. 4.2). However as the node has this ability, it can be useful for detailed analysis performed off-line in the central node, but due to the network data throughput it is not possible to transmit the data from all the nodes simultaneously.

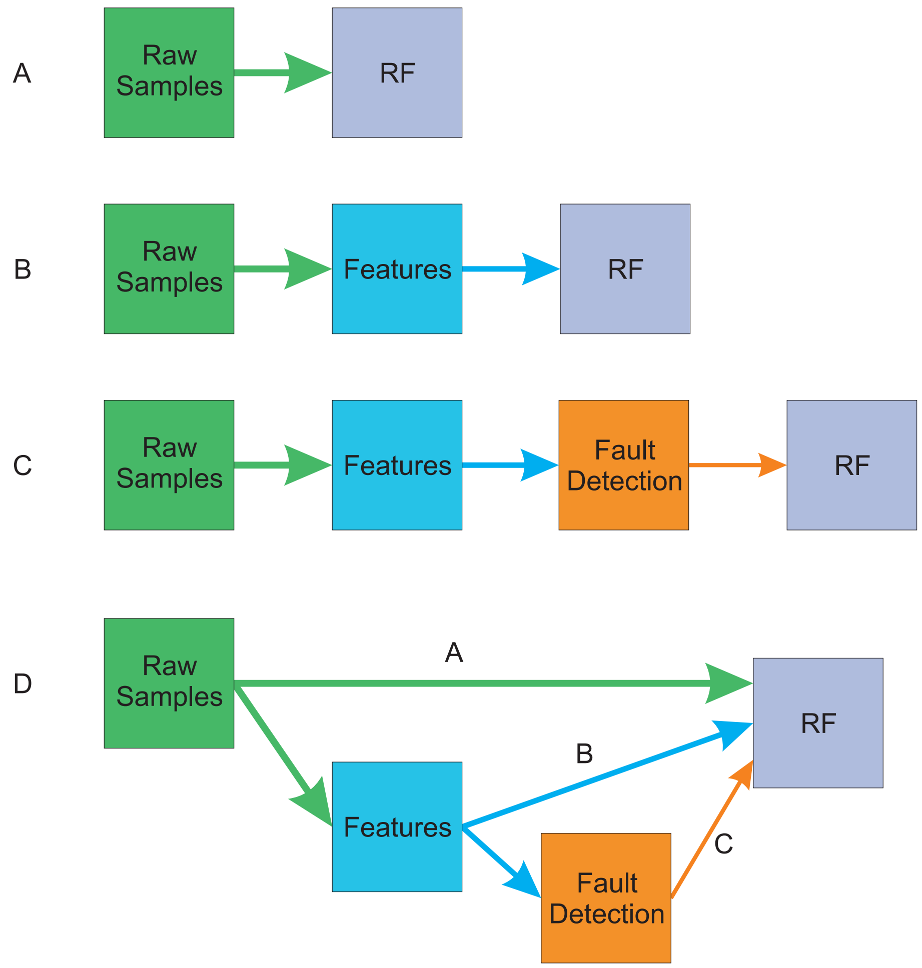

Figure 4.2: In-node methods of a sensor node (RF – Radio Frequency transmission)

The only way how to enable vibrational signal handling in a WSN is that the sampled signals is processed directly in node. The in-node signal processing results in a set of features which contain the compressed information (see case B in Fig. 4.2). The number of features is usually of much lower order than the number of samples of signal. Obtained features may be send to the central node or they serve as the input for fault detection.

In-node fault detection

Transmitting the messages is the most energy consuming operation of a node. To further decrease the energy consumption and thus enable long-term monitoring capability, it is profitable to perform fault detection directly in node. A method of pattern recognition is applied in node to obtain a decision about a fault. This step further compress the useful information and helps to save the energy when it is transmitting.

4.2.2 Sensor node vs. central node

A set of **sensor nodes** is deployed on a tested machine when a machine is manufactured and assembled or when it is after overhaul. Nodes are put on selected positions as close to the source of expected vibrations as possible, e.g. on bearing house.

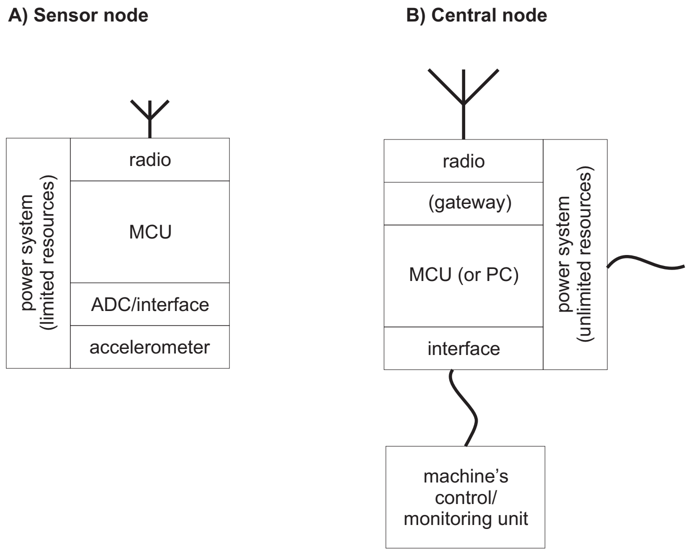

Figure 4.3: Sensor node vs. central node

As depicted in Fig. 4.3 a), the only source of information for a sensor node are accelerometer (one per sensor node) and radio module. An accelerometer connected to Analog to Digital Converter (ADC) provides an instantaneous high-sampled signal. Thanks to radio module a node can receive additional information helpful for fault detection (e.g. shaft rotational speed). But due to the limited time-synchronization this information is not instantaneous neither synchronized. Thus, it is preferable to implement certain level of autonomy within sensor node. It routinely performs its default regime unless it is commanded by the central node to adapt its behavior or perform non-default operation.

Table 4.3 shows a relative comparison of power consumption of sensor node's operational regimes. However a specific WSN system has its own power values, this relative comparison shows that generally the communication (both transmit and receive) has much higher power demands than in-node signal processing. On the other hand the sleep regime has usually multiple-order lower power demands than active regimes (2.2.1). Control of switching and duration of sensor node operational regimes has a crucial effect on its energy consumption, while the source of energy is restricted.

Table 4.3: Relative comparison of power consumption of sensor node individual operations, estimate based on [46] and on [47]

Sensor node operational regime	Relative power consumption (-)
Transmit (Tx)	100
Receive (Rx)	80
Process signal	25
Idle	10
Sleep	0.1

A central node depicted in Fig. 4.3 b), contrary to a sensor node has relatively unlimited power source, which enables a high computational power. Central node has its radio module connected directly or using gateway, when there is different hardware and software architecture of sensor and central node. It is also connected to the machine's control (and monitoring unit, when available), which provides instant information about machine operational regime and another process values. Thus a central node records information received from all sensor nodes and merges them with machine operational regime. A central node commands sensor nodes connected in network to adapt their default behavior.

4.3 Adaptivity and Reconfigurability

Within the proposed architecture there is introduced Adaptivity and Reconfigurability of the monitoring system. Adaptivity enables the system to react to ambiguous machine states by temporarily changing the sensor nodes methods of signal analysis.

While Reconfigurability allows to completely reconfigure the whole monitoring system functionality without a need of physical approach to the monitoring system.

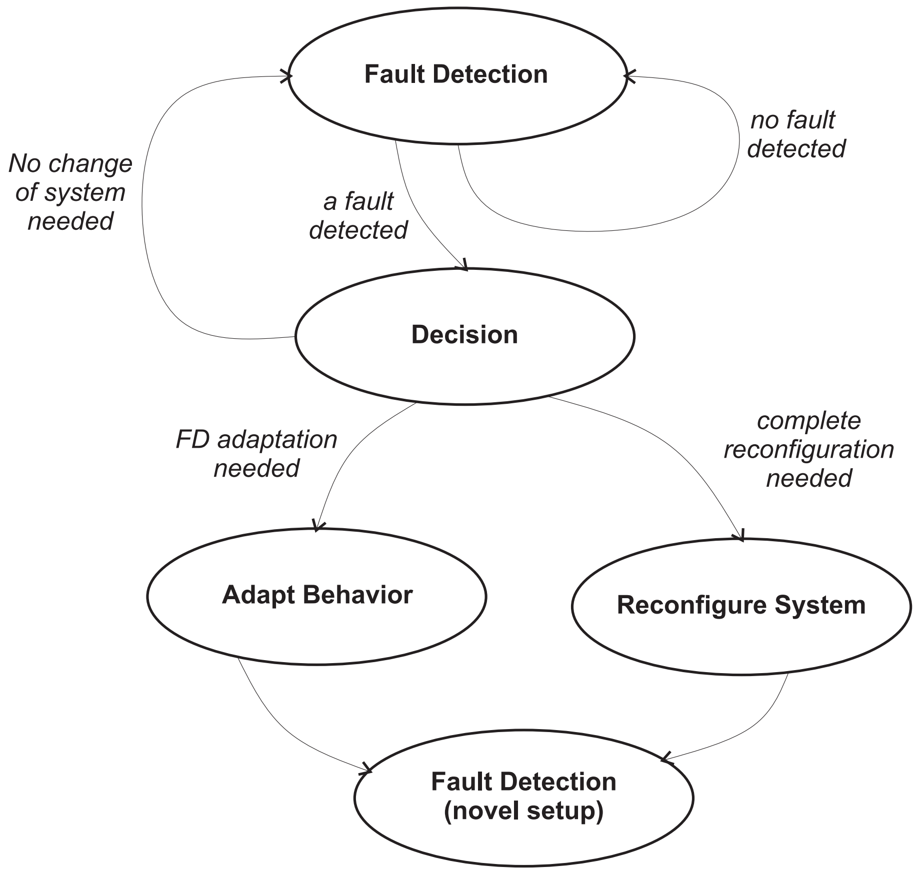

Figure 4.4: Adaptivity and Reconfigurability

Fig. 4.4 describes the Adaptivity and Reconfigurability approach. Default regime of monitoring system is in-node Fault Detection (top of the Fig. 4.4) where each sensor node of the system autonomously carries out in-node fault detection. When a fault is detected a **Decision** (4.3.1) has to be undertaken. Action performed based on the decision carries out additional in-node methods which improve the fault detection result and enable central-node fault diagnosis. When there is, based on the Decision, performed Adaptation or Reconfiguration the monitoring system creates new default Fault Detection state, in Fig. 4.4 depicted as Fault Detection (novel setup).

The Decision is made on the level of the central node, which can combine fault detection results of a group of sensor nodes and additional information such as machine's operational regime.

The decision result for this case is to **Adapt Behavior** (4.3.2) of an individual node's or subset of nodes' behavior or to **Reconfigure System** (4.3.3), i.e. to reconfigure the whole condition monitoring system.

The key criterion for decision is to achieve the trade-off between the quality of information needed for fault detection and diagnosis on the one hand and the energy consumption of the operation which is taken based on that decision on the other hand. The reason of adaptivity and reconfigurability is to maximize the detection and diagnosis capability and immediate precise information while ensure long-term monitoring.

4.3.1 Decision

The inputs for decision taken on the central-node's level are:

- Sensor nodes' fault detection results (could be one result from one sensor node or subset of sensor nodes),
- machine's operational regime and its control system signals,
- additional machine monitoring system results,
- state of machine monitoring system, e.g. energy remaining/available.

Criteria of decision

Action taken based on the decision is evaluated using following criteria:

1. **Scenario rating:** Scenario is a combination of operations, methods and their timing which will be performed based on the decision. There are several possible scenarios, each of them is judged by the scenario rating. Two main elements of scenario rating are the **contribution** of a scenario to fault detection and diagnosis and its **energy demand**.
   The scenario's contribution is performance of methods and their application. Generally for features obtained from a raw signal their contribution is higher as they react to a machine's faulty behavior with higher sensitivity and vice versa. For a fault detection method its performance is evaluated based on its ability to detect and/or localize a fault and on its fault positive and fault negative errors.
   The scenario's energy demand is determined based on the sum of operation power demand, as depicted in Table 4.3, and its duration.
   $$Energy\ Demand_{scenario} = \sum_{i} (Operation\ Power_i \cdot Operation\ Duration_i) \tag{4.1}$$
   where i stands for individual operation within the scenario.

2. **Requirement for additional method:** Based on the results of running fault detection and/or diagnosis process there could arise a requirement for a custom operation of a sensor node to carry out. A custom operation performs a different method than in-node fault detection which improves fault detection or allows fault localization.

3. **Aggregated energy consumption:** The energy drawn from individual sensor nodes by executing operations is controlled by aggregated energy consumption mechanism. This mechanism limits the extensive usage of high-energy-demanding operations by keeping the average energy consumption under the limit given for the time window. The length of time window and the average consumption limit is given mainly by: the type and amount of system's energy resources, planned monitoring period and system's criticality.

4.3.2 Adaptivity

To adapt the monitoring system's behavior means that the default in-node monitoring method is adapted to improve machine condition monitoring result. Adaptation enables to perform not only improved fault detection but also fault diagnosis and localization. Regime of adapted behavior is connected with higher energy demands and so it takes only the necessary time and the monitoring system returns back to default regime as soon as possible. The trade-off between energy consumption and improved performance of fault detection and diagnosis is driven by Aggregated Energy Consumption (4.3.1).

The fundamental elements of adaptive behavior are:

- **Adjust measurement update period T_M:** Measurement update period is the time between individual sensor's collection of readings, see Fig. 4.5–1. The shorter is this period the more continuous is the coverage of machine's behavior but the higher is the energy consumption. A default measurement update period is given by the monitoring system type and requirements for continuous monitoring, criticality and monitoring system expected life. When an in-node fault is detected this period could be shortened to confirm machine's faulty behavior or to refuse it as false alarm.

- **Adjust transmit update period T_T:** Transmit update period is the time between sending off the in-node fault detection result to the central node, see Fig. 4.5–1. If $T_T = T_M$ the fault detection result is sent after each individual in-node fault detection procedure. The process of transmitting is very energetically consuming, thus it is profitable to set default transmit update period longer than measurement update period: $T_T = T_M \cdot k$, where $k = 2, 3, 4 \ldots$ When an in-node fault is detected this period could inform the central-node immediately.

- **Send features to central node:** The process of fault detection is as default executed in-node. The most significant limit of in-node fault detection is that node's input is isolated only to its own sensor readings (see 4.2.2). On the request from the central node, the sensor node sends the in-node-computed features, see Fig. 4.5–2. Then the central node executes fault detection and diagnosis with all inputs available as described in 4.3.1. A sensor node can also compute and send different set of features, if its computing and memory capabilities enable it.

- **Send raw samples to central node:** The sensor node is also able to send its acquired raw time samples on the request of central node, see Fig. 4.5–3. Although this operation is from the above mentioned the most energetic

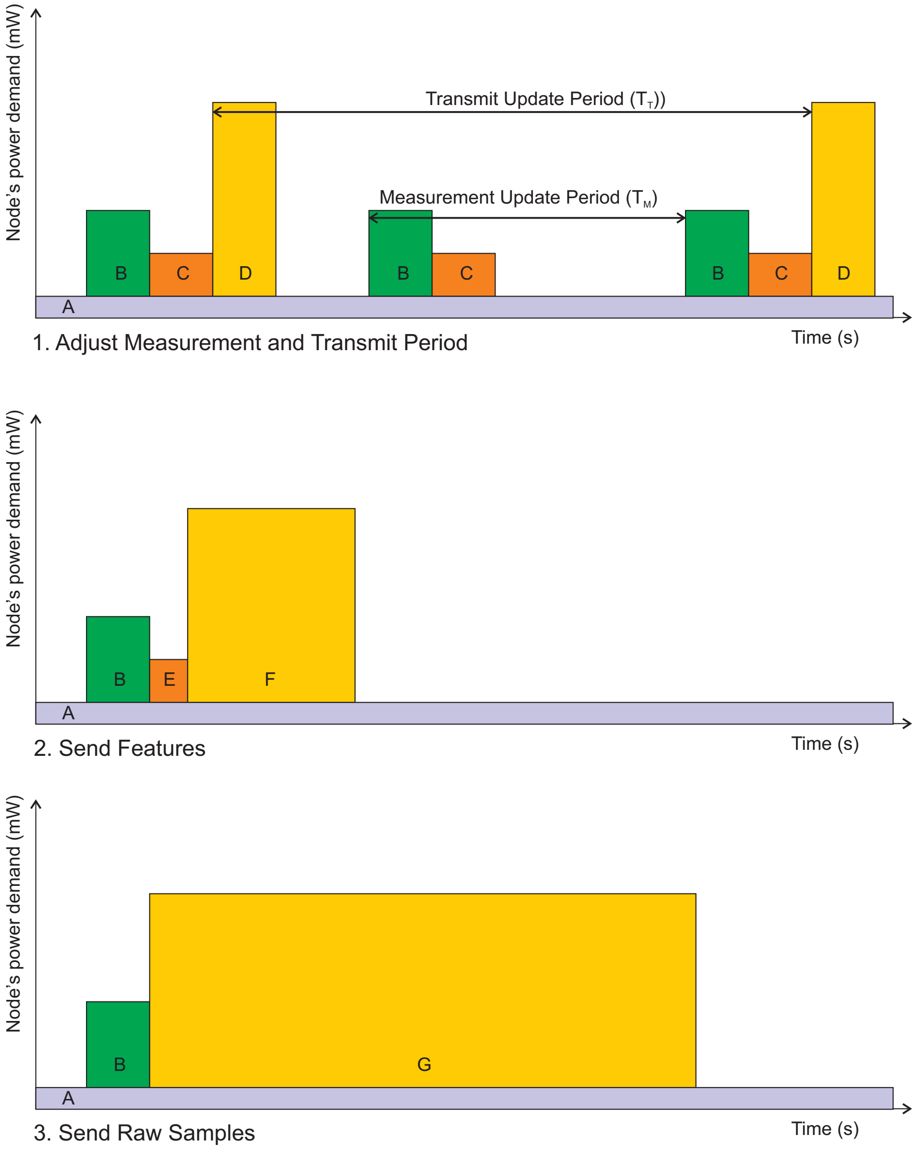

Figure 4.5: Power demands of node's operation during adaptive behavior: A – Idle/sleep, B – Sample, C – Compute features and perform Fault Detection, E – Comte features, F – Send features to central node, G – Send raw samples to central node

demanding, it considerably improves fault detection and fault diagnosis especially. Main benefit of this step is the whole signal processing in the central node with high computational capabilities and connection to whole sensor network and additional information sources. Based on the results the monitoring system is further adapted or reconfigured. Raw data samples and signal analysis results are also stored for further off-line analysis.

4.3.3 Reconfigurability

To reconfigure the system means that the node's behavior is completely changed, new methods of features computation and fault detection are implemented. Reconfiguration ability is achieved thanks to Over-the-Air-Programming (OTAP), when new program is created and built in the central node and then is over the air uploaded and introduced to the target sensor node. When the reconfiguration is evoked steps of Learning phase, described in section 4.4.2 are activated. Reconfiguration may be applied to all sensor nodes within the sensor network or only to a subset o them. Besides, the main portion of this operation is performed by the central node it is very energetic demanding for the sensor nodes: a sensor node must receive the whole program.

