Czech Technical University in Prague

Faculty of Electrical Engineering

DOCTORAL THESIS

Ondřej Kreibich

Wireless diagnostic methods in an aerospace application

Department of Measurement

Supervisor of the doctoral thesis:	doc. Ing. Radislav Šmíd, PhD.
Study programme:	Electrical Engineering and Information Technology
Specialization:	Air Traffic Control

February 2014

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 parents and my brother for their love and support throughout my life.

A very special acknowledgement goes to my girlfriend Ester, who loved and supported me during the final, critical months of my dissertation.

I would like to give my sincerely thanks to my supervisor, Associate Prof. Radislav Smid, PhD., for his guidance and support throughout my studies, and especially for his confidence in me. His comments and questions have helped me greatly in completing the manuscript. I am grateful to him for discussing and interpreting some of the results presented in this thesis.

Finally, my thanks to all my friends who have supported me or influenced me along the way.

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: Metody bezdrátové diagnostiky v leteckých aplikacích

Autor: Ing. Ondřej Kreibich

Katedra: Katedra měření, ČVUT v Praze, FEL

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

Abstrakt: Strategie plánované údržby je založena na průběžném sledování stavu zařízení se systémem včasného varování před nestandardními stavy v chování sledovaného zařízení. Využití bezdrátové technologie WSN v této oblasti by přineslo řadu výhod vycházejících z konstrukce senzorového uzlu sítě, ke kterému nevede žádný přívod. Takové řešení usnadní montáž senzorů na těžko dostupná místa, ale zároveň otevírá možnosti pro zcela nové aplikace, například měření na pohyblivých částech zařízení. Aby takový systém mohl být nasazen do průmyslové praxe, je potřeba zaručit spolehlivý přenos informace mezi senzorovými uzly a bránou napojenou na nadřazený kontrolér, či počítač. Tato práce navrhuje systém pro sledování stavu strojů založený na technologii WSN schopný překonat falešné indikace způsobené dočasnou ztrátou dat, rušením signálu nebo přenosem neplatných dat a to za použití multi – senzorové datové fúze rozhodující se dle parametru kvality, zasílaném senzorovým uzlem spolu s daty. Tento ukazatel je založen na stavu senzorového uzlu (napájení, síla signálu) a porovnání aktuální hodnoty s hodnotami v předchozích datových záznamech. Algoritmus datové fuze rovněž tento indikátor poskytuje. Tento nový přístup umožňuje šíření informace o nejistotě měřené hodnoty ze zdrojového uzlu až k bráně a zároveň vyřazuje neplatná data na uzlech datové fuze. Tím možnost degradace posílané diagnostické informace značně klesá. Přenos rychlých signálů je zajištěn extrakcí a přenosem příznaků ze surových dat, tím dochází k úspoře přenosové šířky pásma sítě. Koncept byl experimentálně ověřen nejen matematickou simulací, ale i na reálném WSN hardware (Imote2). Předpokládaná efektivita systému byla vyhodnocena pomocí poměru signál / šum (SNR) a vlastním detektorem četnosti výskytu chyb (FAR). Výsledky potvrzují, že se navrhovaný přístup vyrovná drátovému propojení senzoru s měřicí ústřednou. A proto lze takový systém aplikovat i na kritická zařízení, jako jsou pohonné jednotky ultralehkých letadel, kde se systém včasné kontroly závad doposud nevyužívá.

Klíčová slova: Bezdrátové senzorové sítě (WSN), Sledování stavu strojů (MCM), Technická diagnostika v letectví, Datová fůze

Title: Wireless diagnostic methods in an aerospace application

Author: Ing. Ondřej Kreibich

Department: Department of Measurement

Supervisor: doc. Ing. Radislav Šmíd, Ph.D., Department of Measurement

Abstract: The early alert monitoring system for an effective scheduled maintenance strategy based on a wireless technology requires reliable transfer of diagnostic information between the sensor and the gateway. This thesis presents a WSN-based machine condition monitoring (MCM) system capable of overcoming a false indication caused by temporary loss of data, signal interference or invalid data. We use multi-sensor fusion driven by a quality parameter, produced by each sensor node according to the data history outliers and the actual state of the node. The fusion node also provides a quality evaluation on its output. This novel approach enables the propagation of information about the uncertainty of a measured value from the source node to the sink node. Thus potential degradation of acquired or transferred diagnostic information is minimized. Instead of raw data the signal features are transferred, so that bandwidth savings are improved considerably. The proposed concept was experimentally verified on real WSN hardware. The performance evaluated using the Signal-to-Noise ratio and false alarm rate detection demonstrates the effectiveness of the proposed approach. The results confirm that the proposed system has similar reliability to a sensor connected by wire to a central unit. The machine condition monitoring system based on WSN with multi-sensor fusion is able to monitor a critical application, and even to monitor light aircraft powerplants.

Keywords: Multisensor Information Fusion, Machine Condition Monitoring, Industrial Wireless Sensor Networks, Condition-based Monitoring in Aviation

List of Figures

List of Tables

Acronyms

AIDS Aircraft Integrated Data System

BITE Built-in Test Equipment

CAT Commercial Air Transport

CBM Condition Based Maintenance

CFDS Centralized Fault Display System

CM Condition Monitoring

CMS Central Maintenance System

EASA European Aviation Safety Agency

ECAM Electronic Centralised Aircraft Monitor

EVMU Engine Vibration Monitoring Unit

FADEC Full Authority Digital Engine Control

GA General Aviation

HP High Pressure

ICAO International Civil Aviation Organization

IWSN Industrial Wireless Sensor Network

LP Low Pressure

MCDU Multifunctional Control and Display Unit

MCM Machine Condition Monitoring

MSG-3 Maintenance Steering Group-3

MTOM Maximum Take Off Mass

ODMs Oil Debris Monitoring system

OSI Open Systems Interconnection

SDAC Systems Data Acquisition Concentrator

SGU Symbol Generator Unit

VLA Very Light Aircraft

VLJ Very Light “personal” Jets

WSN Wireless Sensor Network

ZDO ZigBee device object

Contents

Chapter 1

Introduction

One spring morning in Canada, the pilot and his passenger flew to Toronto to have some radio work done on the aircraft. Once the work was completed, the pilot and two aircraft maintenance engineers carried out engine runs as a final check. Everything appeared to be functioning properly. The pilot and passenger then boarded the aircraft and prepared to return home.

At approximately 12:25 the Cirrus SR20 light aircraft was cleared for takeoff from the runway. Shortly after takeoff the pilot reported a problem and decided to return to the airport. As the aircraft target did not appear on the radar, it is estimated that the aircraft did not reach a height of more than 500 feet. Light grey smoke emanated from the aircraft as it started a shallow climbing left turn. The airport tower controller attempted to contact the aircraft, without success. The tower controller cleared the aircraft to land on any runway.

The aircraft's bank angle increased and its nose dropped suddenly. The aircraft descended quickly and entered a spin. Just before striking the rooftop of a building, the wings levelled and the nose came up. A post-crash fire broke out shortly after impact and consumed most of the aircraft. Both occupants were fatally injured.

This sad story is not fictional, but is one record from many accidents of light aircraft all over the world. It is borrowed from an aviation investigation report for the Transportation Safety Board of Canada [1].

Since the propeller blades were bent back, the damage was consistent with low power or no power being produced by the engine at the time of impact. An examination of the engine showed that one cylinder head had separated from the barrel (see Fig. 1.1). The cylinder head remained held in place by the induction and exhaust systems. No other abnormalities were found with the engine that would have prevented it from producing power.

The investigation discovered that the pilot was certified and qualified for the flight in accordance with existing regulations. Furthermore, the passenger was also a licensed private pilot. The weather at the time was suitable for visual flight and records show that the aircraft was certified, equipped, and maintained in accordance with existing regulations and approved procedures.

Pilots during training have to deal with simulated engine failures and have to be able to cope with a such situation. However, this fatal accident, where two experienced pilots failed to carry out emergency landing procedures properly, provides evidence that unexpected loss of power is a very stressful situation leading to incorrect pilot judgments.

A system able to provide a warning before a serious aircraft power plant failure can help pilots significantly in such a situation. The general idea of this thesis is to introduce such a system based on wireless technology for easy application in

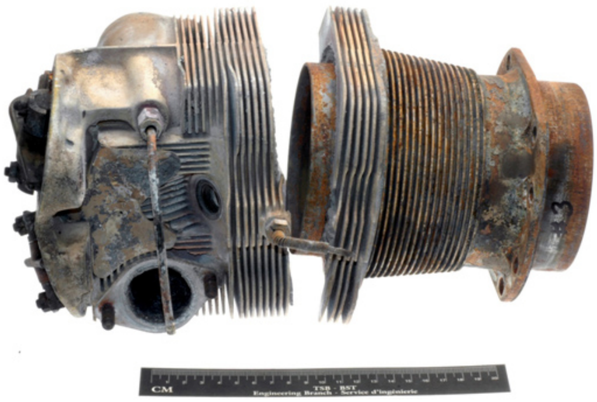

Figure 1.1: Cylinder head separated from the barrel [1]

currently used and newly-produced light aircraft. Assuming that an aircraft power plant is a mechanical rotating device in general, we are able to provide early detection of such a failure (even tens of seconds can prevent a disaster). By machine condition monitoring (MCM) techniques (e.g. FFT spectrum analysis) we are able to detect essential faults of a rotating device, e.g. imbalance, misalignment, bearing faults, and mechanical looseness, which is what probably led to the fatal accident reported on here.

In the literature, there are a few articles that discuss wireless ideas for aircraft onboard engine monitoring systems (e.g. [2]) but, as far as the author knows, no robust MCM system utilizing the strong benefits of Wireless Sensor Network technology has been presented.

1.1 Aviation safety performance

Commercial air transport is one of the safest forms of travel. This has been proved by data collected and published by national, union and international aviation safety authorities like ICAO (International Civil Aviation Organization) and EASA (European Aviation Safety Agency).

If EASA records based on passenger fatalities per 100 million miles flown are considered, it took two decades (1948 to 1968) to achieve the first 10-fold improvement from 5 to 0.5. Another 10-fold improvement was attained in 1997, almost 30 years later, when the rate had dropped below 0.05. In 2009, this rate was 0.01 fatalities per 100 million miles flown, with one not fully clarified accident of an Airbus 330 over the Atlantic [3]. Furthermore, 2010 was the first year in which no fatal accident in commercial air transport operations occurred in Europe [4]. By comparison, in the same time period there were 0.05 fatalities per 100 million miles in railway passenger traffic, 0.46 in sea transport and 1.75 in road transport in the countries of the European Union [5]. Some data using kilometers have been converted into nautical miles, as used in air traffic, for readers' convenience.

In addition to Commercial Air Transport (CAT) operations, there are other categories of aviation, referred to as General Aviation (GA), Aerial Work and Light Aircraft. These terms do not have the same meaning throughout the world, or even within countries. Simply stated, these categories refer to all aviation activity except that performed by major airlines and the armed forces. Another approach refers to these aviation categories according to a particular Maximum Take Off Mass (MTOM) threshold. It is evident that a range of different aircraft and operations fall into these categories, as well as a range of functions such as agriculture, construction, photography, surveying, observation, aerial advertising, trade, recreation, sport and personal flying.

EASA annual safety reviews offer a detailed look at all categories of aircraft. CAT operation shows satisfying safety results, as was indicated above (3 fatal accidents in 2011). GA and Aerial Work have slightly worse statistics (6 fatal accidents in 2010 and 12 fatal accidents in 2011). However, light aircraft (MTOM below 2250 kg) with the very rapidly growing subcategory of microlights (also called ultralights) used for recreational, personal and sports flying do not show such positive safety records as for larger aircraft. EASA data records 129 fatal accidents in 2010 and 169 fatal accidents in 2011 in the EU countries. Unfortunately, there is no unified definition of the light aircraft category across countries or unions, and in addition a lot of accidents in countries where the authorities are less diligent go unregistered. The number of light aircraft accidents per year throughout the world is much worse (more than 1000 per year). The EU statistics for 2011 show interestingly that alongside pilot error almost 40% of all fatal accidents were caused or influenced by a system or component failure.

1.2 Maintenance scheme in current aviation

One of the factors affecting the progress in commercial aircraft safety, alongside improved aircraft design, engineering, the evolution of navigation aids and avionics, has been the development of maintenance schemes.

Commercial aircraft maintenance schemes are nowadays based on on-board fault diagnostic routines and on-ground periodic inspections that have to be carried out on all commercial aircraft after a certain period of time or amount of usage (i.e. A - D Checks) [6]. The manufacturer's initial maintenance schedule is a part of the aircraft documentation developed within work towards aircraft certification. Modern commercial aircraft have inspection intervals defined according to aviation's Maintenance Steering Group-3 (MSG-3). This process is based on in-service operational data, such as mean time between failures, evaluated by specialists in the various aircraft systems in interaction with members of the manufacturer's design group 2.2.

On-board systems, like BITE (Built in Test Equipment) on Airbus airplanes, generate reports with messages about abrupt changes in measured values or actual failures of an aircraft system [7, 8]. These reports warn pilots about technical problems that have already occured (minor, major or transient problems) which could require an immediate return to the gate. Modern aircraft are equipped with a Central Maintenance System (CMS), which also captures parameters based on predefined trigger conditions for long-term analysis of trends in aircraft systems and the flight crew. Oil consumption monitoring of the engine is an example of this [9]. Besides monitoring state and performance values, there are two sources of predictive maintenance data. The first is the quality of the engine oil or, more precisely, its contamination by metal splinters, and the second is the overall engine vibration value (see chapter 2.2.1). Both of these values very exactly reflect the condition of the engine, and their trends provide information about impending failure or its development.

In contrast to the highly-developed maintenance scheme and the strict inspection processes for CAT operation aircraft, light aircraft - defined by safety statistics as having MTOM below 2250 kg - mostly do not use any condition-based approach to maintenance. These aircraft are equipped with factory-produced (or adapted automotive) two-stroke or four-stroke engines, according to CS-VLA regulations for the EU countries. Aircraft in this group have a poor safety record, as mentioned above. This thesis deals mainly with aircraft in the below MTOM 2250 kg category. Generally, we will refer to the target group as *light aircraft*.

1.3 Introduction of an early warning system for light aircraft

The increasing number of light and very light aircraft for personal use (i.e. sightseeing, sports activities, or private flights) has led to an increased number of accidents. Hence the general idea of this thesis is to introduce a low-cost early warning system into newly-produced and also currently operating light aircraft.

It is hard to predict whether an early warning engine monitoring system could have prevented the fatal accident reported above. Unfortunately very light aircraft (VLA) and some light aircraft (LA) are not equipped with any on-board fault detection capability. As a consequence, pilots are not able to detect abnormal power plant behavior in the early stages. In most cases, by the time that widely-used engine indicators (i.e. RPM, engine head temperature, or oil pressure) show performance degradation, a fault has already developed. However, mechanical losses like those that occurred in this case can be detected early. In frequency spectrum analysis, a widely-used method in machine condition monitoring (MCM) of rotating devices - internal assembly looseness - can be detected by the presence of many harmonics in the FFT spectrum due to the non-linear response of the loose parts to the forces from the rotor. Structural looseness increases the frequency bin at $1 \times$ RPM in the radial direction measurement (see Fig.1.2). FFT indications of mechanical responses are more precisely presented in the relevant literature (e.g. [10]). In the case of a complex mechanical device like a combustion engine, structural looseness (e.g. cracks, frame distortion) can lead to other faults (e.g. leakage). For example, Chandroth et al. [11] carried out tests on a diesel engine with artificial faults (a leaking exhaust valve, a leaking air valve, etc.). These faults were also detectable in the FFT spectrum (by frequency bins around 10 kHz).

1.3.1 Early warning MCM system based on a Wireless Sensor Network

Electric harnesses and connectors are usually much more expensive to install and maintain than the cost of the sensors themselves [12]. Merely comfortable installation and maintenance without increasing the overall weight and complexity of the avionics could be beneficial in this area. A wireless approach is the only solution that can meet this requirement. Wireless sensors could be freely mounted on moving, rotating parts and in many types of environments, including hazardous areas [13].

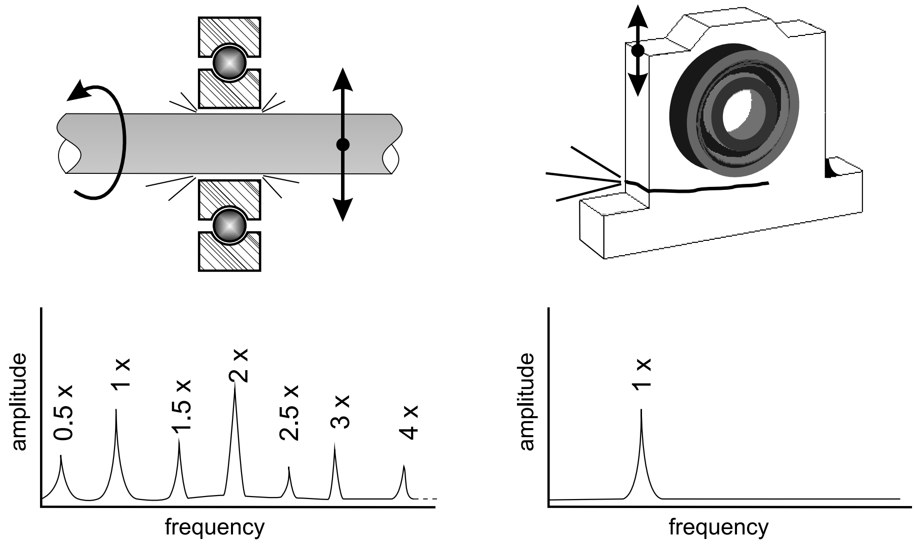

Figure 1.2: FFT indication examples of mechanical looseness; left, internal assembly looseness; right, figure structural looseness

Technology with sensing, data processing and communication capabilities is referred to as a Wireless Sensor Network (WSN) [14]. WSNs are characterized by flexibility, self-organization, self-configuration, inherent intelligent-processing capability, and the ability to be deployed rapidly [15]. The sensor nodes employ miniaturized hardware design, miscellaneous sets of energy-efficient communication protocols [16, 17], various communication technologies [18], suitable power sources and energy management [19]. However, sensor nodes have constrained hardware resources due to power supply from batteries or from energy harvesting systems. This has a significant effect on transfer rate and computing complexity.

Precise real-time monitoring or high-performance data analysis for distinguishing among sources of mechanical faults does not of course mean that a constrained wireless technology can be introduced, or that wires can be eliminated completely. However, WSN technology provides a convenient platform for an early warning in the event of complications. For this reason, several papers have referred to systems of this kind as "killer applications" [20].

The appropriate monitoring system for an effective scheduled maintenance strategy remains a goal not only for the aircraft industry but also for many manufacturing companies. A wireless approach can also bring the same benefits to this area, and almost all statements in this thesis are also applicable to other manufacturing industries. In the last two decades, the adoption of network communication technologies in factory monitoring and automation systems has developed tremendously [21]. Much research is being done nowadays on wireless solutions within a factory and also in the process domain [22]. Recent research papers refer to wireless sensors and sensor networks for deployment in industry as Industrial Wireless Sensor Networks (IWSNs) [23, 24].

A few recent papers have gone on to express a need for WSN for engine health monitoring applied to all aircraft categories [2]. In any case, legislation, certification and regulation lie beyond the scope of this thesis.

How to read the thesis

The reader can quickly become familiar with the goals and the main results of this thesis by first reading Aims of the doctoral thesis (chapter 3), then Multi-sensor fusion (chapter 5), and finally the summary in the Conclusions (chapter 8, section 8.1). This should enable the reader to comprehend the relations between the chapters, and to put the details of each chapter into context.

Chapter 2

State of the art

2.1 General concept of condition monitoring

Condition Monitoring (CM) can be defined as a technique or a process for monitoring the operating characteristics of machines in such a way that changes and trends of the monitored characteristics can be used to predict the need for maintenance before serious deterioration or breakdown occurs, and/or to estimate the "health" of the machine. It embraces the life mechanism of individual parts, or of the whole equipment, the application and development of special purpose equipment, the means of acquiring the data and the analysis of that data to predict trends [25]. The term CM is frequently extended to MCM - *Machine Condition Monitoring* - in the area of machinery. Other fields also use the term *fault detection* to characterize the same sense as CM or MCM [26].

CM links directly with *Condition Based Maintenance* (CBM), a technique for diagnosing failure mechanisms and making a prognosis of the remaining useful life before failure. This enables corrective maintenance action to be undertaken on the identified failing component(s) at a convenient time before the anticipated time of failure [27]. CBM significantly improves safety and mainly reduces maintenance costs by increasing the availability (uptime) of plant machinery and equipment.