Although reconfiguration is the most energy demanding method it is very beneficial in log-term horizon in the case when the machine condition default regime stops performing well. This situation happens when machine's healthy behavior changes during its operation, see the instance labeled by red circle in Fig. 4.6. The default in-node fault detection methods would cause only false alarm and the system would became infective for the long-term perspective.

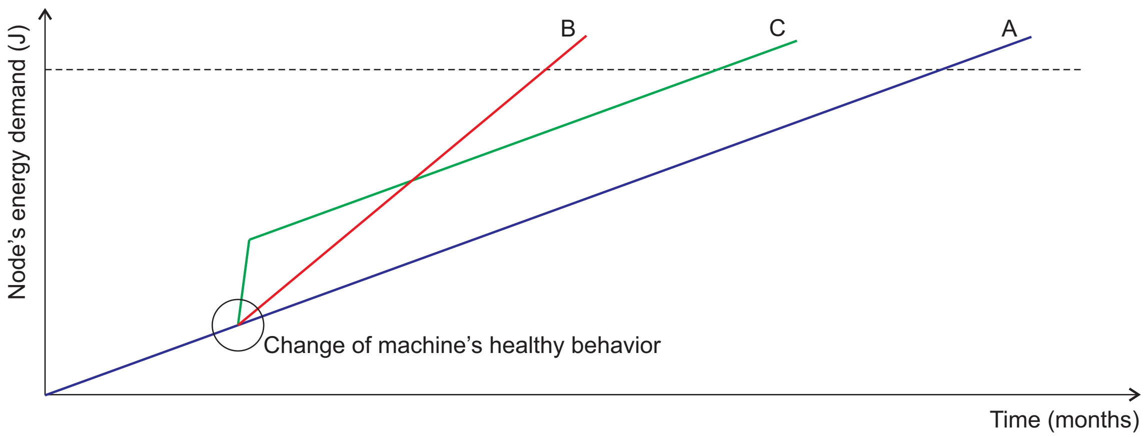

Figure 4.6: A node's energy consumption: A – monitored machine does not change its health behavior, B – monitored machine change its health behavior and node's default in-node fault detection creates false alarms, C – Monitoring system reconfiguration is applied

4.4 Phases of operation

The key idea of monitoring system's phases of operation is to split data acquisition, fault detection method developing and condition monitoring between different units of the system. The computationally intensive operations are held in power unrestricted central node while the performing regular fault detection is held in computationally week and energy restricted sensor nodes.

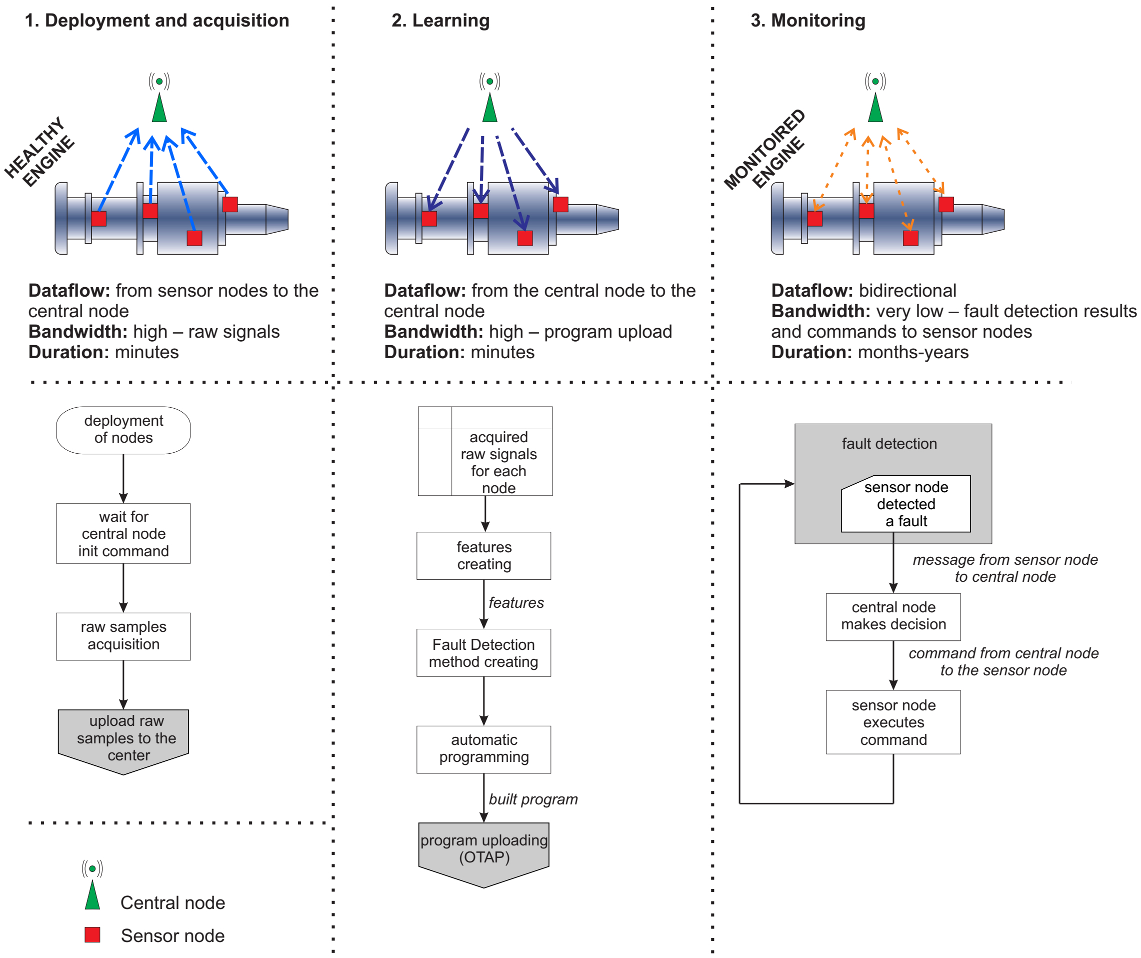

Figure 4.7: Phases of operation

The monitoring system is autonomously learns the engine's healthy behavior and then monitors the change of its behavior which indicates an incipient failure.

The usage of the proposed system is split up into three phases: 1. Deployment and acquisition, 2. Learning, and 3. Monitoring (see Fig. 4.7). At the beginning of the operation the sensor nodes are deployed on the monitored engine and the fault detection algorithm is created, after that the monitoring phase is initiated. The first and second phase are relatively short compared to the third phase, when the systems performs health monitoring. The key features of this process are:

- At the beginning of the process the healthy behavior of an engine is recorded as a baseline.
- An individual fault detection algorithm is developed for each specific sensing point.

- Development of fault detection algorithms takes place in the central node.

4.4.1 Phase 1: Deployment and acquisition

In the first phase the baseline of vibrational behavior of the healthy engine is recorded. The state of health is confirmed by a conventional testing procedure, in case of aircraft engine after an overhaul. The sensor nodes are deployed on a monitored engine where they will remain during all three phases, i.e. they will remain at their location for the whole monitoring system's life cycle. When a machine runs at default operational regime (e.g. nominal rotational speed), the central node initiates process of sampling by the command addressed to all sensor nodes. The sensor nodes are equipped with a program optimized for raw data sampling and transmitting them to the central node. The acquired signal is split into a sequence of individual messages, which are stepwise delivered into the central unit. Integrity of transmitted data has the highest importance contrary to on-line demand. The data messages cannot be transmitted from all nodes simultaneously because of limited data throughput. This phase finishes when complete data sequences have been delivered from all sensor nodes into the central node. The central node has the full authority over the sensor nodes, while sensor nodes perform just raw data sampling and sending. See the process flow of data acquisition in Fig. 4.8.

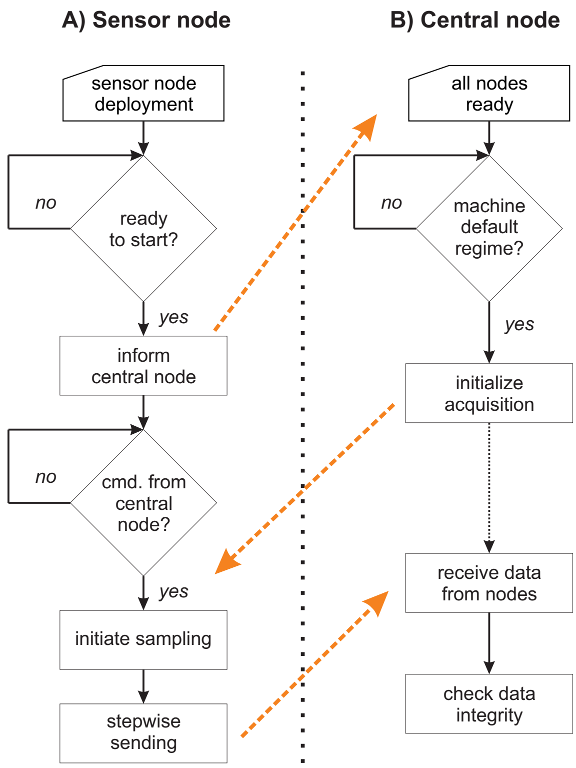

Figure 4.8: Deployment and acquisition

4.4.2 Phase 2: Learning and programming

When the acquisition phase is finished the raw signal samples from all sensor nodes are assembled in the central node, where process of features and fault detection method creating and automatic programming is held. While the the complexity

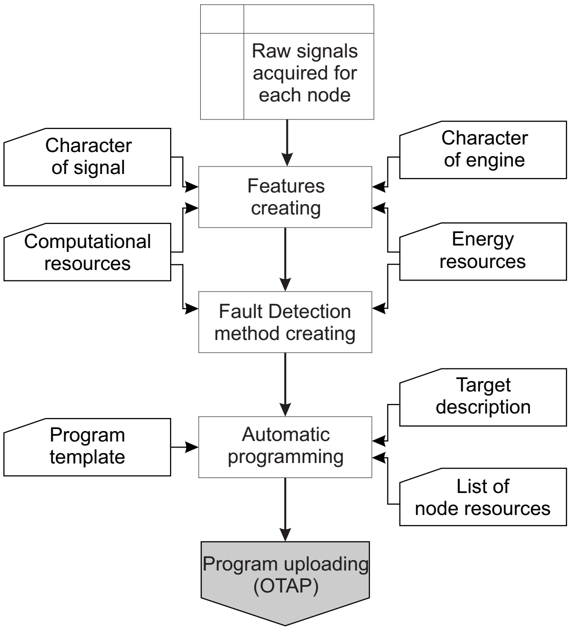

Figure 4.9: Learning phase: Develop and upload a sensor-node specific Fault Detection algorithm

of above mentioned creating process may be relatively high thanks to high computational and energy resources of the central node, the execution of the proposed methods must be feasible in target low-power sensor nodes.

Features creating

The first step within the learning phase is to create and implement appropriate features which enable the condition of the machine to be detected.

The input blocks Character of engine and Character of signal (Fig. 4.9) give the information about components of engine, its default regime of operation and about position of a sensor node. This provides requirements for time-frequency resolution of a given feature (minimal sampling frequency, minimal number of samples) and also requirements for a feature to be able to detect a typical failures of engine's components. The better the feature describes an engine's behavior the better contribution it has. See details in Section 6.

The inputs Computational and Energy resources (Fig. 4.9) give the information about the computational and energy restrictions for features computing. Computational restrictions are given mainly by the type of hardware of sensor node, see details in Section 2.2.1. The energy restrictions are given by the energy consumption of sensor node and its power regimes (4.3), by the node's energy resources available and also by the planned sensor node operating time.

The energy demand of in-node computed feature is given particularly by the time needed for computation and power consumption during this operation, see 4.2:

$$Energy\ Demand_{feature} = (Operation\ Power \cdot Operation\ Duration) \tag{4.2}$$

The output of the feature creation is a set of features, which are in compliance with following criteria:

- sensitivity on behavior of given engine's components,

- to be feasible on given low-power sensor node,
- the energy demand for their execution is known.

Fault detection method creating

The second step within the Learning phase is proposal of fault detection method (the middle part of Fig. 4.9), specifically one-class classifier. Details of fault detection methods and classifiers are described in Section 5. The fault detection method is also restricted by computational and energy resources of sensor node. Moreover, the computational demand of a classifier increases exponentially with the number of input features (5.2), so the number of features selected for fault detection must be relatively low.

The performance of a classifier method is evaluated based on its ability to detect a fault and on its fault positive and fault negative errors.

The result of learning step is a classifier model trained for each individual sensor node.

Automatic programming

Using a Program template and based on Target description (microcontroller architecture), and List of node's resources (the available resources as volatile and non-volatile memory and peripherals) the created features and classifier are implemented to a ready-to-upload, compiled program specific for each sensor node.

This program is uploaded via Over-The-Air-Programming (OTAP) to the sensor node. The OTAP technique enables the node to be booted to a specific regime, when it expects to download the new program segmented in a sequence of messages. When the new program is downloaded, the sensor node reboots and starts executing the new program.

4.4.3 Phase 3: Monitoring and Fault Detection

In the third phase the proposed system performs its main objective: to monitor engine's condition and perform fault detection.

The sensor nodes perform its default operation: in-node fault detection according to the **default scenario**. This scenario sets default transmit update period ($T_{Tdefault}$) and default measurement update period ($T_{Mdefault}$) which may be individual for each sensor node. See the update periods in Fig. 4.5. Parameters of default scenario are based on:

- Need for continuous monitoring: $T_{Mdefault}$ is shorter with demand on continuous monitoring (risk of missing short-time change of behavior), while it is longer where several snapshots are sufficient for engine's condition monitoring.
- Criticality and rapid reaction of change: Both $T_{Tdefault}$ and $T_{Mdefault}$ are small when there is demand of very fast reaction and adaptivity of monitoring system and vice versa.
- Type of sensor node energy resources and planned monitoring phase duration: $T_{Mdefault}$ and $T_{Tdefault}$ play the key role in expected monitoring system life

cycle. For given expected life cycle the trade-off between sensor node energy resource and $T_{Mdefault}$ and $T_{Tdefault}$ must be found.

Fig. 4.10 shows simplified flow diagram for the monitoring phase and depicts different roles of a sensor node and the central node. Sensor nodes perform in-node fault detection and are capable to adapt its behavior, perform another monitoring methods or completely reconfigure as described in Section 4.3. Central node controls the monitoring and fault detection process and when necessary commands the sensor nodes to adapt (4.3.2) or reconfigure (4.3.3). Based on the character of engine's monitoring system the central node also initiates the monitoring phase.

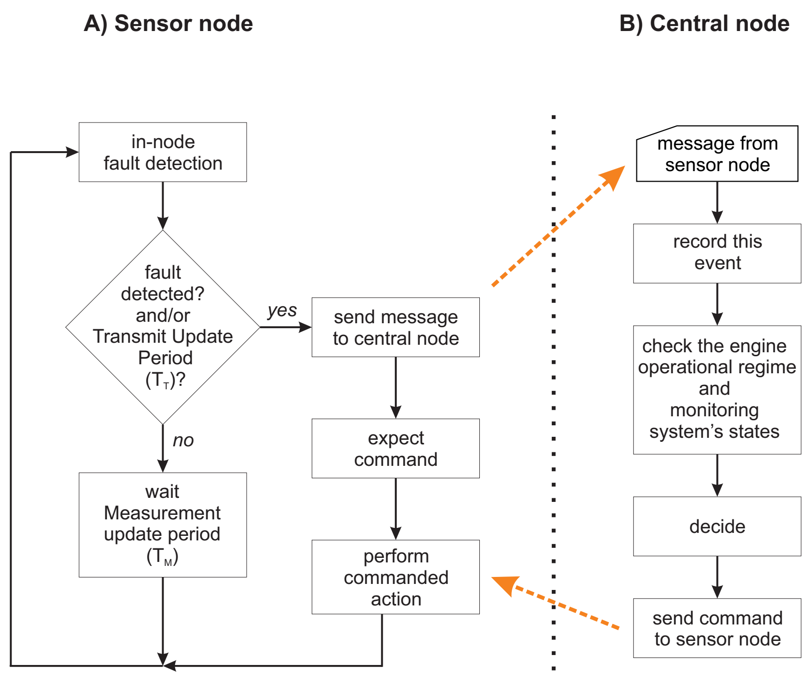

Figure 4.10: Phase: Monitoring

4.5 Summary

This chapter introduces the novel and original approach for WSN based condition monitoring. This approach enables to employ computationally intensive methods of vibrational signal processing and methods of condition monitoring in computationally weak nodes of WSN.

Thanks to the approach key attributes Adaptivity, Reconfigurability and Three phases of operation enables the immediate fault detection capability while providing long-term monitoring.

Chapter 5

Novelty fault detection

From the requirements defined in Section 3 there arises a need of development of monitoring system which is able to detect a fault of a machine, while there is no example of a fault when a system is created. In other words there arises the question: **"How to detect a fault of monitored machine, when there is no example of fault available?"**

Usually it is easy to obtain a measurement of monitored machine at normal working conditions, which represents healthy state of a machine. In opposite it is very difficult, expensive and often impossible to obtain a measurement which represents a state of machine's fault. Such a measurement would require introduction of some kind of destruction to a machine.

Automatized fault detection system involves data-driven approach of machine learning: pattern recognition (or classification). Conventional methods of pattern recognition require a data set which comprises all classes (situations) of behavior, while each class is sufficiently represented. These conventional methods can not be used in the case, when only one class is described well and there is no (or very little) information about the other classes. However, solution of this specific problem can be accomplished by method called one-class classification.

5.1 Features

Feature x is obtained (computed) from the sampled signal. Set of features represent a pattern (i.e. a object, a state). An object is a single point in the d-dimensional space. The most important task is to find such features which create compact and separable regions in the feature space for each class to be classified.

For multi-class recognition task a process of discovering of appropriate features is provided by feature selection or extraction. The goal of feature selection is to find the most suitable features from a large set of all possible features (from measurement vector). It means that the most important features are used, whereas the non-important are discarded. In opposite, the feature extraction approach transforms the space of original measured vector into space with reduced dimension. It leads to features which contain condensed information from the signal. Quantitative criteria like inter/intra class distance, Chernoff–Bhattacharyya distance, probabilistic distance measures, probabilistic dependence serve to determine quality of selected or extracted feature. [48]

For one-class recognition task it is not possible to use above mentioned methods of features extraction and selection, because it is not known, how the features would react in the case of machine's fault.

Thus the features must be obtained based on the knowledge of the machine substance, its components and expected behavior. This is rather model-based approach, than data-driven selection/extraction. Furthermore, in the case that features are computed at low-power microcontrollers, the computation demands has to be taken into account.

5.2 Classification

A classifier is a function which outputs a class label for each input object. An input object is specified by the feature vector $\mathbf{x} = (x_1, \cdots x_d)$. Dimension of feature space is determined by number of features d.

The recognition system is commonly operated in the two phases: training (learning) and classification (testing), see in the Figure. 5.1. In the training phase there has to be found preprocessing techniques, feature selection/extraction methods and has to be set up a function of a classifier. In the testing phase, the system independently process features $\mathbf{x}$ and classifies the input pattern to one of the categories $\omega_1, \omega_2, \cdots, \omega_c$.