Previously, *time-based maintenance* had for a long time been the most widely used maintenance strategy. Machines were examined and crucial parts were changed according to a time schedule or running hours. This type of approach is also called *preventive maintenance*. Although the actual condition of the machine was unknown, time-based maintenance prevented many failures. Nevertheless, unexpected accidents did not fully disappear and time and money were wasted on unnecessary shutdowns. Non-critical supporting devices could be maintained after a failure, i.e. *corrective maintenance*. Both time-based maintenance and corrective maintenance schemes still remain in existence. Basically, if no reasonably consistent age interval exists between time of potential failure and time of functional failure, on-line monitoring is inapplicable. Secondly, there is concern about the initial cost of CBM. There is in fact the same commercial demand for CBM systems that are reliable but competitively-priced products. CBM components provide not only monitoring but also an optimized mix of maintenance technologies, supervising, management, and logistics technologies.

2.1.1 CBM, Supervision and Fault Management

Recently published literature (e.g. [28]) also uses the terms *fault management* and *fault supervision*. These terms cover the same actions as CBM, i.e. condition monitoring (fault detection), fault diagnosis and actions to avoid a failure or a malfunction. Besides being general-purpose terms for other industrial areas (home appliances, automotive systems etc.) the difference (if any) lies in a greater emphasis on the management of related actions if a fault occurs. This mainly means:

- maintenance / service procedure generation,
- reconfiguration / self-maintenance,
- improvement of reliability through changes in design or quality control,
- compensation of faults in a way that does not lead to breakdown of the system.

The last-mentioned approach is known as *fault tolerance*. This means that *redundancy* is implemented. Redundant schemes can be designed for hardware, software, information processing, and mechanical or electrical components. Crucial systems and devices for aircraft, spacecraft and nuclear or chemical plants use multiple redundancy, usually a triplex structure where two failures are tolerated and a third failure allows the operator to control the system manually.

Many systems need to be certified according to reliability and safety demands (RAMS). The design of a product should remove operational faults and prevent future recurrences. A range of fault analysis methods have been developed, such as: reliability analysis, event tree analysis (ETA), fault tree analysis (FTA), failure mode and effect analysis (FMEA), hazard analysis (HA) and risk classification.

Although some of these theories continue to be applied, many of them lie beyond the scope of this thesis. Detailed descriptions can be found in the cited literature, e.g. [28].

2.1.2 CM architecture

It has been mentioned that there is no unique terminology within CM/CBM techniques due to their broad applications and the advanced knowledge that is drawn from many scientific fields. New scientific discoveries are still being made in this area. However, one contemporary set of international standards has helped to build up a complete CM system. These standards are issued as ISO 13374 - 1,2,3. These standards support an open system architecture for condition monitoring and diagnostics of machines (e.g. MIMOSA OSA-EAI Architecture) remotely resembling the OSI model in communications. Machine condition assessment ISO 13374-1 [29] is broken down into six distinct, layered processing blocks (see Fig. 2.1). The first three blocks are technology-specific, requiring signal processing and data analysis functions targeted at a particular technology. They are as follows:

- Data Acquisition (DA) block: converts an output from the transducer to a digital parameter representing a physical quantity and related information (such as time, calibration, data quality, data collector utilized, sensor configuration)
- Data Manipulation (DM) block: performs signal analysis, computes meaningful descriptors, and derives virtual sensor readings from raw measurements.

- State Detection (SD) block: facilitates the creation and maintenance of normal baseline "profiles", searches for abnormalities whenever new data are acquired, and determines in which abnormality zone, if any, the data belong (e.g. "alert" or "alarm").

The final three blocks normally attempt to combine monitoring technologies in order to assess the current health of the machine, predict future failures, and provide recommended action steps for operations and maintenance personnel. The three blocks and the functions that they should support are as follows.

- Health Assessment (HA) block: diagnoses any faults and rates the current health of the equipment or process, considering all state information.

- Prognostic Assessment (PA) block: determines future health states and failure modes based on the current health assessment and projected usage loads on the equipment and/or process, as well as remaining useful life predictions.

- Advisory Generation (AG) block: provides actionable information regarding maintenance or operational changes required to optimize the life of the process and/or equipment.

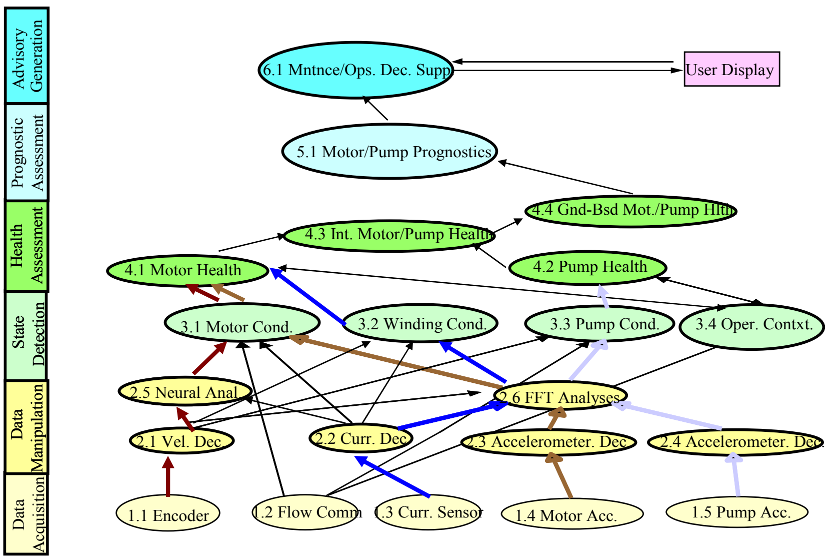

Figure 2.1: OSA CBM Compliant Modular System [30]

ISO 13374 standards are based on the idea of breaking the CM system down into generalized components or functions. This enables distributed data processing within a CM architecture (see Fig. 2.2). The functionality of one or more data processing blocks from the processing architecture can be implemented into a module interacting with other modules to form a complete integrated system. Open systems architecture design enables the integration of improved prognostic capability within new or existing system designs, allowing maximum flexibility for future upgrades to the system. A distributed computing system can have more computing power than a mainframe. This significantly reduces processing time and improves system performance.

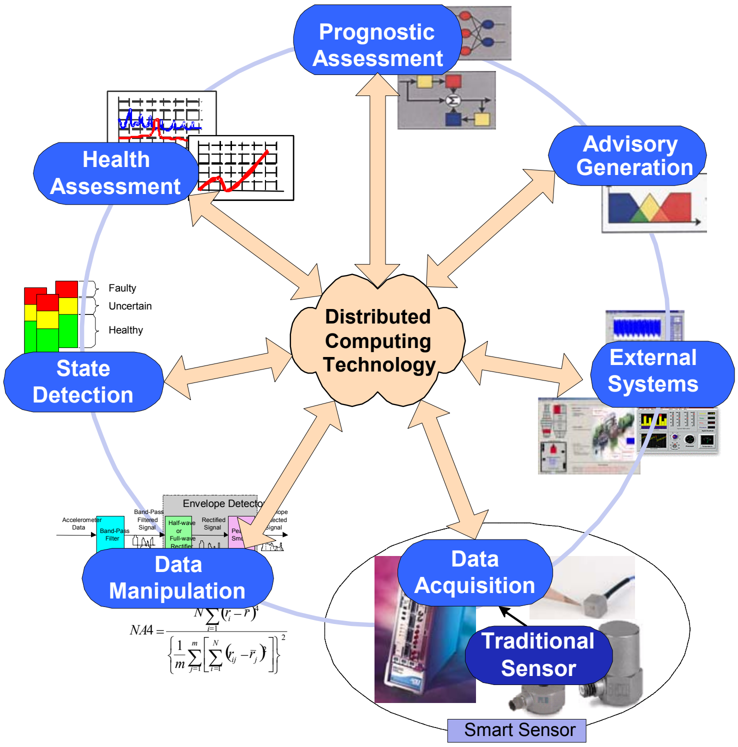

Figure 2.2: Distributed data processing within CM standard architecture [30]

In the last two decades, the adoption of network communication technologies in factory monitoring and automation systems has developed tremendously [21]. Since open system architecture is not dependent on a communication medium between the modules, communication can be accomplished using popular network and middleware technologies. The modules therefore do not need to reside on the same machine but may reside anywhere on a local or worldwide network. Much research is being done nowadays on wireless solutions within a factory and also in the process domain [22].

2.2 Condition monitoring of aircraft engines

On December 29th, 1978 F. Stanley Nowlan and Howard F. Heap published report number A066-579, "Reliability-Centered Maintenance" [31]. The report was the culmination of several years of work aimed at determining a new, more cost-effective way of maintaining complex systems. This report first published failure curves in addition to the well-known 'bath-tub' curve. With the introduction in the 1960s of the first Boeing 747, questions were raised about the sense of continuing with traditional maintenance based on the traditional preventive maintenance paradigm. Nowlan et al. demonstrated that only about 11% of aircraft equipment failures were age related, and the other 89% were totally time-random events. Most equipment failures have no relationship to length of time in-service. Most failures are unpredictable. But if a potential failure is detected early, it is possible to plan and do the repair cost-effectively before it becomes a breakdown.

Many new studies [32] improved on the results of the Nowlan report, which established findings for many fields of industry, not only for aerospace. The report mainly defined reasons for using monitoring based on the actual condition of a device.

Although our thesis is primarily focused on wireless diagnostic methods in aerospace applications, almost all ideas and conclusions in this thesis are also applicable to devices far removed from aerospace applications.

2.2.1 Turbofan and turboprop engines

Present-day commercial turbofan engines are designed to accomplish flight-line inspection, maintenance and borescoping with minimum disruption of other components. The design also has the necessary flexibility for extensive on-wing maintenance, using a condition monitoring maintenance plan mainly based on the level of contamination of the lubricating oil by ferro-magnetic and non-ferromagnetic particles, and on the overall level of engine vibration during operation [8, 33].

Mechanical components wear, erode and corrode, producing chips and depositing various particles into the oil flow. Thus the condition of oil which has been circulating in the lubrication system for some time provides a very exactly reflection of the condition of the system. These particles become larger as bearing or gear damage develops, and are collected by the oil filter and by magnetic chip detectors installed in the scavenge oil flow. Basic magnetic chip detectors are controlled manually, while sophisticated variants are monitored electronically. The FADEC computer monitors the resistance between two installed magnets (or screens for conductive, non-ferromagnetic materials). Modern engines are equipped with oil debris monitor systems (ODMs). These sensors are based on the eddy current principle, and work independently of oil temperature, flow rate, viscosity, air and water content or oil color (darkening). A more precise tool for monitoring oil quality is spectrographic oil analysis, which is performed in a laboratory. There are two accepted methods for performing oil analyses: atomic absorption and atomic emissions. Either atomic absorption or atomic emission will identify the presence of submicroscopic materials that are suspended in the engine oil. An oil analysis identifies the material suspended in the sample, and also the quantity of that material in parts per million (PPM).

Engine vibration indicators provide diagnostic information for use in deciding on engine maintenance actions. According to the Airbus A310 CF6 engine manual [33], the engine vibration measurement comprises: two transducers (piezoelectric accelerometers), a signal conditioner and two vibration indications, N1 and N2, on the ECAM display unit. The primary accelerometer is bolted to the No. 1 bearing housing (flange A) in 6:00 o'clock position. It uses a 50 pC/g sensitivity pickup. This position can be equipped with the alternate accelerometer, which provides a vibration level signal source in the event of primary accelerometer failure. The N2 Core accelerometer is tuned to sense vibrations in the core rotor assembly. It is installed on the forward flange in 10:00 o'clock position. The scheme is almost the same for many other modern commercial turbofan engines, across all aircraft manufacturers.

The signal conditioner is an electronic unit located in the avionics compartment. It receives an analog signal from the vibration accelerometers and tachometer sensors (rotational speed of the LP and HP rotor). Output signals are transmitted to the SDAC, which transforms these signals into digital signals and transmits them by busbar to the ECAM SGU and AIDS. Latest Airbus airplanes are equipped with an EVMU instead of the signal conditioner. This unit has a similar function that of the signal conditioner. The EVMU receives analog signals from the sensors mounted at the engine (speed and vibration) and communicates directly with the other computers, e.g. CFDS, ECAM and AIDS. These computers, together with other systems, constitute the engine vibration sensing system, and enable the crew to monitor engine vibration on the ECAM system. The engine or cruise page shows the vibration level of N1 and N2 as a numerical indication, where the scale is from 0 to 10 units (it is approximately equal to 0 - 150 mm/sec). When vibration indication reaches 3 for N1 or 5 for N2, the engine page is displayed automatically and the indicator flashes green on the R ECAM display unit. The engine vibration sensing system also provides maintenance staff with data such as vibration indication, excess vibration, fan imbalance position and amplitude, imbalance corrective weight calculation, storage of balancing data, speed recording, BITE and MCDU communication with other systems, accelerometer selection and frequency analysis for component vibration search.

Frequency analysis of the acceleration signal is performed on the ground. The EVMU does the analysis at a selected N1 or N2 speed and uses any valid accelerometer. The maximum frequency analysis is 500 Hz and the frequency increment between adjacent spectral lines is 4 Hz.

Engine oil is monitored for metal splinters introduced by degradation of moving parts:

- bearings,
- gears,
- valves,
- contact between moving parts and their housing

Rising vibration level addresses the following issues:

- rotor/stator contact
- rotor imbalance
- blade defects
- bearing and gear defects

2.2.2 Piston engines

Present-day light piston aircraft are not equipped with any system able to prevent a failure. The latest piston engines for light aircraft are diesel water-cooled and turbocharged engines with a single-lever digital engine management system (FADEC), e.g. Centurion 2.0 or SMA SR305-230 used in small Cessna, Diamond and many other airplanes. Present-day engines combust mainly aviation gasoline (avgas) or automotive gasoline (mogas). Factory manufactured aircraft piston engines are heavily modified automotive engines or specially designed units for this purpose only. Many subsystems used in these engines mirror similar advances in automotive technology [34]. In the case of very light aircraft there is a similar situation. Factory manufactured aircraft use a serially produced engine (Rotax, Zenoah etc.) Home-built aircraft are also equipped with serially produced engines, or self-builders adapt automotive two-stroke or four-stroke engines according to CS-VLA regulations (in the EU countries).

Maintenance of aircraft is constrained by minimum cost and minimum downtime. In the category of light aircraft, the inspection and maintenance work load is usually divided into smaller operations that can be accomplished in short time periods that satisfy the complete airplane inspection requirements of both 100-hour inspections and annual inspections. All maintenance work and associated work must be accomplished by an appropriately approved organisation. Microlights are maintained by pilots or by their owners according to the aircraft or main part manufacturer maintenance schedule, if the aeroplane is operated for private purposes only.

2.2.3 Wireless CM

The main benefit of a wireless approach, as opposed to a wired system, is that there is unrestrained sensor placement, installation and maintenance. A wireless sensor can be freely mounted on moving, rotating parts and in many types of environments, including hazardous areas. In addition, a wireless sensor is easier to install in new or already-functioning machinery equipment.

CBM provides an optimal maintenance service if the CM system provides correct and useful information about the actual condition of the machine. This is one reason why industrial CM systems remain in centralized wired fashion, even though new technologies could in most cases provide more options and improvements at lower cost.

Currently, the most widely-used CM system is based on vibration analysis, and uses a central computing unit directly or via a data acquisition module connected to sensors (an accelerometer) as input and to a display unit or an OPC server via an industrial bus or a computer network [27]. A typical system of this kind is an Amot bearing condition monitor (see Fig. 2.3). The Signal Processing Unit (SPU) can simultaneously process signals from sensors. The Interface Unit houses the necessary circuitry to distribute a DC power supply to the engine-mounted SPU and CPU. It also provides the terminations for integrating the with additional diagnostics. The Central Processing Unit (CPU) provides data storage, full class reporting and a local user interface.

Some producers of CM systems already offer a wireless monitoring solution where the wire connection is replaced by wireless technology. Two basic approaches are used. The first approach uses broadband wireless technology (e.g. Wi-Fi) to replace signal cables from sensors to a central processing unit. The power supply remains wired due to the high energy consumption of broadband wireless sensors (Wi-Fi uses up to 100 mW transmitting power). This enables high-speed data transfer from sensors to a freely mounted central unit. The benefit is a constantly uncluttered workspace even if many sensors are engaged (see Fig. 2.4). The second approach uses energy-efficient wireless technology for a self-powered wireless sensor. This type of system has to be composed from low-energy consumption hardware, an optimized software code and adapted signal processing methods. For example, the sensors

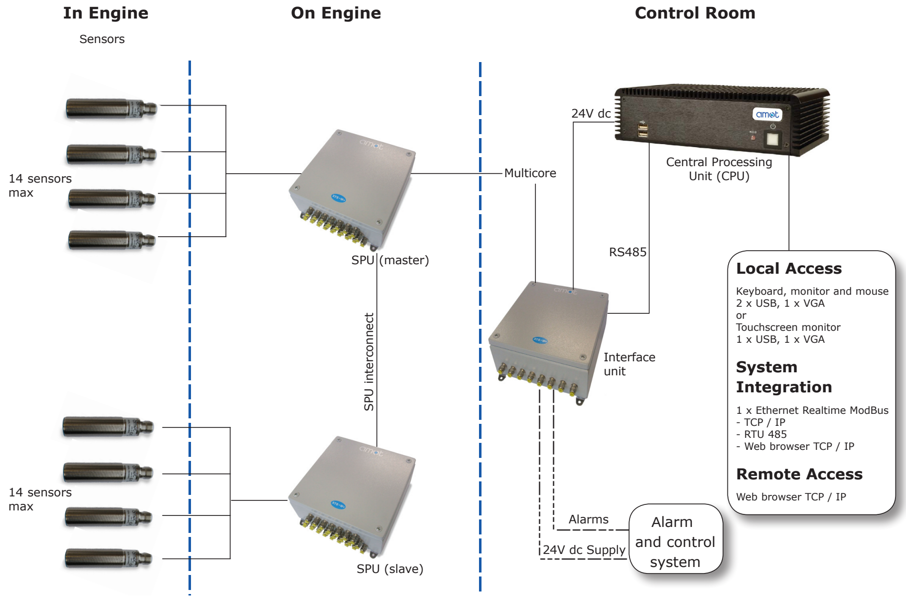

Figure 2.3: An example of the most widely-used centralized CM system [30]

acquire signals in time slots or react to specific excitation (some event occurs or the threshold of a measured value is exceeded), otherwise they are in sleep mode to save energy. The sensors are then wire-free stand-alone units. A typical example of a commercial product of this kind is the Status Check, produced by TIMKEN company.

Main features of the SKF WMx system:

- IEEE 802.11b/g standard
- Spectrum/time domain data
- External wake-up
- Up to 40 kHz bandwidth
- Up to 12 800 lines resolution
- 10 to 30 V DC power

Main features of the TIMKEN Status Check system:

- 2.4 GHz ISM band
- Transmission range: one kilometer line-of-sight; 300 meters in plant
- Data-collection interval: adjustable (five user-selectable rates)
- Battery life up to 4.5 years, depending on the data-collection interval

Recent research papers refer to wireless sensors and their networks for industrial deployment as Industrial Wireless Sensor Networks (IWSNs) [23, 24]. IWSN ideas result from an effort to introduce and adapt already known wireless sensor network (WSN) technology into industrial applications. The centralised star topology that

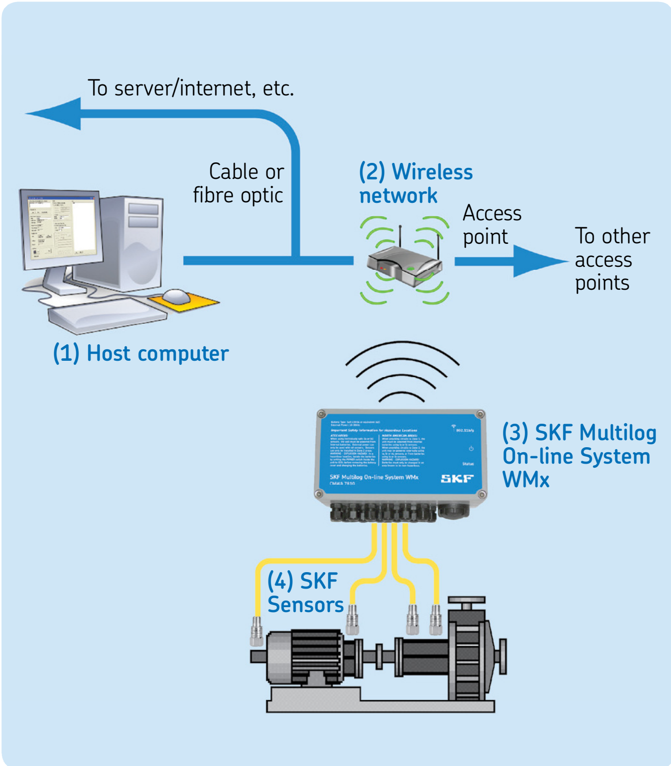

Figure 2.4: An example of the SKF broadband centralized wireless CM system [35]