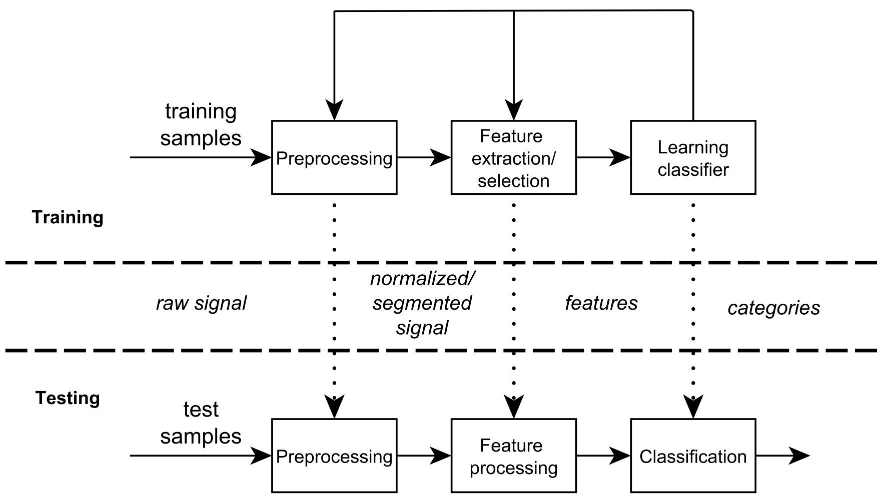

Figure 5.1: Training and testing phase of pattern recognition [49]

Classifier learning can be supervised or unsupervised. For the supervised learning the training samples are labeled by a class. While in the unsupervised learning the classes must be found in the pattern without this information. [49]

In the process of classification, one of the four situations depicted in Table 5.1 can happen. Two kind of errors may be distinguished:

- error of first kind ϵ_I – for MCM it means a false alarm,
- error of second kind ϵ_{II} – fault which was not detected.

Optimizing a classifier to maximize its performance on the training set does not always lead to the desired performance on the testing set, because the training set may not describe the real complete situation but just its subset. Therefore there

Table 5.1: Result of an object classification

	Object from target class	Object from outlier class
Classified as a target object	true positive	false positive (error ϵ_{II})
Classified as an outlier object	false negative (error ϵ_I)	true negative

is requirement for a good generalization of a classifier, i.e. an ability to perform well on the testing set. Main factors which may harm a good generalization and performance of a classifier are overfitting, too many unknown parameters of classifier model (function) and curse of dimensionality.

The overfitting or also overtraining problem arises when a classifier is trained too precisely on the training samples with a low complexity. This is strengthened by a large number of features per object (pattern) and also by too complex (many unknown parameters) function of classifier.

The term curse of dimensionality describes the fact that a volume of the feature space increases exponentially with a number of features. That implies that some compromise between number of features, complexity of classifier and amount of training data must be found. [50]

5.3 One-class classification

One-class classification is a special case of classification problem, where only information about one class – *target* class is available. This means that in the training process there are only the samples which represent the object of target class. The boundary of a classifier has to be estimated just based on this target class. The task is to find a such boundary that accepts as much target class while minimizes the chance of accepting the *outlier* objects. [50]

Several methods are known [50] to construct a model $f(\mathbf{x}; \mathbf{w})$ (where $\mathbf{x}$ is a feature vector and $\mathbf{w}$ is a vector o weights) of one-class classifier. Three main approaches may be distinguished: probability density estimation, boundary methods or reconstruction methods.

In **density methods** a probability density of training samples is estimated and a threshold is determined. Usually a probability density function is assumed as Gaussian or Poisson. Function of Gaussian model $f(\mathbf{x}; \mu, \mathbf{C})$ is shown in 5.1

$$f(\mathbf{x}; \mu, \mathbf{C}) = \frac{1}{(2\pi)^{\frac{3}{2}}|\mathbf{C}|^{\frac{1}{2}}} \cdot \mathrm{e}^{-\frac{1}{2}(\mathbf{x}-\mu)^{\mathrm{T}} \cdot \mathbf{C}^{-1} \cdot (\mathbf{x}-\mu)} \tag{5.1}$$

where $\mathbf{x}$ is the classified object in the feature domain and weights $\mathbf{w}$ are μ – vector of mean values, and $\mathbf{C}$ – covariance matrix. Weights μ, $\mathbf{C}$ and threshold are found in the process of training, while the main computational effort is the inversion of the covariance matrix $\mathbf{C}$.

Tested object is accepted or rejected based on the threshold Θ in 7.2:

$$h(f(\mathbf{x})) = \begin{cases} \text{target} & \text{if} \quad f(\mathbf{x}) \leq \Theta \\ \text{outlier} & \text{if} \quad f(\mathbf{x}) > \Theta \end{cases} \tag{5.2}$$

The threshold Θ has to be found based on the trade off between the fraction of the target class accepted and the fraction of outliers rejected [50]. It could be determined analytically thanks to the assumption that the objects have Gaussian distribution. Because the assumption of Gaussian distribution does not always hold ideally, it is better to determine the threshold Θ empirically in the process of training based on the fraction of target class to be rejected chosen by the user. However, this setup results in certain false negative error ϵ_I (i.e. false alarm), it prevents classifier from overtraining and helps to provide better generalization.

The computation complexity is d^3 and to store the classifier model takes $d+d^2+1$ constants $(\mu + \mathbf{C} + \Theta)$.

When the amount of training samples is big enough more flexible density model can be used e.g. mixture of Gaussians or Parzen density estimation. Mixtuture of Gaussians solves better a situation when the data in dataset are not unimodal and convex. But the user has to provide another parameter – number of Gaussians. That also increases computational demands of testing phase. While the computational cost for training a Parzen density estimator is almost zero, the testing is very expensive because all training objects have to be stored. [50].

The greate advantage of density methods is in relatively easy computation and direct connection with the substance of solved problem.

Boundary methods come out of the Vapnik statement, that when just a limited amount of data is available, one should avoid solving a more general problem as an intermediate step to solve the original problem [51]. That means that it is not necessary to estimate the complete data density (as in density methods), when only the boundary around the objects is necessary. Boundary methods are based on the distances thus they are very sensitive to the scaling of features and the output can not be interpreted as a probability. Into these methods may be included: k-centers, Nearest Neighbor method (NN-d) or Support Vector Data Description (SVDD) [48, 50].

Reconstruction methods make assumptions about the clustering characteristics of the samples or their distribution in subspaces. A set of prototypes or subspaces is defined and the a reconstruction error is minimized [50]. Here may be placed methods such as k-means, learning vector quantization (LQM), principal components analysis (PCA) or auto-encoders and diabolo networks based on neural networks. Reconstruction methods are usually computationally demanding (both in training and testing phase) and the direct connection with a problem substance is mainly lost.

5.4 Summary

This chapter summarizes the state-of-the art methods suitable for in-node fault detection. The method of one-class classification was identified as favorable approach novelty in-node fault detection. Especially the density method is very convenient for the given application thanks to its low computational complexity of the testing phase.

Chapter 6

Vibration-based monitoring of turbine engines

The typical application for vibrational monitoring of aircraft systems, components and powerplant is the gas turbine engine. Generally the engine is one of the most important and expensive elements of an aircraft. Further the gas turbine engines are found in a large number of platforms and in large number of applications. Typically as turbofan engines of large passenger aircrafts and business jets, turbo-shaft application in helicopters or in Auxiliary Power Units or as turboprop engines in propeller driven aircrafts.

This chapter introduces methods of vibration-based signal analysis with focus on methods suitable for distributes signal processing implemented in Wireless Sensor Networks.

For the vibration-based monitoring of the aircraft turbine engines it is important to understand an engine operation and to analyze its components significant for vibrodiagnosis. Further by the application of signal analysis methods are extracted features which describe engine's behavior and allow to detect and/or diagnose a faulty behavior.

6.1 Aircraft Gas Turbine Engines

The general classification of aircraft gas turbines and their components significant for vibration monitoring is described in Table 6.1:

Additionally based on an engine design there may by multiple shafts, multiple compressor and turbine stages each rotating at different velocity and additional systems such as an accessory gearbox.

From the vibrodiagnostic perspective gas turbine engines are rotating machines which suffers mainly in:

- Imbalance, misalignment, looseness (improper fit) of shafts, compressors, turbines.
- Bearings faults.
- Gearboxes faults.

Table 6.1: Aircraft turbines and their components significant for vibration monitoring

Turbine Engine	Significant components for vibration monitoring
Turbojet	Compressor, shaft, turbine
Turbo-shaft	Compressor, shaft, turbine, gear box (gear box connected to free turbine stage)
Turboprop	Compressor, shaft, turbine, gear box, propeller (gearbox which powers a propeller is connected directly to the shaft)
Turbofan	Fan, compressor, shaft, turbine.

6.2 Time domain methods

Some features can be calculated from the raw time signal obtained from sensor. But usually data must be conditioned or preprocessed using technique as amplification, mean value removal, time synchronous averaging or filtering [52]. Conditioning and preprocessing improves the possibility to extract useful information from signal, but in some cases these techniques are demanding or unfeasible. Methods of signal preprocessing are depicted in Figure 6.1. Following features are divided into groups according to signal preprocessing technique.

6.2.1 Raw time signal

The simplest approach in time domain is to measure root-mean-square (6.1). This feature measures the power content in vibration signature. It describes the overall noise level, but can not localize the faulty element. It can very effective detect major out-of-balance of a rotary machine, but can not detect appreciable changes in early stages of gear and bearing damages [52].

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N} {x_i}^2} \tag{6.1}$$

where N is the number of samples in measured discretized signal, and x_i is the value of the i-th sample of the signal.

More robust is to use the ratio of the peak level of the measured signal to the RMS level, which is called crest factor (6.2). For normal operations, crest factor ranges between values 2 to 6. Higher values usually point to machinery problems [52].

$$crest = \frac{peak}{RMS} \tag{6.2}$$

It is usually also quite useful to use statistical moments, which describe probability density curve and its deviation from Gaussian distribution. The first and second moments are known as a mean value (6.3) and variance (6.4). The third normalized moment is known as the coefficient of skewness (6.5), fourth normalized moment is called kurtosis (6.6). For an undamaged bearing, the kurtosis value is close to 3. A greater value indicates impending failure [53].

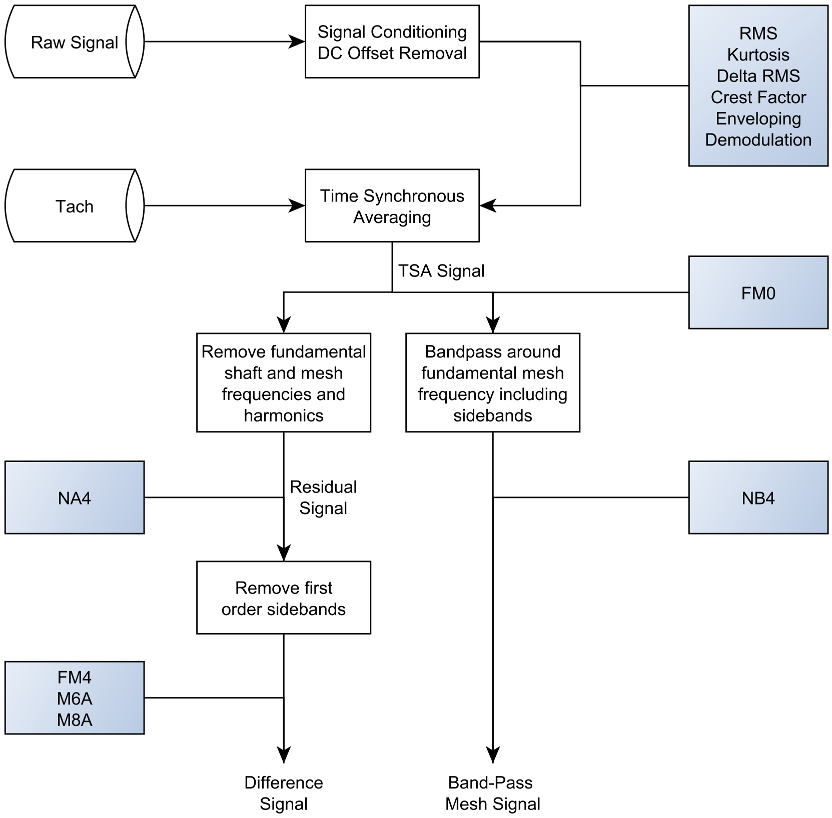

Figure 6.1: Time signal – extraction methods [52]

$$\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{6.3}$$

$$\sigma^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2 \tag{6.4}$$

$$skewness = \frac{\sum_{i=1}^{N}(x_i - \overline{x})^3}{N\sigma^3} \tag{6.5}$$

$$kurtosis = \frac{\sum_{i=1}^{N}(x_i - \overline{x})^4}{N\sigma^4} \tag{6.6}$$

6.2.2 Time synchronous averaged signal – TSA

Time synchronous averaging is a technique, which extracts repetitive signals from additive noise. Signal is divided into segments with the same length based on the synchronizing signal (tacho pulse). Then the signals segments are averaged, thus the random noise is suppressed.

$FM0$ (6.7) detects major tooth faults, which usually manifest by an increase of the peak-to-peak level of the signal, but do not change the meshing frequency.

$$FM0 = \frac{PPA}{\sum_{i=1}^{n} A(f_i)} \tag{6.7}$$

where PPA is peak-to-peak value of TSA and $\sum_{i=1}^{n} A(f_i)$ is the sum of amplitudes correspond to the mesh frequency and its harmonics. $FM0$ is a good indicator of a major, but not minor tooth damage. [52]

6.2.3 Residual signal – RES

Residual signal is obtained from TSA signal by removing fundamental shaft frequency and its harmonics.

$NA4$ (6.8) was developed to detect the onset of damage and to continue to react to this damage as it spreads and increases in the amplitude [54].

$$NA4 = \frac{N \sum_{i=1}^{n} (r_i - \overline{r})^4}{\left[\frac{1}{m} \sum_{j=1}^{m} \left(\sum_{i=1}^{n} (r_j - \overline{r}_j)^2\right)\right]^2} \tag{6.8}$$

where r is residual signal, $\overline{r}$ is mean of the residual signal, N is the total number of data points in time record, m is the current time record in the run ensemble [52]

6.2.4 Difference signal – DIF

Difference signal is created by removing the sidebands of the primary meshing frequencies of the RES signal [52].

$FM4$ feature detects changes in the vibration pattern resulting from damage on a limited number of gear teeth [54]

$$FM4 = \frac{N \sum_{i=1}^{n} \left(d_i - \overline{d}\right)^4}{\left[\sum_{i=1}^{n} \left(d_i - \overline{d}\right)^2\right]^2} \tag{6.9}$$

where d is the difference signal and $\overline{d}$ is its mean value.

Features $M6A$(6.10) and $M8A$ (6.11) were proposed to detect surface damage on machinery components. It is expected, that both features are more sensitive to the peaks in DIF signal in compare to $FM4$.

$$M6A = \frac{N^2 \sum_{i=1}^{n} \left(d_i - \overline{d}\right)^6}{\left[\sum_{i=1}^{n} \left(d_i - \overline{d}\right)^2\right]^3} \tag{6.10}$$

$$M8A = \frac{N^3 \sum_{i=1}^{n} \left(d_i - \overline{d}\right)^8}{\left[\sum_{i=1}^{n} \left(d_i - \overline{d}\right)^2\right]^4} \tag{6.11}$$

6.2.5 Band-pass mesh signal – BFM

Band-pass mesh signal is the TSA signal filtered around the primary gear mesh frequency, including as many sidebands as possible [52]

$NB4$ (6.12) uses the envelope of a BFM signal. This feature points to the damaged teethes, which express as a transient load fluctuations.

$$NB4 = \frac{N \sum_{i=1}^{n} \left(E_i - \overline{E}\right)^4}{\left[\frac{1}{m} \sum_{j=1}^{m} \left(\sum_{i=1}^{n} \left(E_i j - \overline{E}_j\right)^2\right)\right]^2} \tag{6.12}$$

where E is the envelope of the BFM signal and $\overline{E}$ is its mean value.

6.3 Frequency domain methods

Spectral analysis of the vibrational signal of a rotating machine is a very useful and widely used tool for machine condition monitoring. From the essence of a machine, which rotates periodically implies that the faults are also periodically repeated in the signal and thus can be remarkable in the frequency domain. Fourier transform decomposes the time signal into the set of harmonics components:

$$\mathrm{F}(f) = \int_{-\propto}^{\propto} \mathrm{x}(t)\mathrm{e}^{-2\mathrm{j}\pi f t}\mathrm{d}t \tag{6.13}$$

where $\mathrm{F}(f)$ is a complex spectrum of a time signal $\mathrm{x}(t)$.

Often is used a Power Spectral Density (PSD) for random continuous signals. PSD is a distribution function, which describes the distribution of the power of signal in frequency domain.

To compute Discrete Fourirer Transform (DFT) of N points takes N^2 arithmetical operations. Using algorithm of Fast Fourier Transform reduces the number of arithmetical operations to $\mathrm{N}\log_2(\mathrm{N})$.

Techniques of MCM from spectrum are based on examination of present frequency components, their magnitudes and phase shifts. Each element of a machine has its specific contribution to the spectrum. If the parameters of elements are known it is possible to compute the frequencies were the faults are expected. Then the levels, ratios and other features of this frequencies can be tracked. In global the frequency domain methods are good in localizing the defects, but can not react to the earliest stages of faults.

Bellow are described methods of vibordiagnostic signal analysis based on spectrum evaluation.

6.3.1 Misalignment

Misalignment occurs when shafts, couplings and bearings are not aligned along their centerline. Angular misalignment is caused by improper joint of two shafts in such way, that there is induced a banding force on the shaft. Parallel misalignment occurs when two shaft are aligned parallelly but shifted from each other. Missalignment can mainly cause a bearing fault due to introducing a higher load than a bearing is designed.

Angular misalignment can be recognized by axial vibration at running frequency (1st order frequency) and also by 180 ° phase shift across the coupling of machine. Parallel misalignment produces radial vibration at twice the running speed frequency (2nd order frequency). Phase shift from 0 ° to 180 ° can be observed from horizontal to vertical position of sensor. Usually angular and parallel misalignment are combined. Severe misalignment is typical by emerging of higher orders in spectrum (3rd to 10th order).

6.3.2 Imbalance

Imbalance occurs when the center of rotation is not the same as the shaft's mas centerline. Static imbalance (can be observed at rest) only one force (weight) is involved, while in couple imbalance there are two forces involved. Couple imbalance is observed only when a shaft is rotating. Usually the static and couple imbalance are combined into adaptive imbalance. Imbalance can, similarly as misalignment, contribute mainly to bearing fault.

Imbalance is usually emerged by increase of radial vibrations at 1st order frequency, while very low axial vibrations. There is phase shift of 90 ° between horizontal and vertical radial vibrations and and there is usually no phase shift across the machine.

6.3.3 Looseness

Looseness is caused by improper fit between component parts: typically distortion of the base frame, loose bolts, cracks in bearing pedestal, bearing liner loose in its cap or loose shaft.

It is typical by long string of harmonics (up to 10th harmonics) and also by subharmonics multiples of 0.5. Occurrence of looseness is unstable and can significantly vary from one measurement to the other.

6.3.4 Defects of rolling bearings

Bearing failure is one of the most frequent reasons of machine breakdown. A bearing may fail for many reasons e.g. ineffective/contaminated lubrication, heavier loading than anticipated or improper installation, however a bearing defect is usually caused by some other machinery problem.