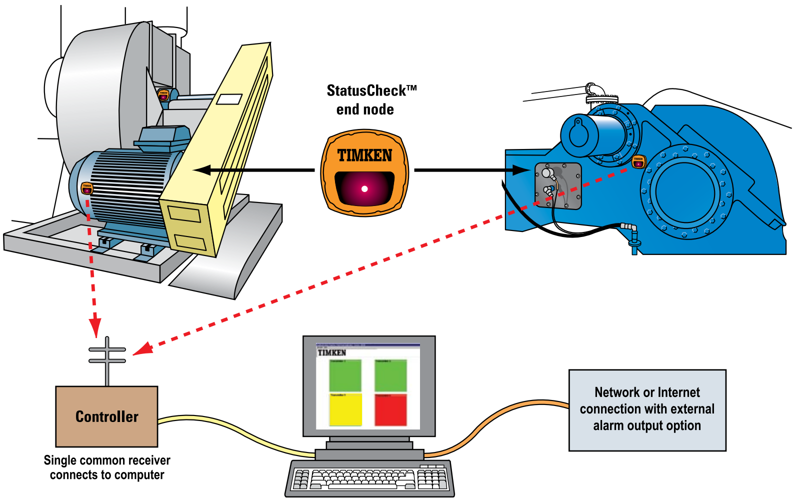

Figure 2.5: An example of the TIMKEN low-energy consumption centralized wireless CM system [36]

many systems currently use does not utilize the sophisticated networking available under a WSN system. The WSN concept is based on a large mesh network between nodes equipped with a sensor to ensure a reliable measurement task over a wide area. The sensor nodes employ a miniaturized hardware design, miscellaneous sets of energy-efficient communication protocols [16, 17] and communication technologies [18], suitable power sources and energy management [37]. The sensor node is able to carry out only simple computing tasks due to restrained resources, but if signal and data processing distribution among nodes is engaged, the network achieves robustness and sophisticated functionality.

Various already published subsystems can be well adopted for CM. However, propagation of maximum information content picked up from a device by sensors to a sink node, while the system complexity and the energy consumption of each node remain reasonable, remains a major challenge for WSN application in the CM area. To the best of the author's knowledge, several studies for high-sampling IWSN systems have been presented in the literature, but no commercial product is available, not even in another application than CM.

A recent paper applying IWSN in condition monitoring [38] describes a similar requirement to find efficient processing methods to maximize the information content of the transferred signals while minimizing the energy consumption for each WSN node. The authors introduce the use of WSN for condition monitoring, where a high sampling rate is required due to the complexity of vibration signals. They also touch on the constrained hardware resources of WSN embedded systems. A mathematical method needs to be selected in compliance with the limited memory size, the CPU clock, and the precision of the data types. Additionally, the authors mention several previous systems based on various fusion methods, and finally they propose an IWSN system based on decision fusion (Dempster-Shafer fusion of neural network classifiers) focused on an induction motor.

2.3 Wireless Sensor Network technology

The particular design of a WSN depends significantly on the application. Nowadays, it is not complicated to configure a WSN system according to specific application requirements, since there exist multiple variations of hardware design, software equipment, communication standards, routing algorithms and other WSN improvements. However, the fundamental concept of WSNs is defined as a small unit called a node, which senses its surroundings and communicates with neighbouring nodes. This node transmits acquired data (or transmits already received data) to another node to form a network. In this fashion, measured values are transferred via a network of nodes to a sink node. The sink node passes the gathered data to the central computing unit, where final data processing is done and an adequate action is performed. The connection between the sink node and the computing unit can be provided by a common bus (e.g. USB) or by some other network technology, e.g. Ethernet. The central computing unit can also be enhanced by an adequate radio module and forms the sink node itself. Recently published papers referring to WSN propose distributed data processing inside the network [39]. In addition, a multi-agent approach may be implemented [40]. This results in a very robust network with artificial intelligence attributes, which can adapt to sudden unexpected conditions.

The term Wireless Sensor Networks became known at the turn of the 21st century. Today it is hardly possible to identify the first idea dealing with wireless network of sensors. Horst Stormer (Nobel Prize, Physics, 1998) published a popular article in Business Week magazine in August 1999 in which he presented WSN as one of the "21 ideas for the 21st century". The earliest studies date back to the 1950s, when the Sound Surveillance System (SOSUS) was developed by the United States military to detect and track Soviet submarines. The United States Defense Advanced Research Projects Agency's (DARPA) Distributed Sensor Network (DSN) program began in 1980 formally to explore the challenges in implementing distributed wireless sensor networks. Subsequently civilian scientific research was initiated by an academic partnership between Carnegie Mellon University and the Massachusetts Institute of Technology.

Nowadays there are many commercial products based on the expansion of WSN research in the last decade. Present-day products are focused on light industry and on consumer electronics. Typical applications are HVAC and light control in smart buildings, street lighting, air quality monitoring, structural monitoring, forest fire detection and natural disaster prevention. Common characteristics of these systems are the small size of the node, the self-powered layout (long-life battery, energy harvesting system), and the communication and networking protocols according to a widespread standard (IEEE 802.15.4, ZigBee).

However, this concept is significantly affected by energy limitation. Energy harvesting systems do not yet produce enough power for a continuous power supply, and the battery capacity is strongly limited by the size of the node. Energy drawbacks result in restrained resources, such as computing power and the transmitting power of the node. Present-day self-powered products are therefore used for simple measurement tasks where a slowly changing physical phenomenon is monitored. The phenomenon is usually temperature, light intensity, mechanical force, pressure, motion, humidity etc.

While there is still enough market demand for these WSNs, moving beyond the limitations is a challenge.

2.3.1 Sensor node

The sensor node (or mote) is an embedded system composed of a simple microcontroller, a sensor or a batch of sensors, an RF transceiver and a power source, as depicted in Fig. 2.6. The hardware design is considerably adapted to the low-power nature of the nodes.

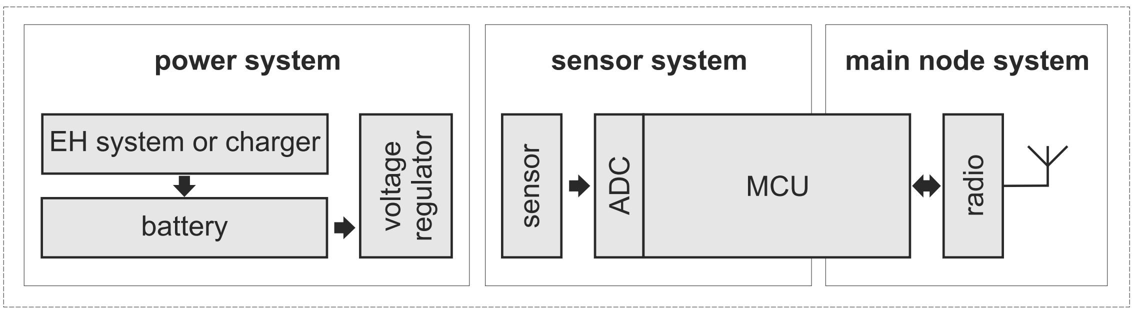

Figure 2.6: Fundamental structure of a sensor node

One of the first finished products was the Mica (see Fig. 2.7), developed by the wireless sensor research group at the University of California, Berkeley. This platform was manufactured by Crossbow, which has introduced many well-known WSN products during the last decade, from the 8-bit Atmega 128 based Mica to powerful 32-bit Imote2 nodes based on the Intel PXA27X processor family. Unfortunately, Crossbow terminated the WSN program and no longer provides any support for their products. A similar trend occurred at another WSN pioneer team, at ETH Zurich, and to its product BTnodes. During the main WSN research boom in the first decade of the 21st century, more than twenty WSN platforms were introduced. Besides Mica, Mica2, Imote and BTnodes, the popular WSNs included eyesIFX, Iris, SHIMMER, Telos, TMote and others. The Computer Engineering group at ETH Zurich elaborated an interesting overview of the history of WSN platforms on their web under the title The Sensor Network Museum [41]. Many present-day WSN products are derived from "museum" platforms with reduced design size and utilizing the latest ultra low-power electronic components.

Figure 2.7: The Mica experimental platform by Crossbow [42]

- Mica purpose: experimental platform,
- a 128-KiB flash program memory,
- a 4-KiB static RAM,
- an internal 8-channel 10-bit ADC,
- the standby current is a few μA,
- 1.25×2.25 inch,

Present-day sensor node products can be categorized according to their application area. There are low-power nodes (e.g. commercial IQRF, programmable XBee and academic PowWow) based on popular low power MCU, e.g. PIC16 from Microchip and MSP430 from TI. These products are intended for continuous operation or for operation with short interruptions. At the other end of the spectrum are microelectronics producers (e.g. Linear Technology, ST microelectronics) offering 32-bit MCU, usually as fully integrated System-on-Chip (SoC). Nodes based on these chips have enough computing power for demanding signal processing tasks. However, energy consumption increases with computing power. A typical operation is deep sleep mode for a long time with a short high performance computing task in the event of excitation.

Such chips almost entirely integrate the node with all radio circuitry components, microprocessor, Flash and RAM memory, and peripherals (amplifiers, ADCs, DACs). They require only a simple antenna connector and some isolated components (e.g. crystals, decoupling capacitors). For example, Silicon Labs' EM35x Ember® ZigBee® series includes a 32-bit ARM® Cortex™ - M3 processor, a fully-compliant IEEE 802.15.4 - 2003 transceiver, an AES encryption accelerator, an ADC and sub-μA sleep mode currents – all in a single 7 mm x 7 mm QFN package (see Fig. 2.8). Developing a sensor node in a WSN with this type of highly integrated solution requires little more than adding a battery and a sensor to the EM35x SoC mounted on an inexpensive circuit board [43].