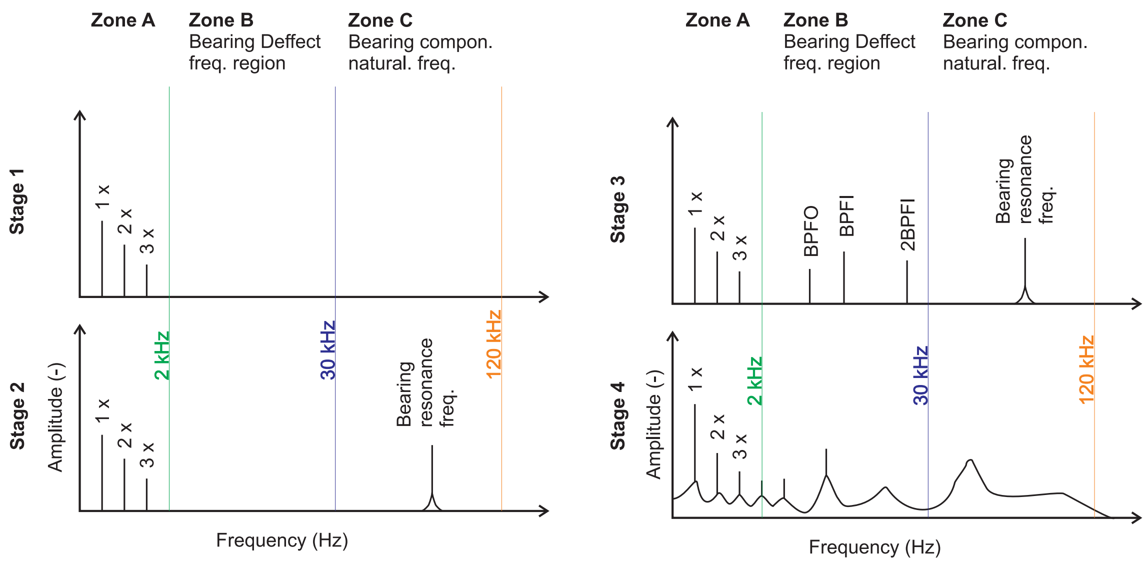

Figure 6.2: Bearing faults

Bearing faults can be classified into four stages based on the degree of severity as depicted in the Fig. 6.2.

In **Stage 1** the very beginning of bearing fault is emerging in ultrasonic frequencies over 200 kHz up to units of MHz. **Stage 2** is typical by natural bearing frequencies, i.e. resonance of bearing material, ranging from 30 kHz to 120 kHz. Due to relatively high frequencies detected in stage 1 and 2, it is not possible to use conventional accelerometers.

In **Stage 3** there are observed discrete bearing frequencies generated by balls passing over a defect. Discrete frequency is determined by the shaft speed ω_s, bearing geometry and defect location. The characteristic defect frequency for the radial ball bearing can me estimated as (6.14) for outer race defect, (6.15) for inner race defect, (6.16) for cage defect and (6.17) for ball spinning.

$$\omega_{or} = Z\omega_c = \frac{Z\omega_s}{2}\left(1 - \frac{d}{D}\cos\alpha\right) \tag{6.14}$$

$$\omega_{ir} = Z(\omega_s - \omega_c) = \frac{Z\omega_s}{2}\left(1 + \frac{d}{D}\cos\alpha\right) \tag{6.15}$$

$$\omega_c = \frac{\omega_s}{2}\left(1 + \frac{d}{D}\cos\alpha\right) \tag{6.16}$$

$$\omega_b = \frac{D\omega_s}{2d}\left(1 - \frac{d^2}{D^2}\cos^2\alpha\right) \tag{6.17}$$

where ω_s is the shaft rotation frequency, d is the diameter of ball, D is the pitch diameter, Z is the number of balls, and α is the contact angle [53, 55].

In this stage also a bearing wear is visible and bearings have approximately 1–5 % of remaining life. Bearing in this stage should be replaced immediately.

In the final **Stage 4** discrete bearing frequencies and natural frequencies disappear, while 1st order grows and the broadband noise floor increases. Severity of bearing damage is very high, there is less than 1 % of remaining life.

More information may be found in [1] or [55].

6.3.5 Gears defects

Thanks to spectral analysis it is possible to observe the energy changes in gear rotating and meshing frequency (6.18) and their harmonics.

$$f_m = f_s N \tag{6.18}$$

where f_m is the meshing frequency, f_s is the gear speed and N is the number of teeth. From the parameters of the gear elements it is possible to find and localize defects at certain frequencies. Gearbox spectrum is characteristic by amplitude modulation caused by damaged teeth. This can be evaluated directly from the spectra or advanced techniques as enveloping (see 6.4.2) and cepstral analysis (see 6.4.3) may be used.

6.4 Other methods

6.4.1 Order analysis

Order analysis is a special case of spectrum analysis, when sampling is synchronized with shaft rotational speed. The key benefit is when the rotational speed of a machine is not constant (e.g. run-up, run-down). First order at the x-axis relates to the shaft rotational speed and thanks to synchronizing it is related to the certain position of the shaft. Then the higher orders can be related for example to the number of the teeth (e.g. meshing frequency) in a gearbox.

6.4.2 HFRT – Envelope

High-Frequency Resonance Technique (HFRT) is used to monitor the high-frequency response of the mechanical system periodic impacts. Each time the rolling element of a bearing hits a part with a defect or the tooth of gearbox meets a faulty tooth, the impulse and its response in the material is produced. This response is at much higher frequency than defect frequencies and its energy is extended at relatively narrow frequency band. The method analyzes frequencies of the structural response. First, the vibrational signal is band-pass filtered around the excited structure resonance frequency. Second, the envelope is created using rectifier, peak-holder and smoother. Then the spectra of envelope is estimated and compared with the defect frequencies. [52, 55]

For simple machines good results may be obtained, but for more complex machines it is difficult to select appropriate structural frequencies. [55]. Enveloping also proved in early detection of bearing faults [52].

6.4.3 Cepstral analysis

Cepstrum, which can be described as the spectrum of the logarithmic power spectrum (6.19), is useful in detecting spectrum periodicity, e.g. components which are uniformly spaced in spectrum. This method can for example detect periodic impulses caused by faulty bearing (6.4.2) and so the wear or damage of tooth in gearbox. Cepstrum analysis can in this case prove the amplitude modulation, because the sidebands are uniformly disposed around the gear mesh frequency and thus point to fault, which has not to be so obvious from spectrum. [52, 55]

$$x_c = \mathrm{F}\left[\ln \left|\mathrm{X}(\omega)\right|\right] \tag{6.19}$$

where $\mathrm{X}(\omega)$ is spectrum of $\mathrm{x}(t)$.

When the $\mathrm{X}(\omega)$ is complex, the cepstrum is known as the *complex cepstrum* although its module is even and its phase is odd, the complex cepstrum is real-valued. When the power spectrum is used to replace spectrum $\mathrm{X}(\omega)$, the resulting spectrum is known as the *power spectrum* or *real cepstrum* and is thus a scaled version of complex spectrum where the phase of the spectrum has been set to zero. [55]

6.4.4 Time–frequency domain methods

While methods based on spectral analysis are good in localizing defects compare to time domain methods, they are less effective with nonstacionary phenomena associated with localized faults. Time–frequency analyses such as Short-Time Fourier Transform (STFT, spectrogram), Wavelet Transform (WT), the Winger-Ville distributions (WVD) or the Choi-Williams distributions (CWD) are designed to give the best in both time and frequency resolution. They describe the energy distribution over frequencies changes over the time. [55]

But main drawback of time–frequency methods are high computational cost and non-trivial evaluation of features.

6.5 Summary

See the Table 6.2 which summarizes the methods for vibration based monitoring and also brings their assessment of applicability for in given situations.

Table 6.2: Vibration-based monitoring methods summary

Feature	Comp. & Memory Demands	Comment
Time-Domain – from raw signal (RMS, crest, amplitude, kurtosis, skewness)	*	Describe well the overall machine behavior
Time-Domain – from conditioned signal (FM0, NA4, Fm4, M6A, M8A)	**	Synchronized averaging requires a tacho pulse, machine rotation freq. has to be known; filtering increases comp. demands;–
FFT – only amplitude spectrum	***	Describes machines overall behavior more detailed than time-domain methods
FFT – amplitude and phase	***	Requirement of very precise time synchronization of nodes to perform phase analysis of signals measured at different spots
Order analysis	***	Requires instant information about the shaft position or at least precise instant rotational frequency
Enveloping (HFRT)	****	Useful for analyzing high-frequency signals (material structural response)
Cepstral analysis	****	To evaluate a feature requires precise information of rot. frequency
Time-Frequency methods	*****	Besides computational cost non-trivial evaluation of features; useful for non stable regimes of engine

This chapter analyzed the methods for vibrational monitoring of aircraft turbine engines with focus on methods suitable for distributed signal processing implemented in Wireless Sensor Networks.

Chapter 7

Experimental evaluation of the proposed approach

Evaluation of proposed novelty fault detection method and of proposed WSN framework for vibrational condition monitoring was carried out by means of designed experiment using a small turbojet engine test case.

Proposed framework and methods were implemented in WSN platform Crossbow IRIS. As a test case the small aircraft turbojet engine was selected representing the typical use case.

7.1 Design of experiment

The experiment is designed to prove key abilities of proposed system. Its main aim was to evaluate proposed fault detection method and proposed WSN based vibrational monitoring system framework, especially its features of adaptivity and reconfigurability.

The main features of this experiment are:

- Once the suitable spots for vibrational monitoring were selected, the sensor were deployed and remained for the whole duration of the experiment.
- The engine was operated in different regimes of operation and different states of its health.
- Conventional data acquisition: To achieve the repeatability of the experiments the engine's vibration were at first phase recorded using conventional equipment, see the Fig. 7.1,a).
- Simulated engine run: evaluation of proposed framework was then performed by reproducing recorded signal, see the Fig. 7.1, b).

In the phase of simulated engine run the digitalized stored signal was reproduced using Digital to Analog Converter and after signal conditioning (low-pas filter, gain, offset) was brought directly to a sensor's node Analog to Digital Converter, see the Fig. 7.1, b). In this phase the laptop has two roles: First, to reproduce engine's vibrations of given operational phase and state of condition; second, to act as central node. This scheme allows to focus directly on the evaluation of in-node algorithms and its behavior of the engine monitoring system.

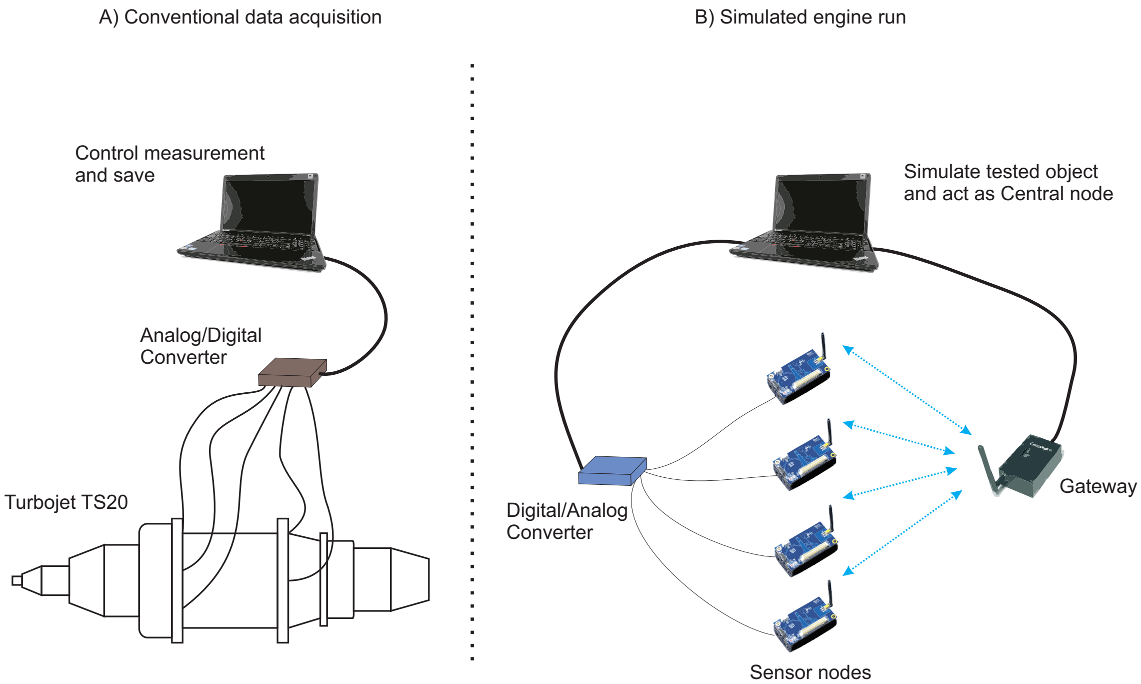

Figure 7.1: Setup of experiment

7.2 Object of experimental evaluation

7.2.1 Turbojet TS20

TS20 is a small experimental single shaft turbojet engine originated from TS-20B turboshaft which served for starting a large turbojet engine AL 7F-1. It is composed of a single 20-blades radial compressor and single stage 26-blades turbine. It is equipped by two bearings: the first ball bearing at front part and the second roller bearing at rear part. See detailed description and mechanical parameters in [56]. The tested turbojet engine TS20 is located at University of Defence, Brno, Czech Republic and operated by Department of Air Force and Aircraft Technology.

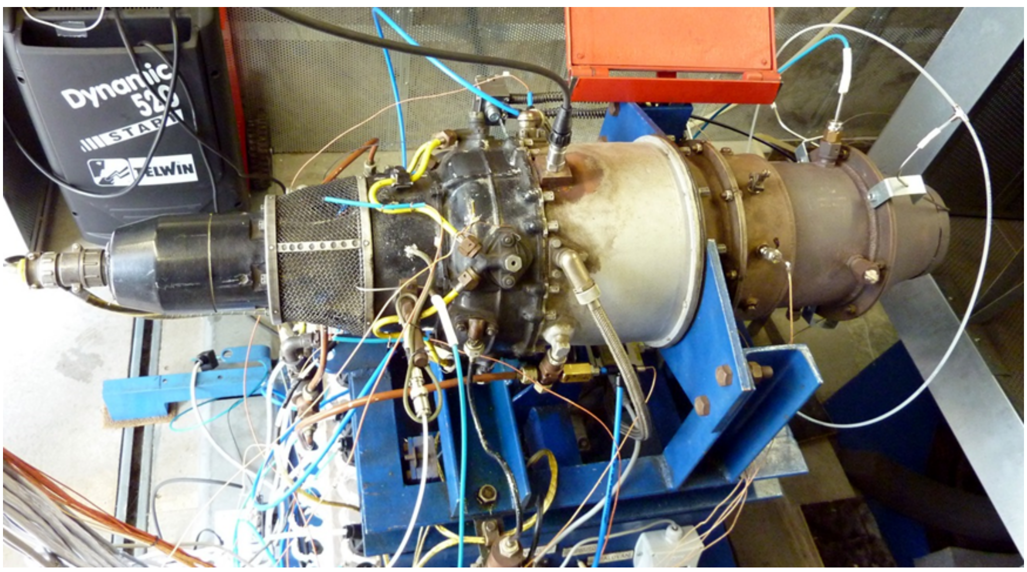

Figure 7.2: Turbojet engine TS20 at University of Defence Brno

Engine is started using an electrical starter. When the revolutions builds up to its nominal value it operates at constant speed for up to 50 seconds. The variation of constant speed is not bigger than 5%. See summary of operational regimes in Tab. 7.1.

Table 7.1: TS20 operational regimes (RPM – Revolutions Per Minute)

Regime	Rotational speed	Description
Cold rotation	7.5k RPM	Stable speed rotating the engine by electrical starter with no ignition
Reduced speed	44k RPM	Nominal reduced stable speed
Full speed	49.5k RPM	Nominal full stable speed
Maximal speed	50.5k RPM	Limiting speed

7.2.2 Vibration monitoring sensors

As sensors of vibrational behavior of the tested engine accelerometers were used. Experiments were undertaken at compressor and at turbine stage. But due to a high temperature at turbine stage it was not possible to perform there long-term measurement. However it was proven that compressor stage is very well representative for the judgment about the overall engine condition.

The sensors were firmly attached using customized holders tightened by screws joining the compressor flange to the middle part of the engine as depicted in the Fig. 7.3. Orientation of accelerometer's axis of sensitivity was radial and axial, while radial vibrations are usually more important for vibrodiagnosis. Two kind of accelerometers were utilized, see details in Tab. 7.2.

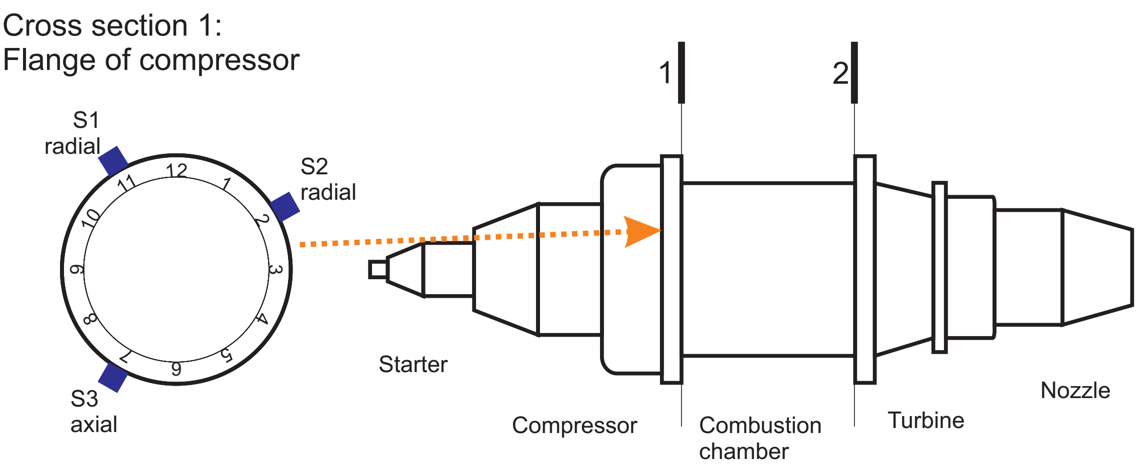

Figure 7.3: Position of sensors

To cover the general engine behavior the expected frequency range of the measured signals is several multiples of engine's rotating frequency (1st order frequency).

Maximal expected frequency is at 26th multiple of 1st order frequency which is related to the turbine stage. The accelerometer S3 does not cover the full expected frequency range however it can be used for relative comparison of vibrations at low frequencies, but must not be used for measurement of absolute value of vibrational energy or for absolute values of amplitudes in frequency spectrum.

Table 7.2: Vibration monitoring sensors details

Sensor mark	Sensor type	Dynamic range	Frequency range	Position and orientation	Engine's part
SN 1	Endevco 75-10	± 500 g	50 kHz	11 O'Clock, Radial	Flange of compressor
SN 2	Endevco 75-10	± 500 g	50 kHz	2 O'Clock, Radial	Flange of compressor
SN 3	Bruel&Kjaer BK4507	± 50 g	6 kHz	7 O'Clock, Axial	Compressor, fuel input

7.2.3 Tested engine's conditions

The engine was monitored at different regimes of operation (7.1) and at different states of condition (7.3).

The combination of regime and condition is marked as A, B, C and D:

Table 7.3: States of engine's condition

Mark	Regime	Description
A	Full speed	Health state
B	Full speed	Serious compressor damage
C	Reduced speed	Minor Damage
D	Reduced speed	Serious compressor damage

Mark A stands for the full speed regime when the engine was considered as in health state. See a short snapshot from the signal taken by sensor S1 in the upper graph in Fig. 7.4 and amplitude spectrum in Fig. 7.4. In the signal there is present a frequency peak at 1st order frequency and also smaller peaks at 2nd and 3rd order frequency. Further there is a significant amount of energy between frequencies 6 kHz – 10 kHz which may be caused by gasdynamic vibrations. Any possible bearings defects were not identified in the signal particularly due to the fact that no sensor was placed directly at the bearing house.

However the analysis of the vibrational signals indicate imbalance of the machine, the experimental engine is normally operated and is considered as health.

Mark C indicates the state of the engine as minor damage when it can be operated for short period of time at reduced speed.

Marks B and D are related to a state after a serious damage when the blades of compressor were damaged by foreign object. The overall energy of signal rose at the whole frequency range, and strong frequency peaks which are not related to any multiple of order frequency emerged. Mark B indicates a run at full speed and mark D at reduced speed.

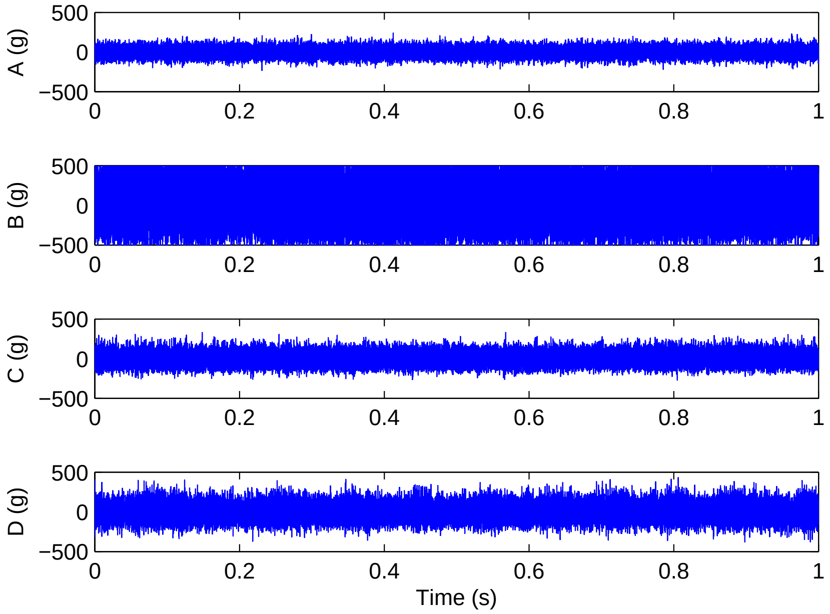

Figure 7.4: Illustration of 1 s record of signals acquired by sensor SN 1

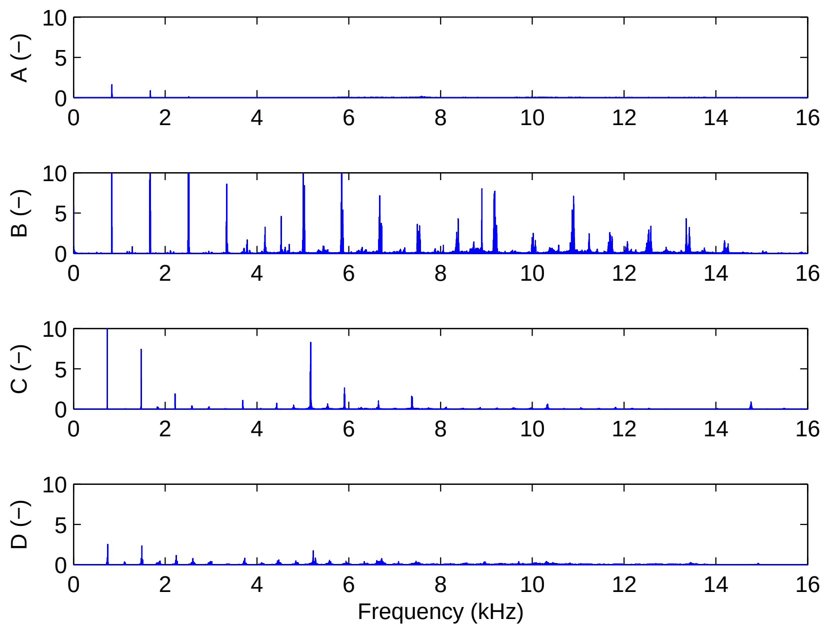

Figure 7.5: Illustration of amplitude frequency spectrum of signals acquired by sensor SN 1

7.3 WSN platform

A WSN platform Crossbow IRIS [47] from middle performance class (as defined in 2.2.1) was selected. See detailed description of a sensor node and a central node bellow.

7.3.1 Sensor node

The sensor node Crossbow IRIS (see Fig. 7.6) is based on the Atmel ATmega1281 low-energy, 8-bit microcontroller (8 KB RAM, 128 KB program flash memory, 512 KB data serial flash memory) and Atmel RF230 IEEE 802.15.4 compliant ZigBee Radio Frequency (RF) transceiver (2.4 GHz, max. data rate: 250 kbps). It is equipped with an eight-channel 10-b Analog-to-Digital Converter (ADC) and powered by a pair of AA batteries. The nodes are programmed under the TinyOS

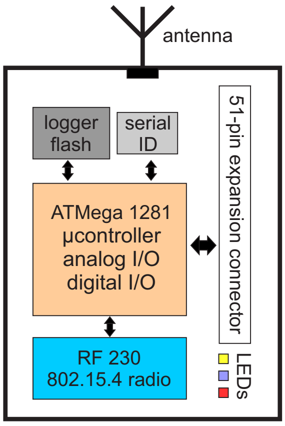

Figure 7.6: Sensor node architecture, [47]

operating system (MoteWorks distribution), which employs event-based programming, see its description in Section 2.2.2. MoteWorks also employs TinyOS support for mesh networking, time-synchronization up to 1 ms, bi-directional communication and Over-the-Air-Programming (OTAP). The achievable sampling frequencies of this setup are 1.8 kHz, 6.3 kHz and 7.7 kHz [47].

7.3.2 Central node

The central node is composed of Crossbow MIB520 gateway and a laptop running under Windows operating system. MIB520 has the same architecture as IRIS sensor node and ensures just communicational interface between sensor nodes and laptop. The central-node's software is realized in LabVIEW programming environment, which is very suitable for this purpose thanks to its event-based state machine nature.

7.4 Phases of operation

This section describes implementation and evaluation of there phases of operation: 1. Deployment and acquisition, 2. Learning and programming, and 3. Monitoring and Fault Detection proposed in Section 4.4.

7.4.1 Phase 1: Deployment and acquisition

Implementation of Phase 1 follows the scheme in Fig.4.8 described in Section 4.4.1.

Deployed sensor nodes are equipped with an acquisition program and wait for the trigger command from the central node to star the measurement.

Central node initiates the measurement when the engine operational conditions are met, i.e. when the engine is at constant speed. Trigger message is spread to the waiting sensor nodes and they start acquisition simultaneously.

Sensor nodes perform its acquisition at the highest possible sampling rate: 7.7 kHz. In this experimental setup the signals previously recorded by conventional equipment were low-pas filtered to meet the Nyquist sampling limit and then fed to the sensor nodes.

Record of 3000 samples is recorded, which means 390 ms of vibrational signal is acquired. Because of the restricted length of one message, the samples are divided into 150 messages by 20 samples and are then sent stepwise to the central node, which takes around 1.5 s; see Fig. 7.7.

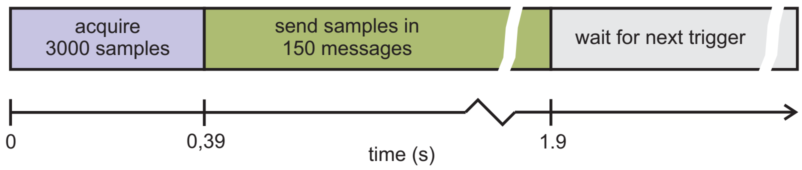

Figure 7.7: Scheduling of the signal acquisition program

To acquire sufficient number of samples for the learning phase, the acquisition cycle is repeated 20 times. The individual acquisition cycles of certain nodes are synchronized and are fully controlled by the central node.

7.4.2 Phase 2: Learning and programming

Implementation of Phase 2 follows the scheme in Fig.4.9 described in Section 4.4.2.

Raw signals acquired in the Learning phase from sensor nodes S1, S2 and S3 are stored in central node.

Character of the engine: The most important characteristics is that it is single shaft turbojet engine with single stage radial compressor and single stage axial turbine, two bearings and no gearbox. See detailed description in 7.2.

Character of the signals: While sensor nodes S1 and S2 sense radial engine's vibration, the S3 has axial orientation, see details in Section 7.2.2 and Table 7.2. All the sensors are close to the 20-blades compressor.

Sensor-node's maximal sampling frequency 7.7 kHz limits the available frequency range for signal analysis only up to 3.85 kHz, which covers 4 multiples and 5 multiples of engine's 1st order frequency at full speed and at reduced speed respectively. Given frequency range allows to focus on general engine's behavior only, which implies selection of suitable methods described in Section 6. However the 7.7 kHz sampling frequency is the minimal for engine's health monitoring.

Energy resources: The energy resource of the sensor node is a 2xAA battery or primary cell pack, which capacity could be between 3200 mAh to 4500 mAh. The IRIS sensor node is experimental platform so its energy resource is not designed to accomplish requirements of engine's health monitoring system. However the design of such energy source is out of the scope of the thesis, see 2.2.1.

Computational resources: The sensor-node's computational capability is given mainly by its 8-bit MCU architecture and 8 KB RAM memory.

Only 1 ms time synchronization of signals acquired by different sensor-nodes and no tacho pulse available at the sensor node level implies that the signal processing methods, described in Table 6.2, which need those inputs are not applicable

Due to above described limits the applicable methods of vibrational signal processing are non-conditioned time-domain methods and amplitude spectral analysis based on Fast Fourier Transform.

Extraction of features

As the first step it is necessary to determine a length of a signal snapshot for the feature extraction. This decision is driven mainly by demanded time-frequency resolution based on the character of analyzed signal.

On the one hand a long snapshot improves frequency resolution i.e. allows to distinguish between a small change of frequency of given signal's elements and also helps to make the time-domain features more prone to random fluctuations. On the other hand a short snapshot allows to better focus on immediate incident and thanks to less time needed for sampling and processing saves the power consumption of a sensor node.

For given setup the most important factors are the sensor-node's energy consumption while a good performance of features is accomplished.

For this demonstration a length of snapshot 256 samples at 7.7 kHz sampling frequency was selected. It gives approximately 33 ms of signal with theoretical 30 Hz frequency resolution. Selected values serve very well for feature extraction needed for engine general behavior monitoring.

The time-domain features RMS (based on equation 6.1), crest (equation 6.2) and kurtosis (equation 6.6) were selected for demonstration. Those three features are robust and describe the general behavior while are inexpensive for in-node computation.

RMS measures the power content of the signal and is very robust but is sensitive to engine's rotational frequency. Crest factor is, compared to RMS, more sensitive to incipient damage, when the peak values develops and at the same time the overall vibrational energy still remains the same. When also RMS develops with increasing damage the crest factor loses its sensitivity. Based on the expert knowledge [53] kurtosis is related to a machine damage especially when it exceeds value 3.

Fig. 7.8 shows time-domain features computed in sensor nodes SN 1, SN 2 and SN 3 for signals acquired during engine's run in state A: full speed, health state and in state C: reduced speed, minor damage.

The features in feature space for all three sensor nodes during the Regime A create a compact bodies which slightly differs from each other, but the ranges of individual features axes are still comparable. However the situation for regime C is different, where especially SN 3 does not give good expectation for successful one-class classification.

For given time-domain features it takes 309 ms to in-node process one point in feature space. Its asymptotic complexity is $O(N)$, where N is number of samples.

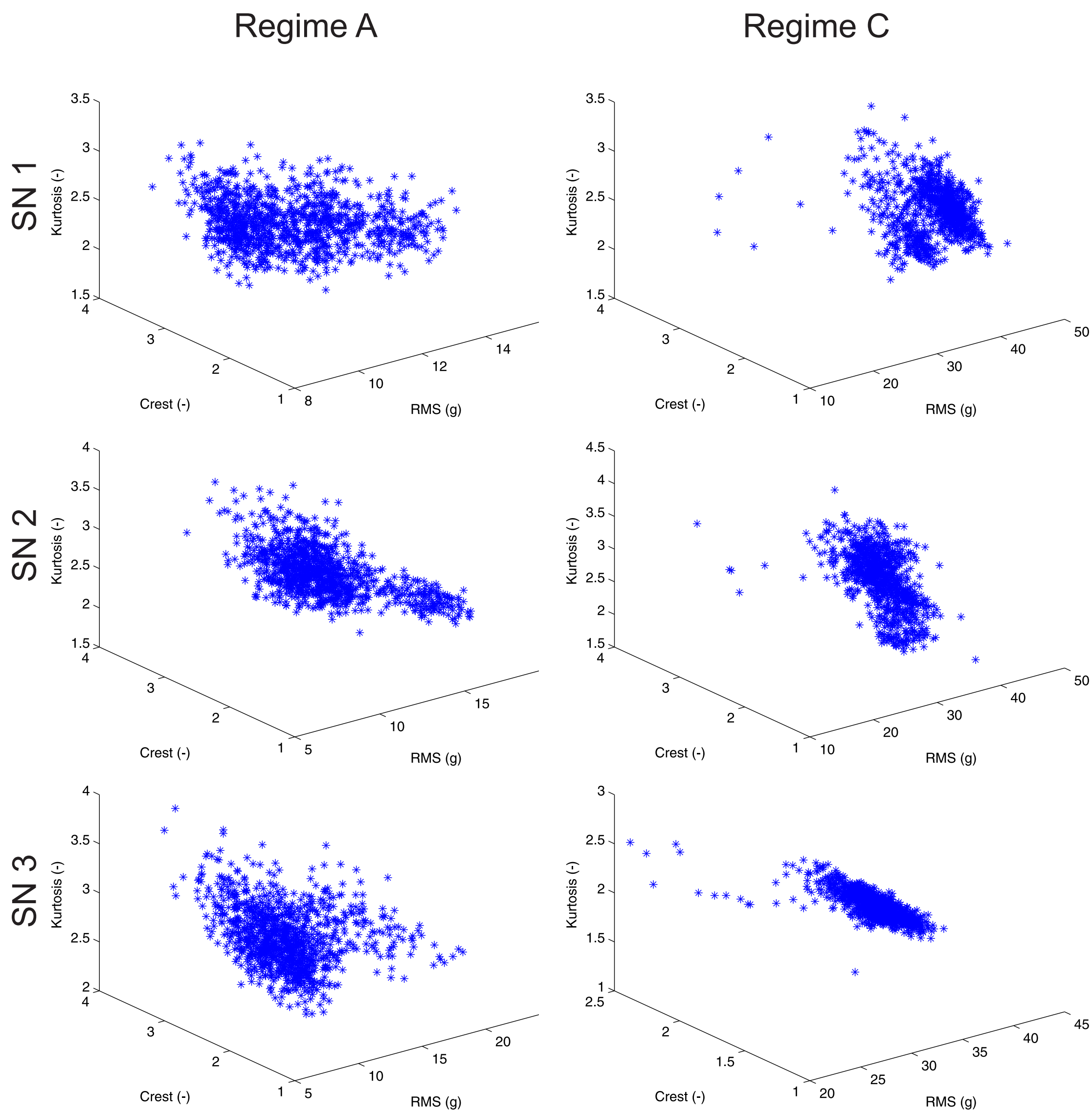

Figure 7.8: Time-domain features extracted on engine regimes A and C

To obtain frequency-domain features the Fast Fourier Transform is implemented. The amplitude spectrum of the 33 ms signal snapshot results in 128 frequency bins, i.e. 128 features in feature space. The in-node processing takes 426 ms and its asymptotic complexity is $O(N \cdot \log N)$, where N is number of samples.

When the feature space needs to be reduced the most robust technique is to obtain energy of several given frequency range intervals or simplified spectrum mask.

This approach is shown in Fig. 7.9 where feature space is reduced to 10 equidistant spaced frequency bands.

Fig. 7.9 shows example of features extracted from frequency domain for sensor nodes SN 1, SN 2 and SN 3 for signals acquired during engine's run in state A in state C. The

Another approach like tracking the energy of narrow frequency range focused to 1st order frequency and its multiplies may be applied while it increases the computational complexity, looses robustness and may require additional information from central node.

Figures 7.8 and 7.9 depict exactly the situation when signals representing health state are acquired during the Learning phase, no signals representing the other states of engines. Suitable features has to be created based on the knowledge about character of signal and engine and are limited by sensor-node's computational and energy resources as proposed in Fig. 4.9.

Table 7.4: Marking of extracted features

Mark	Origin of feature	Detail
F-TD-3	Unconditioned raw time-domain signal, 33 ms snapshot	RMS, crest, kurtosis
F-FD-8	Amplitude spectrum, 128 bins, 0–3.85 kHz	Amplitude spectrum reduced to 8 equidistantly spaced frequency bands.
F-FD-10	Amplitude spectrum, 128 bins, 0–3.85 kHz	Amplitude spectrum reduced to 10 equidistantly spaced frequency bands.
F-FD-14	Amplitude spectrum, 128 bins, 0–3.85 kHz	Amplitude spectrum reduced to 14 equidistantly spaced frequency bands.