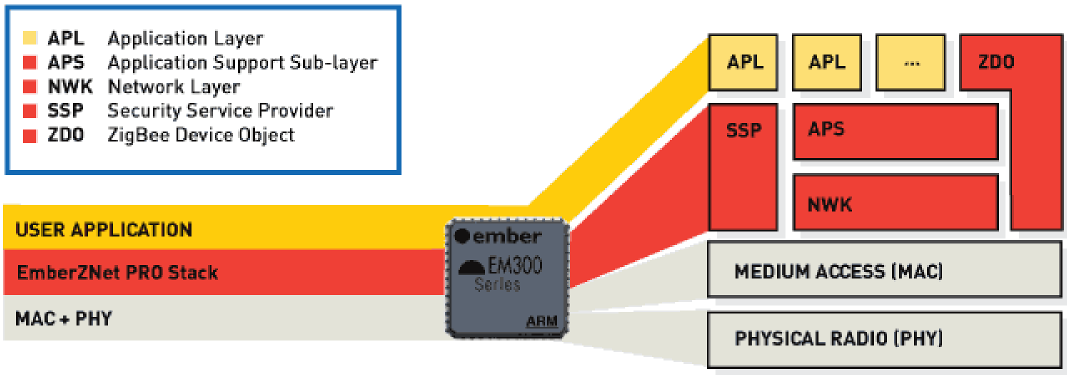

Figure 2.8: EM35x WSN System-on-Chip by Silicon Labs [44]

Recent research in microelectronics and MEMS (Micro Electro Mechanical Systems) technologies [45] has led to wireless sensor nodes based on one ultra-low power chip integrating all node components and sensor circuitry. These nodes will be millimeters in size or even smaller, and they will be equipped with an energy harvesting system [46]. Later, it will be possible to construct fully autonomous sensors, which will work reliably for their whole planned life cycle. They enhance many industrial and health care applications where it is not possible to maintain them (e.g. replace the battery). An unattended sensor node will be integrated into the structure of a component, or a tissue. For example, connecting rods, crank shafts, drive shafts and other crucial mechanical components will be able to monitor their own mechanical parameters and measure their rotating/moving speed, torque, and other quantities that are effective for controlling a complex device. One example could be a connecting rod wirelessly monitored while moving in an engine, see Fig. 2.9. In the same way, building structures in civil engineering or living organisms can be monitored by wireless sensors placed inside them.

Instead of a fully integrated sensor node, many applications use their own processor main board with a separated radio module. The radio module fully arranges communication within a network (transmitting, receiving, routing). Data between a module and MCU is passed through UART or SPI. Frequently-used brand modules are Xbee by Digitech and the CC2500 RF module by TI. Interesting low-cost solutions, mainly based on TI or Nordic semiconductor chips, are now produced in China (e.g. Nrf24L01, WM2500LP7).

Figure 2.9: IRTs connecting rod wireless measurement technology [47]

2.3.2 Operating systems

A custom system developed for a single customer or for some specific organization can be based on a standalone program that fulfills a customer's particular preferences and expectations. However, the system software of the embedded devices is generally based on boot software and a preferred operating system (OS). WSN low-power embedded platforms are designed either as an open-source platform or a closed-source platform. A proprietary system often includes a patented design and a closed OS with a task-specific program code or flash memory dedicated to a user code. An open-source platform enables the user to adapt either system software or hardware design, or both, on the basis of a general public license that applies predominantly non-commercial products. The following OS are used within WSN, the first two being the most popular:

- **TinyOS**: This is an event-based operating system in which individual components act together to form a complete application and to implement all system functions. This component-based structure lets an application designer select from a catalog of system components to meet application-specific goals. Each component acts as a finite state machine that uses commands and events to transition from one state to the next. There is no blocking or waiting in TinyOS. This forces components, after finishing a calculation, to release the CPU for use by other components. TinyOS performs high-level, long-running application processing in special execution contexts, called tasks. When executed, a task runs to completion and other tasks cannot preempt it. However, low-level system events can preempt tasks, allowing TinyOS to temporarily reallocate the CPU to low-level system processing. TinyOS shields the application-level processing from the underlying concurrent scheduling, yet exposes low level system components to meet their real-time requirements [48].

  Programming language: nesC - event-driven extension to the C programming language used to build applications for TinyOS.

Pros: TinyOS is an open source software platform and tool-chain developed by UC Berkeley and actively supported by a large community of users. Numerous prebuilt sensor applications include multi-hop ad-hoc routing, an interactive Matlab interface for control, as well as Java and WWW based monitoring. C-fluent programmers can write custom applications, specifically designed for WSN.

Cons: Programming for a specific issue is strongly affected by the existence of a previous solution or example; it is almost infeasible to port unsupported HW; new versions depend on the work of a limited group of developers; updates and fixes come at intervals of a few years.

- **Contiki**: This is also an open source operating system, which was developed at the Swedish Institute of Computer Science. Contiki provides a full IP network stack, with standard IP protocols such as UDP, TCP and HTTP, in addition to new low-power standards like 6lowpan, RPL, and CoAP. To save memory while providing a nice control flow in the code, Contiki uses a mechanism called protothreads. Protothreads is a mixture of event-driven and multi-threaded programming mechanisms. With protothreads, event-handlers can be made to block, waiting for events to occur. Contiki is designed to run in small amounts of memory. A typical system with full IPv6 networking with sleepy routers and RPL routing needs less than 10 KiB RAM and 30 KiB ROM.

  Programming language: C

  Pros: Contiki runs on a range of different hardware platforms and is designed to be easy to port to new hardware. Contiki is currently developed by a world-wide community of hardcore experts.

  Cons: Due to the multi-thread concurrency model, locking can be problematic, and larger memory is required than in the event-based model.

- **Nano-RK**: This is a real-time operating system (RTOS) with multi-hop networking support from Carnegie Mellon University designed to run on micro-controllers for use in sensor networks. Nano-RK currently runs on the FireFly Sensor Networking Platform as well as on MicaZ motes. It includes a light-weight embedded resource kernel (RK) with rich functionality and timing support using less than 2 KiB of RAM and 18 KiB of ROM. Nano-RK supports fixed-priority preemptive multitasking to ensure that task deadlines are met, along with support for CPU, networking, as well as sensor and actuator reservations. Tasks can specify their resource demands, and the operating system provides timely, guaranteed and controlled access to CPU cycles and network packets. Together, these resources form virtual energy reservations, allowing the OS to enforce system and task level energy budgets [49].

  Programming language: Nano-CL - imperative-style language provides an abstraction from the lower-level details of the sensor networking OS and radio communication. The program is composed of two important sections: the job descriptor and the service descriptor. The user writes a service, which is functionally equivalent to a task that is to be executed on each node.

Pros: Nano-RK is an open software platform like TinyOS, and was developed within a university project. Dynamic memory management scheduling.

Cons: The system has a long period between updates, and it is adapted for two HW platforms only.

- **Lightweight Linux distribution**: There are some open projects e.g. Linux-WSN, Linux-Zigbee and Linux-802.15.4, which want to bring the Linux operating system into the WSN domain. This could bring many benefits into WSN development, but present-day installations are mainly the work of Linux enthusiasts, and there is no complete solution in existence.

  Programming language: C

  Pros: The well-known Linux environment

  Cons: Available for advanced WSN nodes only; insufficient support

- **.NET microframework**: This is a popular runtime environment for embedded applications for small, connected, embedded devices with Visual Studio and C#. This means that programmers can now use the same development tools and language that they use to build desktop and smart device (PDA and smartphone) applications to develop applications for microcontrollers. The .NET Micro Framework also provides an extensible hardware emulator for rapid prototyping and debugging. In 2007, Crossbow released the Imote2 .NET microframework edition to reduce the complexity of the sensor development environment and to increase greatly the productivity of developers, resulting in the proliferation of robust, powerful wireless sensor applications. Unfortunately, this concept has been been implemented very carelessly. Many important functions of the PXA271 microcontroller are unavailable in the application layer (libraries are unfinished or are completely missing).

  Programming language: C#

  Pros: Reduced complexity of program development

  Cons: .NET environment is not optimized for WSN

- **SOS**: The Simple Operating System is an OS for mote-class WSN (Atmel-amega, Oki Arm) developed by UCLA. SOS uses a common kernel that implements messaging, dynamic memory, module loading and unloading, and other services. SOS uses dynamically loaded software modules to create a system supporting dynamic addition, modification and removal of network services [50].

  Programming language: C

  SOS is no longer under active development

- **MANTIS**: The system was developed at CU Boulder as an open source, multi-threaded operating system. Automatic preemptive time slicing for rapid prototyping. It supports platforms including MICA2, MICAz, and TELOS motes. An energy-efficient scheduler is included for duty-cycle sleeping of the sensor node. Small footprint (less than 512B RAM, 14KiB flash)

  Programming language: C

  MANTIS is no longer under active development

- **BTnut**: This is an OS for BTnodes, used for multi-hop networking. It uses a L2CAP connection-less mode.

  Programming language: C

  BTnodes and BTnut are also no longer under active development

- **OpenTag**: OpenTag is a DASH7 Mode 2 software stack, written in C, which is designed to be run on microcontrollers (or radio SoCs). Therefore, OpenTag must be a very tight, compact package, but with a proper configuration [51].

2.3.3 Networking topologies

Wireless sensor networking generally uses topologies already widely utilized in wire/wireless application before the term WSN was itself introduced. In addition to unsuitable bus or ring arrangements, almost all other topologies are used for routing data packets between nodes. The most used types are: point-to-point, star and mesh network (see Fig. 2.10). However, WSN is mainly associated with the mesh topology, i.e. a network with no centralized control or high-power transmitter/receiver able to reach all of the networked devices. A mesh network is very reliable due to redundancy, where nodes are connected with more than one neighbour node. This technology comes from the military area, and it is now used for creating completely new applications where traditional topologies come up short. The failure of a unique transmission line linking any peripheral node to the central node will result in the isolation of that peripheral node from all others. However, this is not the case in mesh networks. Let us imagine a typical present-day WSN application - a street lighting application to control the functionality of a lamp. A star topology is unable to provide whole street coverage. Although a linear topology may appear to be a simple solution for this issue, it is inadequate, due to easy interruption of the signal between two neighbouring nodes caused by tree/building shadows or interference. A mesh topology ensures a robust connection to all locations in the street (see Fig. 2.11).

An overview of the main topologies, and their typical applications:

- **Point-to-point**: used in applications such as tire pressure monitoring systems, garage door openers and television remote controls.

- **Star**: simple home automation or security systems, one corrupted node or link does not affect the rest of the network. The performance of the network is dependent on the central hub.