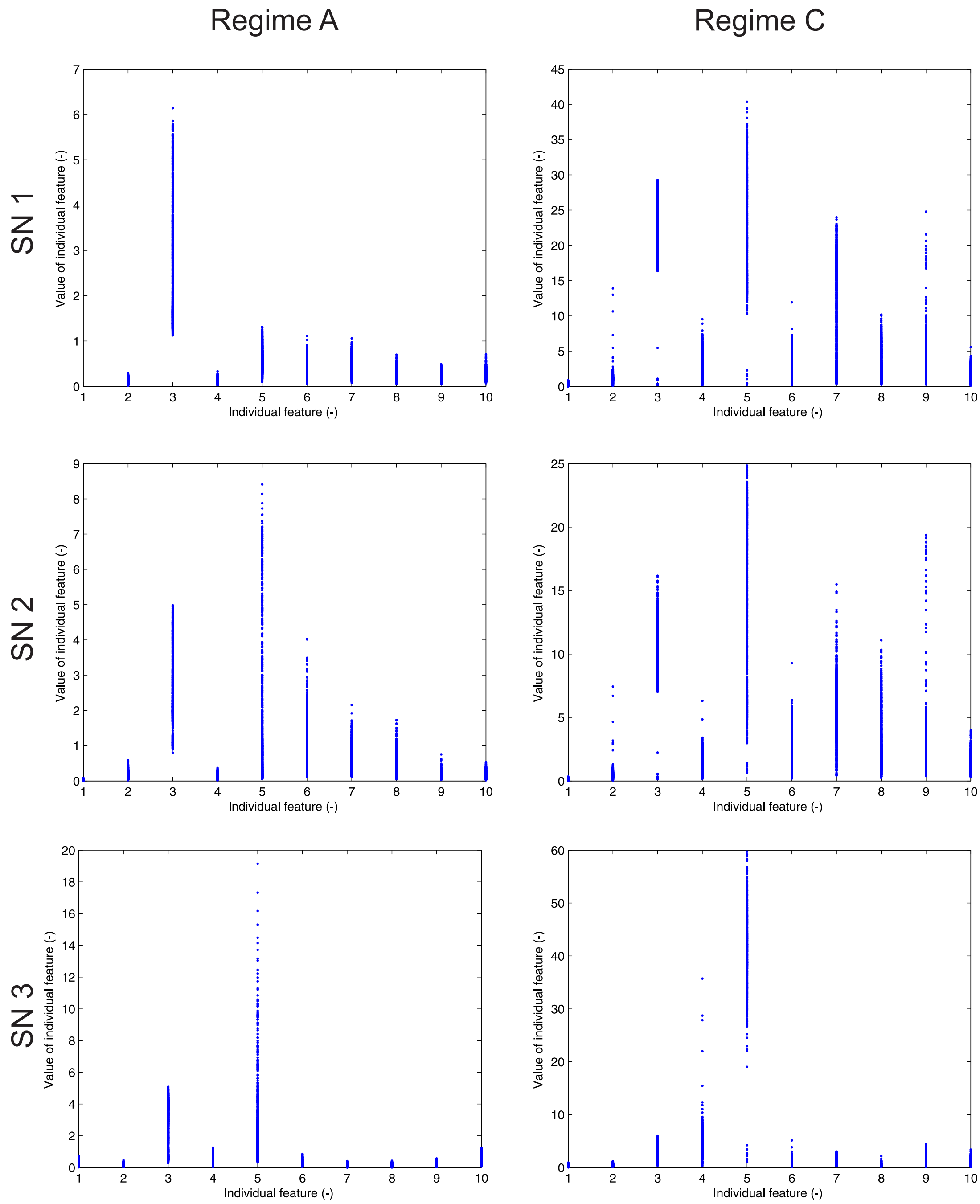

Figure 7.9: Frequency-domain features extracted on engine regimes A and C

Implementation in-node fault detection method

The next step within the Learning phase depicted in Fig. 4.9 is creation and implementation of in-node fault detection method. In this phase there are available vibrational signals from the sensor nodes related to the health state of the engine only. There are no signals representing any engine's fault. For this reason the one-class classification, proposed in Section 5.3 is employed.

Due to sensor-node's computational and memory limits, limited size of training dataset and thanks to nature of features, the density-method-based normal one-class classifier based on Eq. 5.1 is selected.

While the computational complexity of the training phase, which is held in central node is $O(d^3)$, the complexity of testing phase (i.e. fault detection), which is held in sensor node, is $O(d)$, where d is the size of feature space.

To avoid numerical instabilities and to reduce the computational difficulty the Mahalonobis distance $f(\mathbf{x})$ 7.1 is used instead of 5.1.

Then the classifier function is:

$$f(\mathbf{x}) = (\mathbf{x} - \mu)^{\mathrm{T}} \cdot \mathbf{C}^{-1} \cdot (\mathbf{x} - \mu) \tag{7.1}$$

where $\mathbf{x}$ is the classified object in the feature domain and the parameters of the model μ, $\mathbf{C}^{-1}$.

The classifier creates a continuous area around the training objects in the feature domain. If the tested object belongs to this area it is classified as a target, i.e. health state. If it belongs outside the area it is an outlier, i.e. it detects a fault. So the result of fault detection is $h(f(\mathbf{x}))$:

$$h(f(\mathbf{x})) = \begin{cases} \text{target} & \text{if} \quad f(\mathbf{x}) \leq \Theta \\ \text{outlier} & \text{if} \quad f(\mathbf{x}) > \Theta \end{cases} \tag{7.2}$$

where Θ is the threshold directly related to the false negative error rate found during the testing process.

The classifier training is held in computational unrestricted central node. For this demonstration the MATLAB DD Tools toolbox was employed. There is individual classifier trained for each individual sensor node.

In the classifier set point (threshold Θ) must be set a priori. This this setup results in given false negative error which is related to false alarms. It is preferred to set this threshold to create at least 5% false alarm rate. This improves classifier generalization and robustness and lowers false positive rate. The engine health monitoring system overall false alarm rate is mitigated in later steps.

Selected method of feature extraction and fault detection is implemented into prepared nesC program template as depicted in Fig 4.9. Then the program is automatically compiled for given target sensor node running under TinyOS.

When all programs are ready to upload Over-the-Air-Programming is initiated. It is driven by the central node and is finished when all sensor node reboot to their specific program.

7.4.3 Phase 3: Monitoring and Fault Detection

As proposed in Section 4.3.2 and 4.4.3 in the Monitoring and Fault Detection phase the sensor node starts performing its its default operation: vibrational signal sampling and in-node fault detection executing, see Fig. 7.10 a).

The central node has the ability to send a command to sensor node which adapts sensor-node's default behavior by adjusting T_T and T_M periods. Further the central node is capable to send the command with request for features and raw signal samples, see Fig. 7.10 b) and c) respectively.

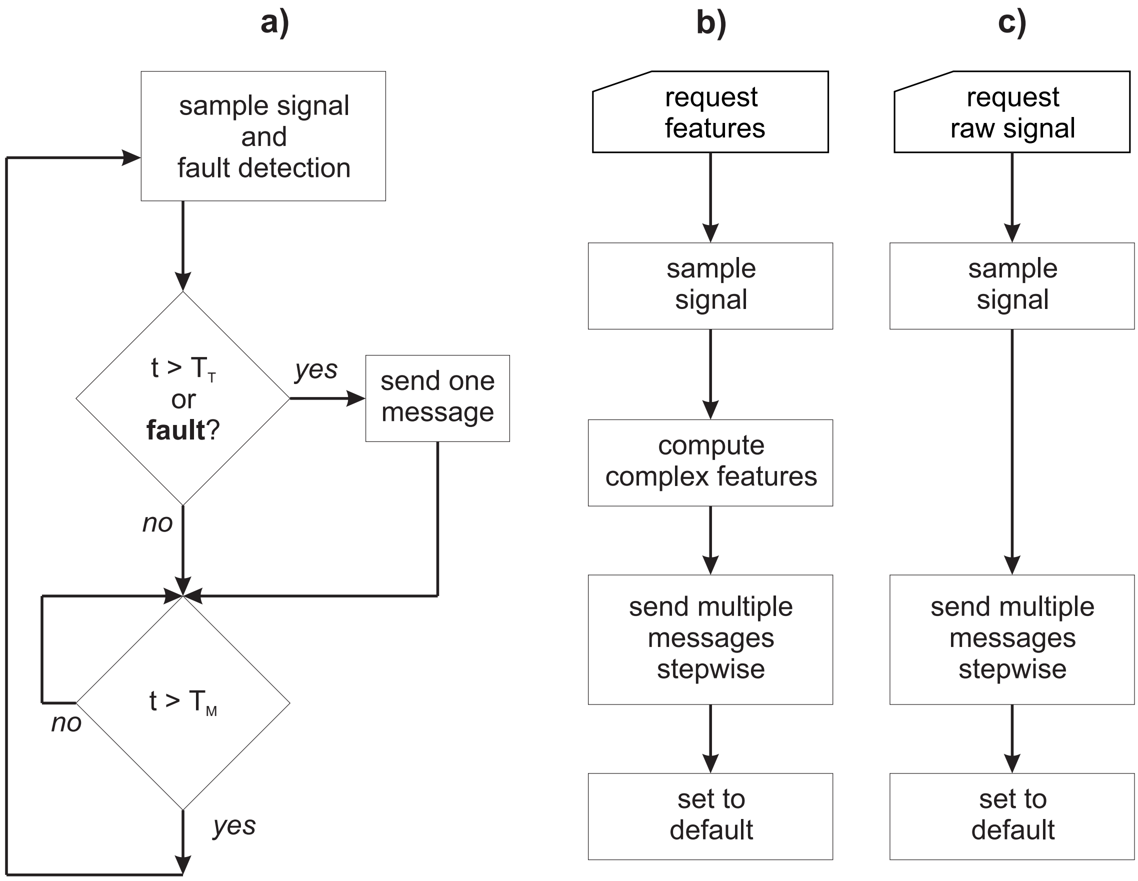

Figure 7.10: In-node program

The in-node operation was implemented in nesC language and was executed under TinyOS operational environment running in sensor nodes. The central-node operation was implemented in LabVIEW language running under Windows operating system.

7.5 Summary

This chapter brings the experimental evaluation of the proposed novelty detection method and of proposed WSN framework for vibrational condition monitoring. The evaluation was carried out by means of designed experiment on small turbojet engine TS20. The proposed framework was successfully implemented using WSN platform IRIS and its capability was confirmed. It was proven that applied approach serve to the condition monitoring purpose as it was designed.

Chapter 8

Results evaluation

This chapter brings the detailed evaluation of results obtained by designed experiment described in Chapter 7.

The performance of in-node fault detection method is evaluated i Section 8.1. While the achievements of proposed WSN framework are summarized in Section 8.2.

8.1 Performance of in-node fault detection method

Performance of in-node fault detection method depends both on contribution of extracted features and by performance of fault detection technique itself. For features obtained from vibrational signal, their performance is higher as they react to a machine's faulty behavior with higher sensitivity and vice versa. For a fault detection method its performance is evaluated based on its ability to detect a fault and on its fault positive and fault negative errors.

See bellow the detailed evaluation of achieved results.

8.1.1 Features performance

Fig. 8.1 shows comparison of time-domain features F-TD-3 (7.4) measured and extracted at given sensor nodes: top SN 1, middle SN 2, bottom SN 3 (see the sensor nodes marking in Tab 7.2). The group of feature objects represents the whole engine run in given health condition.

Left part of Fig. 8.1 compares features extracted on signals acquired during the engine's full-speed run at heath state (A) and engine's full-speed run at engine's serious damage (B), as defined in Tab. 7.3. It is obvious that these two states of health may be easily distinguished with given features for all three sensor nodes. Both states create a compact body in feature space.

Right part of Fig. 8.1 compares features extracted on signals acquired during the engine's reduced-speed run at minor damage (C) and engine's reduced-speed run at engine's serious damage (D). In this comparison the feature objects representing engine's states are mixed up together and it is not possible to easily distinguish between give engine's states.

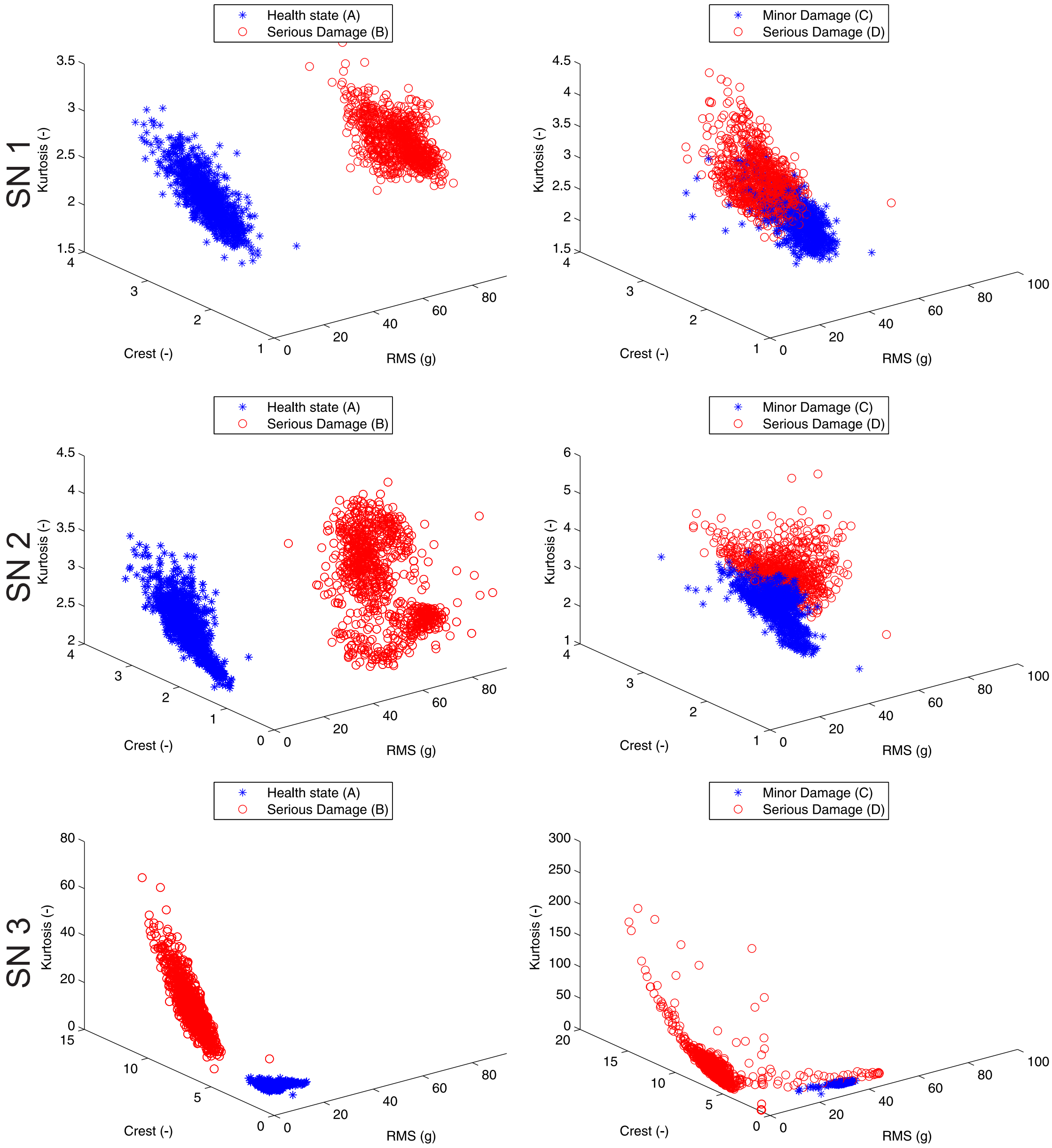

Figure 8.1: Comparison of time-domain features F-TD-3 for sensor nodes SN 1, SN 2, SN 3 and different states of engine's health

Fig. 8.2 shows comparison of frequency-domain features F-FD-14 measured and extracted at given sensor nodes: top SN 1, middle SN 2, bottom SN 3 (see the sensor nodes marking in Tab 7.2). The feature set represents one snapshot taken during the engine run in given health condition. Therefor Fig. 8.2 gives the example of feature space rather than description of the features behavior during the whole engine run.

From the left part of Fig. 8.2 which represents comparison between engine's states A and B its obvious that these two states differs significantly from each other.

While in the right part of Fig. 8.2 the feature objects are more close to each other, it is still obvious engine's states C and D are represented by different pattern.

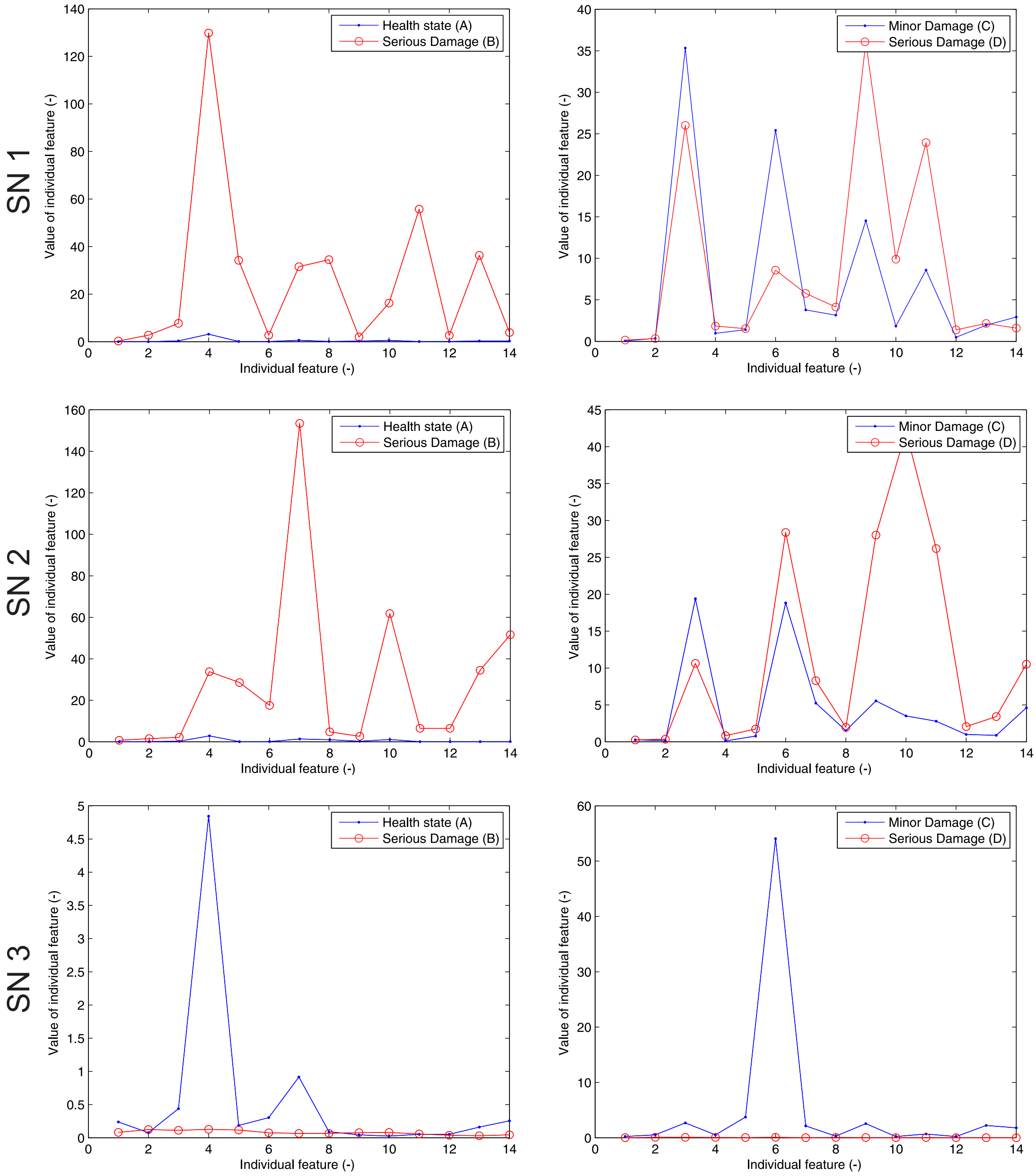

Figure 8.2: Example of frequency-domain features F-FD-14 for sensor nodes SN 1, SN 2, SN 3 and different states of engine's health

Demonstration of features performance given in Fig. 8.1 and Fig. 8.2 shows that time-domain features extracted from unconditioned raw signal are robust and cover the difference between engine's health state and serious damage very well. However they do not perform very well in situation where engine's health and fault states are more similar to each.

The features extracted by robust technique described in 7.4.2 from frequency-domain could give much better resolution between the engine's health states even for distinguishing between minor and serious damage. On the other hand the frequency domain features are computationally more demanding. It takes 38 % more time of the in-node processing compared to the frequency domain features. See details in Section 7.4.2. In the other words they are responsible for 38 % more energy consumption of sensor node computation.

8.1.2 One-class classifier performance

Implemented method of novelty fault detection was one-class classifier. Its main criteria is its sensitivity, i.e. ability to reveal a fault correctly. This ability is measured by the false negative error rate (i.e. false alarm rate) and by false positive error rate (i.e. missed fault).

In the Table 8.1 see the comparison of classifier performance for different type of features. Feature set marked as F-TD-3 are 3 features RMS, crest, kurtosis extracted on time-domain as described in 7.4.2. Feature set marked F-FD-8 and F-FD-14 are created from frequency domain (7.4.2) so they create set of 8 and 14 features respectively (see the features summary in Tab. 7.4).

Table 8.1: Performance of classifier

Situation		False negative rate (%)	False positive rate (%)		
Training : Testing	Sensor Node	For all features	F-TD-3	F-FD-8	F-FD-14
A : B	SN 1	5	0	0	0
A : B	SN 2	5	0	0	0
A : B	SN 3	5	0	0	0
C : D	SN 1	5	48.28	0.44	0
C : D	SN 2	5	2.55	0.11	0
C : D	SN 3	5	0	0	0

First three rows of Table 8.1 depict the situation when classifier was trained on features extracted from signal representing the health state of the engine at full speed – mark A (see the definition of engine's states marks in Table 7.3). Then the classifier is tested on signal marked B – seriously damaged engine at full speed. The false positive rate for all sensor nodes and different feature sets is 0 which means that faulty behavior was reliably detected, while the false negative rate is for 0.05. False negative rate is set during the training process as described above in 7.4.2.

The forth and fifth row of the Table 8.1 shows very poor performance of time-domain feature set F-TD-3, while feature sets F-FD-8 and F-FD-14 performing very well.

The last row of the Table 8.1 stands for sensor node SN 3 which does not have sufficient frequency and dynamic range to be used for reliable measurement of signals produced during engine regimes C and D. Even the fault detection method performs very well in this case the results of this sensor node may be misleading and must be avoided.

Fig. 8.3 shows the receiver operating characteristic (ROC) for sensor node SN 1 of the classifier trained on signal C – minor damage, reduced speed, and tested on signal D – serious damage, reduced speed.

ROC curve further shows that F-TD-3 feature set is not suitable for this specific fault detection. While the feature sets F-FD-8 and F-FD-14 are performing very well. The set point The false negative error assigned to 0.05 gives them a good set point.

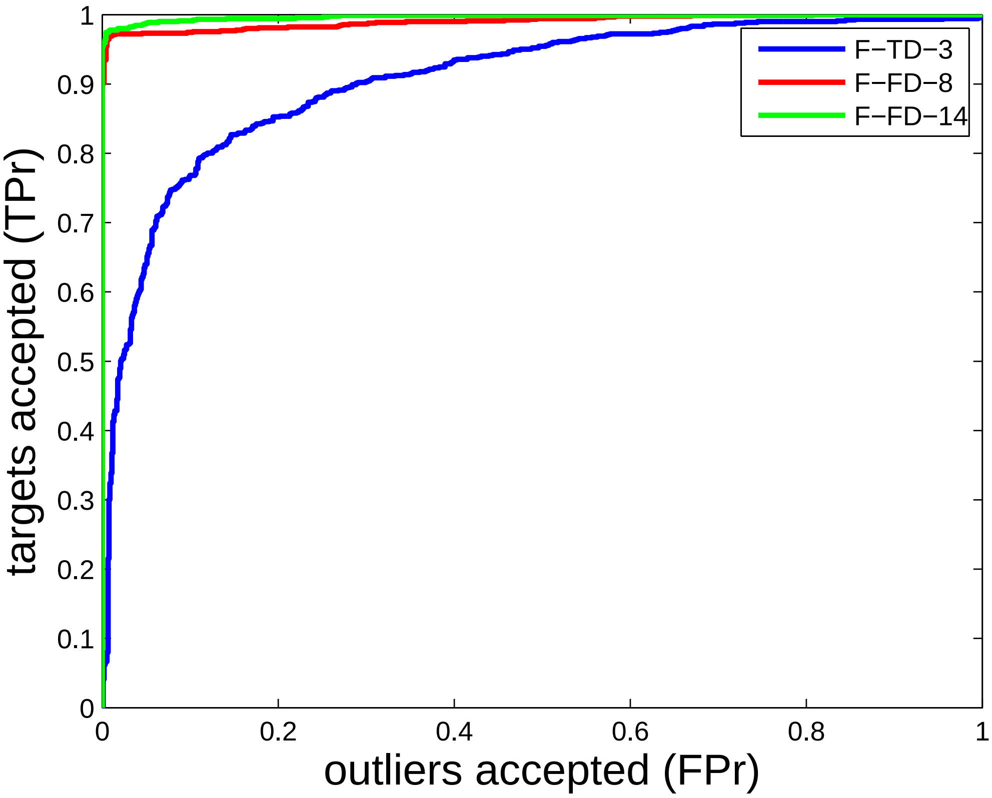

Figure 8.3: Performance of classifier trained on C tested on D for Sensor Node 1

However this performance comparison is based on engine's fault states available in this demonstration, they are not available in the Learning phase. So the classifier set point (threshold Θ) must be set a priori. It is preferred to set this threshold to create at least 5% false alarm rate. This improves classifier generalization and robustness and lowers false positive rate. The engine health monitoring system overall false alarm rate is mitigated by monitoring system adaptive behavior.

The results of fault detection method performance demonstrate that proposed method of in-node fault detection is suitable for real jet engine condition monitoring.

The computationally less demanding fault detection based on time-domain features performed well to distinguish between healthy state and a serious engine fault, which was still difficult to reveal by operator.

The computationally more demanding in-node fault detection based on feature set from amplitude spectrum was able to reliably detect engine incipient fault even in all conditions.

8.2 Distributed WSN framework achievements

Distributed WSN framework for vibrational condition monitoring was designed especially to accommodate computationally intensive methods of vibrational signal processing and methods of condition monitoring in computationally weak wireless sensor network.

The key feature of this novel WSN framework is the ability to maximize the immediate fault detection capability while provide long-term monitoring.

Evaluation of the framework is based primarily on its capability to effectively employ the developed in-node fault detection methods and the ability to react to ambiguous machine states by temporarily changing the diagnostic focus. This was achieved by proposed system's **Adaptivity**, see its results evaluation in Section 8.2.1.

Moreover the introduced distributed WSN framework enables the vibrational condition monitoring system to completely **Reconfigure** its setup. Reconfigurability is summarized in Section 8.2.2

8.2.1 Evaluation of Adaptivity

Fig. 8.4 brings the comparison of in-node operations during the monitoring phase.

The Fig. 8.4 a) shows the default in-node operation during the monitoring phase when each sensor nodes acquires 33 ms of vibrational signal and extracts features F-FD-14 in-node and performs fault detection. In-node operation of in-node feature extraction and classification takes 426 ms. Here the sensor's node transmit update period T_T is set to be equal to measurement update period T_M so each time the sensor node performs in-node fault detection it also sends its result to central node. To send one message it takes on average 10 ms including acknowledgment from central node.

As described in Section 7.4.2 the in-node fault detection method is in the learning phase set so it produces 5% of false alarms. This setup helps to achieve a good generalization of the method.

The Fig. 8.4 b) depicts the situation when in-node fault detection methods has detected engine's fault and sent that result to central node. The central node based on also on the information from the other sensor nodes and information about engine's operational regime decided to command the sensor node to adapt its in-node method. Instead of default operation the sensor node computes the amplitude spectrum of vibrational signal and sends the 128 spectral values in 7 messages to the central node.

Based on this information the central node performs detailed analysis to decide if that was a false alarm or engine's faulty behavior. If the amplitude spectrum is not sufficient of central-node's decision it commands the central node to acquire and send the raw samples of engines vibrations as depicted in the Fig. 8.4 c).

The amount of data transfered from the sensor node to the central node is the best benchmark to compare level of data compression. In implemented setup the atomic unit of data transfer is one message which carries payload of 20 numerical values. As stated in Section 4.2.2 the data transferring is in WSN the most energy consuming operation so the less data are transfered the more is prolonged the sensor nodes life time. Moreover it is not possible to stream raw samples simulta-

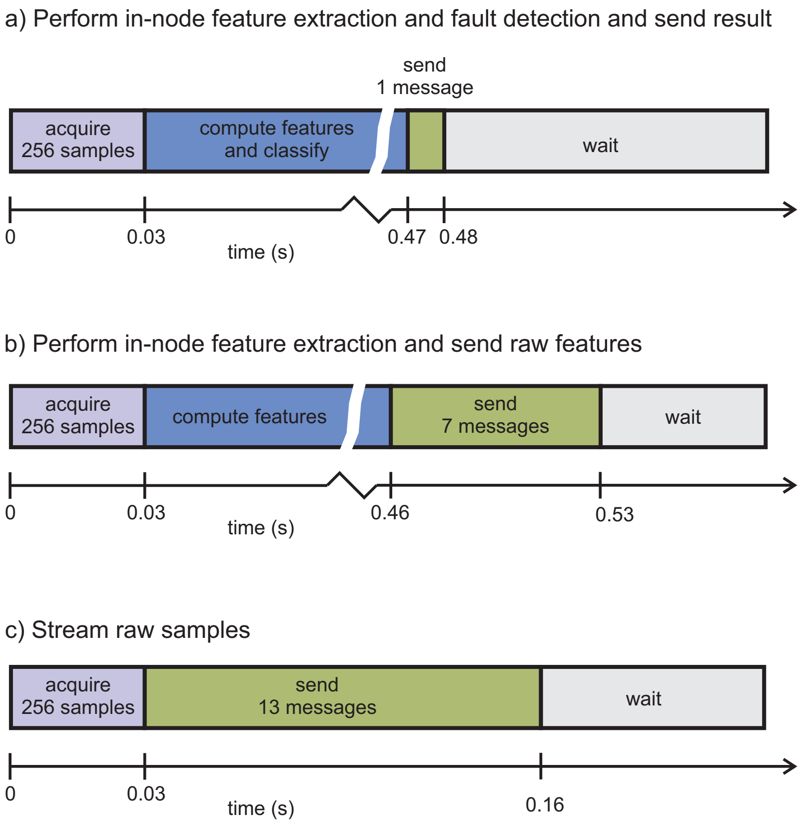

Figure 8.4: In-node operation: Adaptivity

neously from a number of sensor nodes due to low data through put of low-power communication protocols used for WSNs.

Fig. 8.5 compares the amount of data transfered for given setup. The baseline creates the **raw data sampling** as depicted in Fig. 8.4 c) which repeats with period 1 s. It is obvious that number of messages transfered during the raw data streaming is far more higher than for the other two situations: Simple in-node operation and Adaptive in-node operation. In Fig. 8.6 see the detailed comparison of Simple in-node operation and Adaptive in-node operation.

Simple in-node operation depicts the situation when sensor node regularly executes vibrational signal sampling and in-node feature extraction and computation of result of in-node fault detection with Measurement Update Period $T_M = 1$s. And it sends the result of in-node fault detection to the central node with Transmit Update Period $T_T = 2$s.

Finally the **Adaptive in-node operation** depicts the situation when sensor node performs its default in-node operation: vibrational signal sampling, in-node feature extraction and in-node fault detection with periods $T_M = 1s$ and $T_T = 2s$ as described for In-node operation above. Moreover when a fault is detected by sensor node, central node decides to command this sensor node to send its values of amplitude spectrum in 7 messages as depicted in Fig. 8.4 b).

The Adaptive in-node operation shown in Fig. 8.6 directly corresponds to the situation when the engine runs in its health state in normal operational conditions. So the 5% of false alarms raised by in-node fault detection are resolved by described adaptation of behavior. The central node executes detailed analysis and intercorporates also the results of the neighboring nodes so it is able to evaluate the false

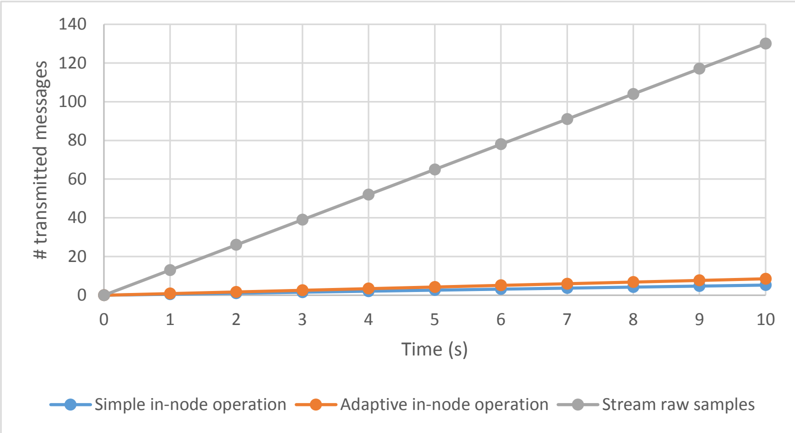

Figure 8.5: Messages transmitted

alarm created by single sensor node.

In the case that central node based on its information also detects the engine failure it commands the sensor node by shortening the T_M and T_T periods and by further commanding the sensor node to adapt for more demanding in-node operation up to by commanding it to send raw data samples as in Fig. 8.4 c).

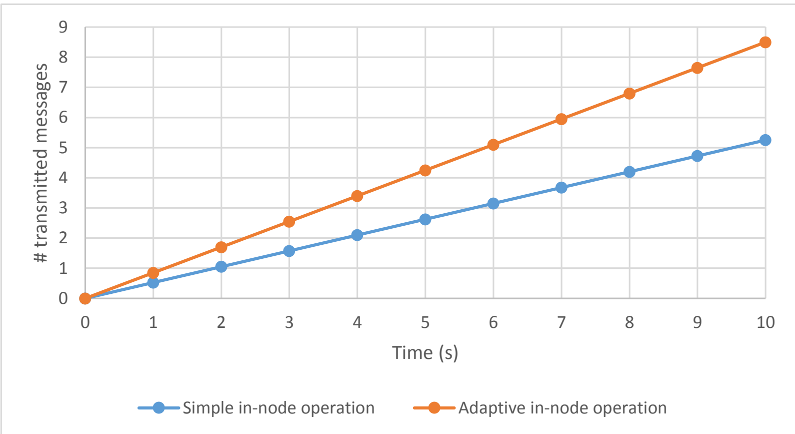

Figure 8.6: Messages transmitted: detail

In the situation that the engine is becoming to its failure the monitoring system is protected from sudden power source discharge by its Aggregated Energy Consumption proposed in Sec. 4.3.1. This approach rather provides raw signal snapshots describing the middle-term failure progress than a short-time information describing the beginning of the failure resulting in fully discharged and thus non-working monitoring system.

The amount of data messages transfered from the sensor node to the central node is the best benchmark to compare level of data compression.

The compression of transfered data is presented in Fig. 8.7 via ratio of reduction of messages transmitted from sensor node to central node for Simple in-node operation and Adaptive in-node operation. Messages Reduction Ratio (MRR) is computed based on Eq. 8.1:

$$MRR\,(\%) = 1 - \frac{\#Messages\ of\ Proposed\ Operation}{\#Messages\ of\ Raw\ Samples\ Streaming} \tag{8.1}$$

Fig. 8.7 shows how the reduction ratio changes with increasing Transmit Update Period T_T for constant Measurement Update Period $T_M = 1$s. For both reduction ratios of Simple in-node operation and Adaptive in-node operation change very rapidly for low values of T_T while for high values of T_T they converge to their maximal values.

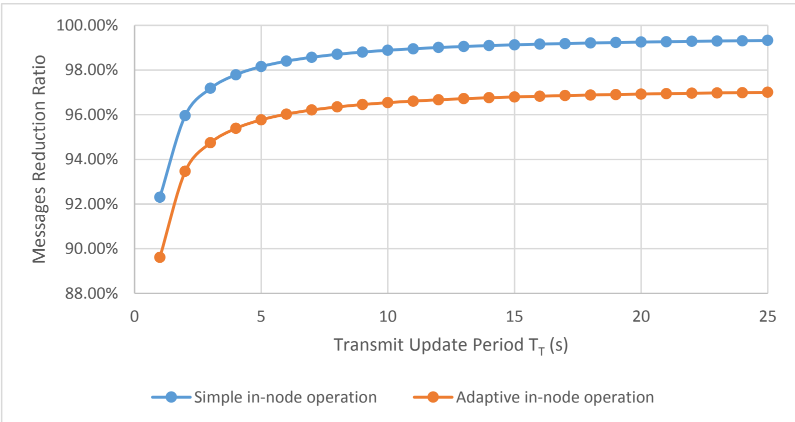

Figure 8.7: Reduction of transmitted messages compared to raw data streaming baseline

Table 8.2 further generalizes the messages reduction ratio for $T_T = T_M \cdot k$, where $k = 2, 3, 4 \ldots$ as proposed in Section 4.3.2

Transmit update period T_T is the key factor for reduction of the sensor-node's energy consumption and for setting of the immediate response of the monitoring system.

Even though Adaptive in-node operation has about 2 % lower Messages Reduction Ratio than the Simple in-node operation, it achieves almost 90 % reduction of messages transmitted form sensor node to the central node for $T_T = 1 \cdot T_M$ which gives the most immediate monitoring system reaction. Moreover the adaptive in-node operation actively solves the in-node false alarms and gives detailed information about the engine's state which is further used for fault diagnostics and condition trending.

Table 8.2: Reduction of transmitted messages

Transmit update period T_T	Raw data streaming	Simple in-node operation	Adaptive in-node operation
$T_T = 1 \cdot T_M$	0 %	92.3 %	89.6 %
$T_T = 3 \cdot T_M$	0 %	97.2 %	94.7 %
$T_T = 5 \cdot T_M$	0 %	98.2 %	95.8 %
$T_T = 10 \cdot T_M$	0 %	98.9 %	96.5 %
$T_T = 100 \cdot T_M$	0 %	99.5 %	97.2 %

8.2.2 Reconfigurability

In the case that the engine changes its default but still healthy vibrational behavior the monitoring system is protected from the sudden discharge by its Reconfigurability proposed in Section 4.3.3.

The Reconfigurability is demonstrated for the case when tested engine changed in its operable regime from Health State (mark A) to its Minor Damage state (mark C). Even there suddenly emerged a minor damage of the engine, the engine's operator decided to continue its operation in regime of reduced speed until its planned overhaul.

Due to the fact that engine changed its state considered as health and also the operational condition was changed from full speed to reduced speed the Acquisition and Learning phase of the monitoring system was executed as proposed in Section 4.4.

Although the state of minor damage allows continuing the engine operation it is necessary to increase monitoring activity to ensure the further damage increase is reliably and immediately detected. Thus the more computationally demanding but more precise frequency domain features F-FD-14 were selected. One-class classifier was trained on datasets acquired by individual nodes and complete in-node fault detection program was uploaded to the sensor nodes via OTAP.

Figure 8.3 demonstrates good performance of feature set F-FD-14 when tested on engine's serious damage at reduced speed.