- **Mesh**: the network is able to re-route according to current conditions, and thus pass a barrier (e.g. outdoor summer (leaves) / winter (snow) change).

- **Hybrid**: applications with very large area deployment.

2.3.4 WSN communication standards

Wireless sensor technology as a general concept has ben supported by many research projects and developments performed in the last decade. Producers of electronic and measurement devices have established alliances around the most widely-used technologies (e.g. ZigBee, EnOcean, and Dust Network). Few technologies have yet

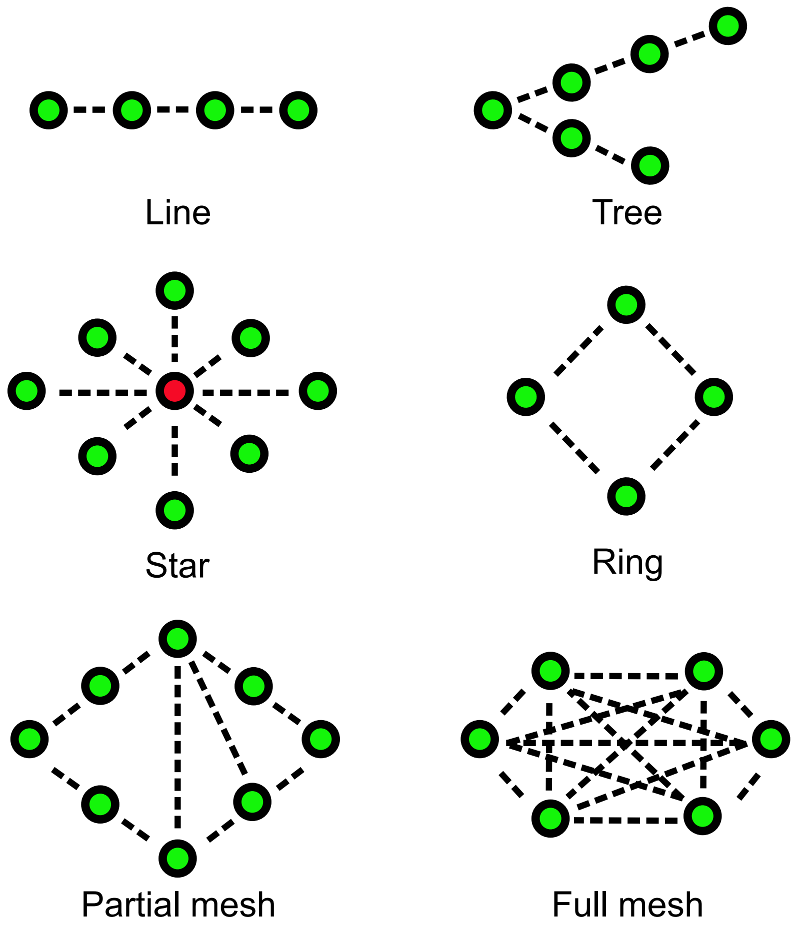

Figure 2.10: WSN network topologies

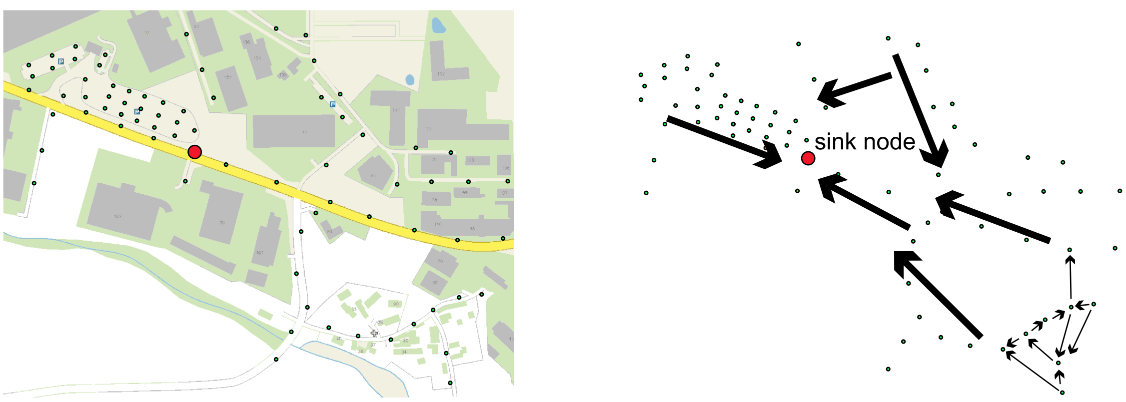

Figure 2.11: Street lighting monitoring based on WSN

become international standards. The most important specification maintained by the IEEE Standards Committee is IEEE 802.15.4. It specifies the physical layer and media access control for low-rate wireless personal area networks. This standard focuses on low-cost, low-speed communication between devices. In contrast with other, more end user-oriented approaches, such as IEEE 802.11, the emphasis is on very low-cost communication of nearby devices with little or no underlying infrastructure, intending to exploit this to reduce power consumption even more. This standard forms the basis for high-level communication protocols such as ZigBee, WirelessHART and the MiWi specification, which further attempts to offer a complete networking solution by developing the upper OSI layers, which are not covered by the international standards.

In contrast to high data rate wireless technologies such as Wi-Fi and Bluetooth, WSN has to communicate with their neighbor nodes with less than a milliwatt transmission power. This drastic reduction in transmission power must be accomplished by a highly efficient protocol processing mechanism. In addition, the topologies have to allow hundreds and potentially thousands of nodes on a single network, much more than the number of devices permitted by Bluetooth or Wi-Fi. An overview of the main WSN standards is as follows:

International WSN standards:

- **IEEE 802.15.4.**: This protocol specifies the physical layer and media access control for low-rate wireless personal area networks (LR-WPANs). The standard was approved by the Institute of Electrical and Electronics Engineers (IEEE).

- **IEC 62591**: This protocol utilizes a time-synchronized, self-organizing and self-healing mesh architecture known as WirelessHART. The standard was approved by the International Electrotechnical Commission.

- **ISA100.11a**: This standard is designed to support a wide range of wireless industrial plant needs. The standard is based on open standards such as IPv6 and UDP. ISA100.11a defines the protocol stack, system management and security functions for use over low-power, low-rate wireless networks. The standard was approved by the International Society of Automation as well as by IEC under number 62734.

- **IEC 14543-3-10**: This standard is geared to wireless sensors and wireless sensor networks with ultra-low power consumption. It includes sensor networks that utilize energy harvesting technology to draw energy from their surroundings – for example from motion, light or temperature differences. The specification is based on the Wireless Short-Packet (WSP) protocol optimized for energy harvesting introduced by EnOcean company, which develops energy harvesting wireless technology, as a pioneer in this field.

WSN standards within alliances and research groups:

- **ZigBee**: based on the IEEE 802.15 standard. The ZigBee network layer supports both star and tree typical networks, and generic mesh networks. The network is based on a coordinator device, which is tasked with creating it, controlling its parameters and basic maintenance. The specification adds to IEEE 802.15.4 a physical layer with four main components: a network layer, an application layer, ZigBee device objects (ZDOs) and manufacturer-defined application objects which allow for customization and favor total integration. Besides adding two high-level network layers to the underlying structure, the most significant improvement is the introduction of ZDOs. These are responsible for a number of tasks, which include keeping the device roles, management of requests to join a network, device discovery and security [52, 53].

- **ANT, ANT+**: a practical wireless sensor network protocol running in the 2.4 GHz ISM band. Designed for ultra-low power, ease of use, efficiency and scalability, ANT easily handles peer-to-peer, star, connected star, tree and fixed mesh topologies. ANT provides reliable data communications, flexible and adaptive network operation and cross-talk immunity. The ANT protocol stack is extremely compact, requiring minimal microcontroller resources, and it considerably reduces system costs [54]. A typical ANT protocol transceiver is a black box that comes pre-loaded with the protocol software and must be controlled by an application processor via a UART, SPI, or USB interface. The transceivers are embedded in equipment such as heart rate belts, watches, cycle power, and cadence meters. Recently, ANT alliance has attempted to diversify to home automation and industrial sectors [53].

- **Dust Network**: Dust Networks products are built on breakthrough Eterna^{TM} 802.15.4 SoC technology, delivering ultra low power consumption for wire-free operation on batteries or energy harvesting. Dust's portfolio of standards-based products includes: SmartMesh IP and SmartMesh WirelessHART products. SmartMesh IP is built for IP compatibility, and is based on 6LoWPAN and 802.15.4e standards. SmartMesh WirelessHART complies with the WirelessHART (IEC 62591) standard, offers the lowest power consumption in its class and is the most widely used WirelessHART product available [55].

- **MiWi**: wireless protocols designed by Microchip Technology, which uses small, low-power digital radios based on the IEEE 802.15.4 standard. The MiWi protocol stacks are small foot-print alternatives to ZigBee, which makes them useful for cost-sensitive applications with limited memory. The MiWi P2P protocol stack supports star and peer-to-peer wireless-network topologies, useful for simple, short-range, wireless node-to-node communication. There exists a unique restriction and obligation to use MiWi only with Microchip microcontrollers [56].

- **DASH7**: the DASH7 alliance provides short-range devices organized in Sub 1GHz based on ISO 18000-7 standards and ISO 18047-7 test methods.

- **6LoWPAN**: is the acronym for IPv6 over Low power Wireless Personal Area Networks. The 6LoWPAN concept proposes an adaptation layer to allow the transmission of IPv6 datagrams over IEEE 802.15.4 networks.

- **OCARI**: is the acronym for Optimization of Communication for the Ad hoc Reliable Industrial network, which is a communication protocol for the Industrial Wireless Sensor Network. It is distinguished from other protocols by: a deterministic access method to the RF medium supporting time-constrained packets relay (called MaCARI); a proactive energy-efficient routing strategy supporting nomadism (called EOLSR); an activity scheduling mechanism that is based on a three-hop coloring algorithm helps to reduce interference and thus optimizes the node's energy consumption (called SERENA) [20].

- **One-Net**: an open-source standard for wireless networking. One-Net currently operates in the 868 MHz and 915 MHz frequencies with 25 channels available for use in the United States. The ONE-NET standard allows for implementation on other frequencies, and some work is being done to implement it in the 433 MHz and 2.4 GHz frequency ranges. The specification allows per-node dynamic data rate configuration for data rates up to 230 kbit/s.