Table 8.1 further shows that this setup has 0 false positive error while having demanded 5% false negative error, which was set to it in the Learning phase.

8.3 Summary

This chapter summarizes the evaluation of results obtained by experiment performed on small turbojet engine TS20. There is evaluation of performance of in-node fault detection method and also the performance of proposed WSN framework which implements adaptivity and reconfigurability.

Chapter 9

Conclusion

This work has investigated the aircraft on-board condition monitoring by employing Wireless Sensor Networks. It focuses especially on vibrational condition monitoring which is specific by its computational and communicational demands of signal processing. Even though that WNS are typically limited in communicational, computational and energy resources, this thesis proposes novel methods and approach on how to achieve immediate fault detection capability simultaneously with long term monitoring.

Implementation of the proposed system into aircraft application would primarily reduce maintenance costs and improve reliability and safety.

9.1 Contribution of this work

This work is contributing to the current state of the on-board vibrational condition monitoring of aircrafts' power power plants by its **original novel approach of distributed WSN based vibrational condition monitoring**.

Proposed approach allows to employ computationally intensive methods of vibrational signal processing and methods of condition monitoring in computationally weak wireless sensor network.

In-node fault detection method based on novelty detection was developed. It employs one-class classification which allows detection of novelty in vibrational signatures relative to the baseline representing healthy state of monitored machine.

Integral part of fault detection method is application of appropriate methods of vibrational signal processing to create feature space which is the input for classification. The approach based on character of vibration signal features extraction, the a priori knowledge of a monitored machine components and on WSN limits and resources was introduced and evaluated.

The fault detection method was evaluated via designed experiment on real turbojet engine running at different regimes of operation and different levels of degradation.

It was proven experimentally that the in-node fault detection method implemented in the sensor nodes detects incipient faults reliably. Detailed evaluation of the fault detection method is summarized in Section 8.1.

A novel distributed WSN framework allowing distributed signal processing to maximize the immediate fault detection capability while provide long-term monitoring was introduced.

The immediate fault detection capability while providing long-term monitoring was achieved by the following key attributes of the proposed WSN framework architecture:

- In-node fault detection
- Adaptivity and Reconfigurability
- Three phases of operation:
    1. Deployment and acquisition,
    2. Learning and programming,
    3. Monitoring and fault detection

The designed framework was implemented in the TinyOS WSN platform IRIS and was evaluated by the means of designed experiment on real turbojet engine.

It was experimentally proven that implemented Adaptive in-node operation reduced the energy consumption of radio communication transmission of 90 % compared the raw time signal streaming while achieving the same quality of information about the monitored machine's health, see details in Section 8.2.

Proposed methods and framework were **evaluated by the means of the experiment designed and performed on small jet engine TS20.**

This experiment demonstrates and justifies proposed work for next experimental implementation into current aircraft powerplant or drivetrain system.

9.2 Future work

Next step to further shift this work to higher Technology Readiness Level is to select a real aircraft platform and adjust the implementation to its specific requirements.

This step was initiated in cooperation with Department of Avionics, Faculty of aeronautics of Technical University of Kosice, Slovak Republic and with Department of Air Force and Aircraft Technology, University of Defence, Brno, Czech Republic. Within this cooperation the database of vibrational signals recorded for small jet engines: MPM20, TJ100 and TS20 was created. This database was opened to broader academic community.

Moreover the methods for fault diagnosis, fault prediction and information fusion which are to be held in the central node could be developed.

The methods of fusion based on classifier combining method presented in the author's paper [57] are very promising to improve fault detection results at the level of sensor in-node and central node operation.

General method of of quality-based sensor fusion for WSNs presented in [58], co-authored by the author of this thesis brings a powerful tool to fuse information at the all levels of engine condition monitoring system.

List of Publications

Work and publications related to the thesis

International Journals

[A1] J. Neuzil, O. Kreibich, and R. Smid, "A distributed fault detection system based on IWSN for machine condition monitoring," *Industrial Informatics, IEEE Transactions on*, vol. 10, pp. 1118–1123, May 2014.

[A2] O. Kreibich, J. Neuzil, and R. Smid, "Quality-based multiple sensor fusion in an industrial wireless sensor network for MCM," *IEEE Transactions on Industrial Electronics*, vol. 61, pp. 4903–4911, 9 2014.

International Conference Proceedings

[B1] J. Neuzil, O. Kreibich, and R. Smid, "Advanced signal processing in wireless sensor networks for MCM," in *CM 2012 / MFPT 2012 - The 9-th International Conference on Condition Monitoring and Machinery Failure Prevention Technologies*, (Oxfordshire, GB), pp. 1087–1093, 2012.

[B2] O. Kreibich, J. Neuzil, and R. Smid, "Application of wireless sensor networks in condition monitoring of rotating devices," in *CM 2012 / MFPT 2012 - The 9-th International Conference on Condition Monitoring and Machinery Failure Prevention Technologies*, (Oxfordshire, GB), pp. 1094–1101, 2012.

[B3] J. Neuzil, R. Smid, and O. Kreibich, "Rotary machine condition monitoring using one-class classification in wireless sensor networks," in *The Eight International Conference on Condition Monitoring and Machinery Failure Prevention Technologies*, (Dublin, IE), pp. 702–710, 2011.

[B4] J. Neuzil, R. Smid, and O. Kreibich, "Distributed classification in wireless sensor networks for machine condition monitoring," in *The Seventh International Conference on Condition Monitoring and Machinery Failure Prevention Technologies*, (Dublin, IE), pp. 1–11, 2010.

Local Publications

[C1] J. Mikes, O. Kreibich, and J. Neuzil, "A lightning conductor monitoring system based on a wireless sensor network," *Acta Polytechnica*, vol. 53, pp. 878–882, December 2013.

[C2] O. Kreibich and J. Neuzil, "Bezdratove sensorove site," *MM Prumyslove Spektrum*, vol. 11, pp. 38–39, březen 2011.

[C3] J. Neuzil and O. Kreibich, "Wireless sensor networks for machine condition monitoring," in *POSTER 2010 - Proceedings of the 14th International Conference on Electrical Engineering*, (Praha, CZ), 2010.

Work and publications not directly related to the thesis

International Journals

[D1] M. Kubinyi, O. Kreibich, J. Neuzil, and R. Smid, "Emat noise suppression using information fusion in stationary wavelet packets," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control (UFFC)*, vol. 58, pp. 1027–1037, 5 2011.

[D2] M. Kubinyi, O. Kreibich, J. Neuzil, and R. Smid, "Novel s-transform information fusion for filtering ultrasonic pulse-echo signals," *Przeglad Elektrotechniczny*, vol. 87, pp. 290–295, January 2011.

Table 9.1: Participation of authors on given papers.

Publication	Author	Authorship (%)
[A1]	Neuzil, J.	40
	Kreibich O.	30
	Smid, R.	30
[A2]	Kreibich O.	40
	Neuzil, J.	30
	Smid, R.	30
[B1]	Neuzil, J.	80
	Kreibich O.	15
	Smid, R.	5
[B2]	Kreibich O.	80
	Neuzil, J.	15
	Smid, R.	5
[B3]	Neuzil, J.	65
	Kreibich O.	15
	Smid, R.	20
[B4]	Neuzil, J.	65
	Kreibich O.	15
	Smid, R.	20
[C1]	Mikes J.	$33.\overline{3}$
	Neuzil, J.	$33.\overline{3}$
	Kreibich O.	$33.\overline{3}$
[C2]	Kreibich O.	65
	Neuzil, J.	35
[C3]	Neuzil, J.	95
	Kreibich O.	5
[D1]	Kubinyi, M.	55
	Neuzil, J.	15
	Kreibich O.	15
	Smid, R.	15
[D2]	Kubinyi, M.	55
	Neuzil, J.	20
	Kreibich O.	20
	Smid, R.	5

References

[1] R. B. Randall, *Vibration-based condition monitoring : industrial, aerospace, and automotive applications.* Wiley, 2011.

[2] R. Isermann, *Fault-Diagnosis Systems.* Springer, 2006.

[3] "Flight standards information management system, v3 - general technical administration, ch40 approve a maintenance reliability program for 121/135, safety assurance system: Approving 14 cfr parts 121 and 135 reliability programs," Federal Aviation Administration. [Online]. Available: http://fsims.faa.gov/WDocs/8900.1/V03%20Tech%20Admin/Chapter%2040/S_03_040_001.htm

[4] Health and Usage Monitoring Systems (HUMS). [Online]. Available: https://aerospace.honeywell.com/en/product-listing/health-and-usage-monitoring

[5] Honeywell. (2011) Hums infographic: Health and usage monitoring system. [Online]. Available: http://aviationweek.com/site-files/aviationweek.com/files/uploads/2014/11/Honeywell_HUMSinfographic.pdf

[6] Rolls-Royce, "Engine health management," 2016. [Online]. Available: http://www.rolls-royce.com/about/our-technology/enabling-technologies/engine-health-management.aspx

[7] Honeywell, "Heath and performance services that significantly reduce cost of ownership," Honeywell Internationall Inc., 2016. [Online]. Available: https://www51.honeywell.com/aero/common/documents/myaerospacecatalog-documents/BA_MSP-documents/PTDM_Brochure_-_Final.pdf

[8] T. Z. Kazem Sohraby, Daniel Minoli, *Wireless Sensor Networks: Technology, Protocols, and Applications.* Wiley, 2007.

[9] "Tinyos mission statement," 8 2011. [Online]. Available: http://www.tinyos.net/scoop/special/mission.html

[10] C. Buratti, A. Conti, D. Dardari, and R. Verdone, "An overview on wireless sensor networks technology and evolution," *Sensors*, vol. 9, no. 9, p. 6869, 2009. [Online]. Available: http://www.mdpi.com/1424-8220/9/9/6869

[11] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks.* Wiley, 2007. [Online]. Available: http://books.google.cz/books?id=170R-1aZsQYC

[12] A. Harb, "Energy harvesting: State-of-the-art," *Renewable Energy*, vol. 36, no. 10, pp. 2641–2654, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0960148110002703

[13] V. Gungor and G. Hancke, "Industrial wireless sensor networks: Challenges, design principles, and technical approaches," *Industrial Electronics, IEEE Transactions on*, vol. 56, no. 10, pp. 4258 –4265, oct. 2009.

[14] J. M. Gilbert and F. Balouchi, "Comparison of energy harvesting systems for wireless sensor networks," *International Journal of Automation and Computing*, vol. 5, no. 4, pp. 334–347, Oct. 2008. [Online]. Available: http://link.springer.com/article/10.1007/s11633-008-0334-2

[15] C. Rohner, L. M. Feeney, and P. Gunningberg, "Evaluating battery models in wireless sensor networks," in *Wired/Wireless Internet Communication*, ser. Lecture Notes in Computer Science, V. Tsaoussidis, A. J. Kassler, Y. Koucheryavy, and A. Mellouk, Eds. Springer Berlin Heidelberg, Jan. 2013, no. 7889, pp. 29–42. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-38401-1_3

[16] R. Bischoff, J. Meyer, and G. Feltrin, "Wireless sensor network platforms," in *Encyclopedia of Structural Health Monitoring.* John Wiley & Sons, Ltd, 2009. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1002/9780470061626.shm085/abstract

[17] L. Hou and N. Bergmann, "System requirements for industrial wireless sensor networks," in *2010 IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, Sep. 2010, pp. 1–8.

[18] "MEMSIC, Inc - Wireless Sensor Networks." [Online]. Available: http://www.memsic.com/wireless-sensor-networks/

[19] "ZigBee Wireless Networking Systems | Silicon Labs." [Online]. Available: http://www.silabs.com/products/wireless/mesh-networking/zigbee/pages/zigbee.aspx

[20] "IQRF." [Online]. Available: http://www.iqrf.org/weben/index.php

[21] "WSN-3202 - National Instruments." [Online]. Available: http://sine.ni.com/nips/cds/view/p/lang/en/nid/207088

[22] M. O. Farooq and T. Kunz, "Operating systems for wireless sensor networks: A survey," *Sensors (Basel, Switzerland)*, vol. 11, no. 6, pp. 5900–5930, May 2011. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3231431/

[23] R. S. Suet-Fei Li, "Low power operating system for heterogeneous wireless communication systems," 2001.

[24] V. Gungor and G. Hancke, "Industrial wireless sensor networks: Challenges, design principles, and technical approaches," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 4258–4265, Oct. 2009.

[25] A. W. Holger Karl, *Protocols and Architectures for Wireless Sensor Networks.* Wiley, 2005, no. ISBN: 978-0-470-09510-2.

[26] R. Wagner and R. Barton, "Performance comparison of wireless sensor network standard protocols in an aerospace environment: ISA100.11a and ZigBee pro," in *2012 IEEE Aerospace Conference*, Mar. 2012, pp. 1–14.

[27] T. Lennvall, S. Svensson, and F. Hekland, "A comparison of WirelessHART and ZigBee for industrial applications," in *IEEE International Workshop on Factory Communication Systems, 2008. WFCS 2008*, May 2008, pp. 85–88.

[28] S. Petersen and S. Carlsen, "WirelessHART versus ISA100.11a: The format war hits the factory floor," *IEEE Industrial Electronics Magazine*, vol. 5, no. 4, pp. 23–34, Dec. 2011.

[29] M. Nixon, "A comparison of wirelesshart and isa100.11a," Emerson Process Management, White Paper, 2012.

[30] T. Zheng, A. Ayadi, and X. Jiang, "TCP over 6lowpan for industrial applications: An experimental study," in *2011 4th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, 2011, pp. 1–4.

[31] P. Rawat, K. D. Singh, H. Chaouchi, and J. M. Bonnin, "Wireless sensor networks: A survey on recent developments and potential synergies," *The Journal of Supercomputing*, vol. 68, no. 1, pp. 1–48, Apr. 2014. [Online]. Available: http://dx.doi.org/10.1007/s11227-013-1021-9

[32] K. L. Marko Paavola, Riku-Pekka Nikula, "A wireless approach to water supply pump diagnosis," in *The Sixt International Conference on Condition Monitoring and Machinery Failure Prevention Technologies Proceedings*, 2009.

[33] P. Wright, D. Dornfeld, and N. Ota, "Condition monitoring in end-milling using wireless sensor networks," in *36th North American Manufacturing Research Conference*, 2008.

[34] M. reza Akhondi, A. Talevski, S. Carlsen, and S. Petersen, "Applications of wireless sensor networks in the oil, gas and resources industries," *Advanced Information Networking and Applications, International Conference on*, vol. 0, pp. 941–948, 2010.

[35] K. Tan, S. Huang, Y. Zhang, and T. Lee, "Distributed fault detection in industrial system based on sensor wireless network," *Computer Standards & Interfaces*, vol. 31, no. 3, pp. 573 – 578, 2009, industrial Networking Standards for Real-time Automation and Control. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0920548908000445

[36] I. Jawhar, N. Mohamed, and K. Shuaib, "A framework for pipeline infrastructure monitoring using wireless sensor networks," in *Wireless Telecommunications Symposium, 2007. WTS 2007*, april 2007, pp. 1 –7.

[37] S. K. Korkua and W.-J. Lee, "Wireless sensor network for performance monitoring of electrical machine," in *North American Power Symposium (NAPS)*, Oct. 2009, pp. 1–5.

[38] B. Lu and V. Gungor, "Online and remote motor energy monitoring and fault diagnostics using wireless sensor networks," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 11, pp. 4651–4659, Nov. 2009.

[39] K. Tan, S. Huang, Y. Zhang, and T. Lee, "Distributed fault detection in industrial system based on sensor wireless network," *Computer Standards & Interfaces*, vol. 31, no. 3, pp. 573–578, Mar. 2009.

[40] M. Paavola and R.-P. N. andKauko Leiviska, "A wireless approach to water supply pump diagnosis," in *The Sixth International Conference on Condition Monitoring and Machinery Failure Prevention Technologies*, vol. 2009, Dublin, 2009, pp. 991–1002.

[41] Y. Kaicheng and Z. Chaosheng, "Vibration data fusion algorithm of auxiliaries in power plants based on wireless sensor networks," in *2011 International Conference on Computer Science and Service System (CSSS)*, Jun. 2011, pp. 935–938.

[42] V. Gungor, B. Lu, and G. Hancke, "Opportunities and challenges of wireless sensor networks in Smart Grid," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 10, pp. 3557–3564, Oct. 2010.

[43] G. Hackmann, F. Sun, N. Castaneda, C. Lu, and S. Dyke, "A holistic approach to decentralized structural damage localization using wireless sensor networks," *Computer Communications*, vol. 36, no. 1, pp. 29 – 41, 2012.

[44] L. Hou and N. Bergmann, "Novel industrial wireless sensor networks for machine condition monitoring and fault diagnosis," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 10, pp. 2787–2798, Oct. 2012.

[45] L. Hou and N. Bergmann, "Induction motor fault diagnosis using industrial wireless sensor networks and Dempster-Shafer classifier fusion," in *IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society*, Nov. 2011, pp. 2992–2997.

[46] (2016, 8) CC1310 SimpleLink Ultralow Power Sub-1-GHz Wireless MCU. Texas Instruments. [Online]. Available: http://www.ti.com/lit/ds/symlink/cc1310.pdf

[47] (2016, 8) IRIS Wireless Measurement System. MEMSIC Inc. [Online]. Available: http://www.memsic.com/userfiles/files/Datasheets/WSN/IRIS_Datasheet.pdf

[48] F. van der Heijden, *Classification, Parameter Estimation and State Estimation: An Engineering Approach Using MATLAB.* Wiley, 2004.

[49] A. Jain, R. Duin, and J. Mao, "Statistical pattern recognition: a review," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 1, pp. 4 –37, jan 2000.

[50] D. M. J. Tax, "One-classclassfication: Concept-learning in the absence of counter-examples," Ph.D. dissertation, Delft University of Technology, Netherlands, 2001, classification.

[51] V. Vapnik, *Statistical Learning Theory.* Wiley, 1998.

[52] M. Lebold, “Review of vibration analysis methods for gearbox diagnostics and prognostics,” in *Proceedings of the 54th Meeting of the Society for Machinery Failure Prevention Technology*, 2000.

[53] N. Tandon and A. Choudhury, “A review of vibration and acoustic measurement methods for the detection of defects in rolling element bearings,” *Tribology International*, vol. 32, no. 8, pp. 469 – 480, 1999.

[54] J. Zakrajsek, D. Townsend, and H. Decker, “An analysis of gear fault detection methods as applied to pitting fatigue failure data,” in *47th Meeting of the Mechanical Failure Prevention Group*, 1993.

[55] S. G. Braun, Ed., *Encyclopedia of Vibration.* Elsevier, 2001.

[56] S. Med, “Diagnostics of JMP-TS-20 in laboratory conditions K-316,” mathesis, University of defence, 1999.

[57] J. Neuzil, O. Kreibich, and R. Smid, “A distributed fault detection system based on IWSN for machine condition monitoring,” *Industrial Informatics, IEEE Transactions on*, vol. 10, no. 2, pp. 1118–1123, May 2014. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6661382

[58] O. Kreibich, J. Neuzil, and R. Smid, “Quality-based multiple sensor fusion in an industrial wireless sensor network for MCM,” *IEEE Transactions on Industrial Electronics*, vol. 61, no. 9, pp. 4903–4911, 9 2014. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6678819