- **Z-Wave**: an interoperable wireless RF-based communications technology designed specifically for control, monitoring and status reading applications in residential and light commercial environments. It supports full mesh networks without the need for a coordinator node. It operates in the sub-1GHz band. Designed specifically for control and status apps, supports data rates of up to 100 kbit/s, with AES125 encryption, IPv6, and multi-channel operation. MAC and PHY are described by the ITU-T G.9959 specification.

- **Wibree**: an extension to Bluetooth introduced by Nokia. It extends the capabilities of the Bluetooth protocol to better enable low-power uses and to better process data intermittently, rather than continuously. Wibree has been included in the Bluetooth specification, as the Bluetooth low energy technology (BT 4.0).

- **OSIAN**: The Open Source IPv6 Automation Network is a free and open-source implementation of IPv6 networking for a wireless sensor network. OSIAN extends TinyOS and brings direct Internet connectivity to WSN technology.

- **PurePath**: a wireless technology (from TI) for transmitting uncompressed CD quality wireless audio over a state-of-the-art 2.4 GHz RF link. It uses a number of RF channels dynamically chosen for lossless audio transmission, resulting in minimal interference with other RF devices in the 2.4 GHz band. The technology enables audio applications with multichannel and multipoint audio streaming capabilities. It enables streaming up to four audio channels i.e. two speaker sets play different stereo audio streams. The PurePath specifications and purpose lie almost outside the definition of WSN, but this kind of product could also be used in high sampling measurements.

The enumeration declares a large number of communication concepts within WSN. Proper selection is based on application demands. Table 2.1 summarizes the range of standardized communication features.

2.3.5 WSN simulation tools

A simulation of the key characteristics of the computer network, in particular installation, provides significant time and money savings before actual implementation.

Table 2.1: WSN standardized communication features

Feature	Lower limit	Upper limit	Comment
Frequency band	sub-1 GHz (433 - 915 MHz)	2.4 GHz	↑ frequency then ↑ bit rate
Bit rate	1.2 kbit/s	250 kbit/s (IEEE802.15.4) up to 5000 kbit/s (PurePath)	↑ bit rate then ↑ consumption
Security	none	AES, 3DES, RC5, RC6, MISTY	↑ security then ↑ protocol stack
Protocol stack size	3 KiB (MiWi)	100 KiB (ZigBee)	-
Topology	point to point	mesh or cluster tree	-

Today, a broad variety of different simulation tools are used for simulating WSN performance. Some research papers [57, 58] have presented a comparison of existing simulators by implementing the same scenario. The results cannot be compared with each other, due to different simulation analysis capabilities and protocol support. One tool has a convincing analysis, another is based on an open source approach, while a third tool has a nice GUI. Selection of the simulation tool is mainly a question of functionality that is provided and user experience. There follows an overview of the most popular simulator and emulator WSN tools:

- **OPNET**: commercial network simulation software by Riverbed. The IEEE 802.15.4 / ZigBee model is under development, and a non-supported version is available. The source code is based on C/C++. The analysis of simulated data is supported by a variety of built-in functions. Various graphical presentations for the simulation results exist. Energy models are not directly supported by OPNET [57].

- **NetSim**: commercial network simulation software by Tetcos, available as seven components which users can choose and assemble. The source code is based on C/C++. Latest networking technology is supported, as well as WSN and PAN networks.

- **NS (network simulator)**: a discrete-event simulator written in C++, intended for networking research. It is free and open source, but it is not supported commercially. Development of the simulator is ongoing on the current NS-3. Nodes in NS-2 are considerably more sophisticated than the typical sensor node. Layers are included in the models that are not practical in a sensor node implementation, and the presence of these layers is likely to distort a simulation [57].

- **OMNeT**: a component-based, modular and open-architecture simulation environment with strong GUI support and an embeddable simulation kernel.

The modules are programmed in C++. A module for IEEE 802.15.4 has been developed based on the Mobility Framework, but is not yet available to the public. OMNeT++ is capable of running most TinyOS simulations by NesCT application [59], which converts TinyOS source to simulator compatible C++ code. Other frameworks useful for WSN are EYES and Castalia.

- **Prowler**: The probabilistic Wireless Network Simulator running under Matlab provides a generic simulation environment. Its target platform is the MICA mote running TinyOS.

- **GloMoSim, J-Sim, Sidh, SENS, and SENSE** are no longer supported simulators.

Emulator tools:

- **TosSim**: simulates the entire TinyOS applications. It works by replacing components with simulation implementations. TOSSIM is a library: you must write a program that configures a simulation and runs it. TOSSIM supports two programming interfaces: Python and C++. Unfortunately, the simulator is not compatible with the Imote2 platform. It is strictly dependent on TinyOS development.

- **AVRORA**: an open-source cycle-accurate simulator for embedded sensing programs. Avrora is written in Java and is released under the BSD license. It can emulate two typical platforms, Mica2 and MicaZ, and can run AVR elf-binary or assembly codes for both platforms.

- **ATEMU**: a software emulator for AVR processor based systems. Complete emulation of the AVR instruction set. Partial support for all MICA2 board components. Able to run TinyOS based code. No longer supported.

Chapter 3

Aims of the doctoral thesis

Sudden loss of aircraft power plant power is a very stressful situation in which pilots have a tendency to make an incorrect judgment leading to an accident. A warning signal or a message from an aircraft condition monitoring system can alert the pilot to oncoming failure in advance. A well-informed pilot can make a more confident decision [60]. The system has to detect a fault before it develops into a serious failure. In addition, comfortable installation and maintenance are required, without increasing the overall weight and complexity of the avionics. This type of situation is a major challenge for wireless sensor networks, where it is attractive to have a simple mounted sensor node capable of monitoring abnormal device behavior, communicating with neighboring sensors in order to process data, and transferring the results to an indicator on the instrument panel of an aircraft.

3.1 Specific aims of the doctoral thesis

The early alert monitoring system for an effective scheduled maintenance strategy based on wireless technology requires reliable transfer of diagnostic information between the sensor and the gateway. This thesis aims to improve WSN reliability to the level achieved with a wired connection, by means of:

- **WSN-based MCM system design**: The present-day WSN scheme used for monitoring purposes (presence of enemies, forest fires, etc.) is not feasible for signal monitoring using high sample rates. The aim is to propose a new scheme based on distributed signal processing methods, taking into account the nature of diagnostic signals.
- **WSN reliability improvement**: A redundancy-based fusion concept is capable of overcoming a false indication caused by temporary loss of data, signal interference or invalid data. This is especially true if multi-sensor fusion is driven by a quality parameter corresponding with sensor node imperfections (signal jamming, health of a sensor node - battery discharging).
- **WSN bandwidth savings**: A raw diagnostic signal represents a huge number of samples in addition to the redundancy concept, which increases the amount of data in the input section of the network. Instead of raw data, the signal features will be transferred and a compression method will be engaged.
- **Verification**: The proposed methods will be simulated in a high-level interpreter language (e.g. Matlab) to optimize the efficiency.

- **Performance evaluation**: The proposed multi-sensor fusion system will be implemented in real WSN hardware and performance tests will be carried out.

This novel approach will enable information about the uncertainty of a measured value to be propagated from the source node to the sink node. In this way, potential degradation of acquired or transferred diagnostic information will be minimized.

Chapter 4

Machine Condition Monitoring based on a Wireless Sensor Network

A wide-ranging wireless diagnostic system monitoring fast-changing signals (e.g. mechanical vibrations) based on WSN technology advantages is not feasible without engaging distributed signal processing methods, see section 2.3.

The key idea for introducing these systems is based on distributed signal processing methods, mainly information fusion, see Fig. 4.1.

- The entry level consists of sensor nodes equipped with a built-in sensor or ADC with an externally connected sensor. If more sensors of the same type are connected to a sensor node, raw data fusion can be performed to reduce the information produced by the sensor node.
- To reduce drawbacks caused by wireless data transfer or restricted power capabilities (battery discharging), the sensor nodes can be placed in redundant fashion. Several sensors sense the same phenomenon. Due to the feature fusion node, the correct information is transferred, even if one or more sensors are corrupted.
- Various phenomena can be classified, and partial results can be combined into a final condition evaluation of the monitored device.

This concept ensures efficient information transfer through a wireless sensor network, suppressing node resources overload and/or communication channel overload.

We focus especially on the segment between the sensor nodes and the feature fusion node. Established redundancy in a number of sensors suppresses the disadvantages of WSN systems (susceptibility to sensor degradation, unreliable RF links, etc.), see Fig. 4.5. The theoretical assumptions underlying the solutions of these tasks proceed from analyses in the following fields:

- characterization and classification of diagnostic signals on the basis of their source,
- diagnostic information transfer through WSN,
- reducing corrupted data by redundancy,
- signal information extraction - transfer of signal features instead of raw signal,

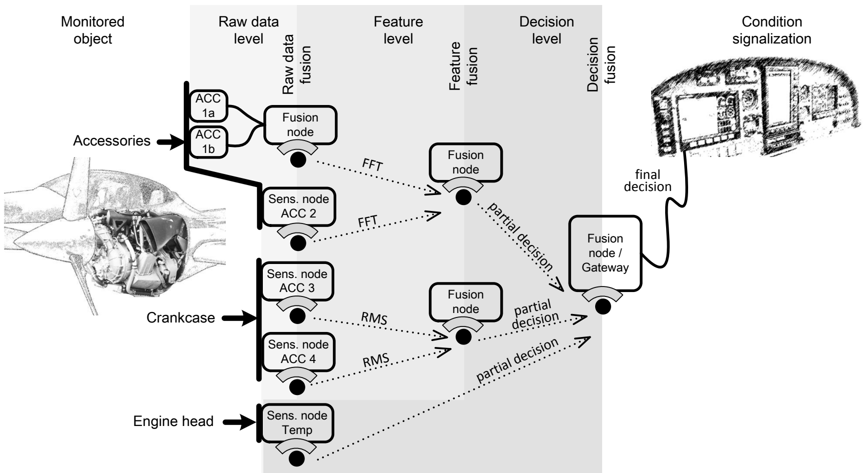

Figure 4.1: WSN monitoring system based on information fusion

- information fusion techniques to aggregate multiple signals from one object to one representation,
- a data quality indicator propagated together with data through WSN to drive the information fusion process.

4.1 Diagnostic signal classification

Data sets of diagnostic signals, e.g. vibration and temperature, acquired from various mechanical devices by conventional wire systems, were used as a point of reference for the WSN-based MCM fundamental ideas proposed in this thesis. The following general statement arose from the acquired data:

- Compact, rigid devices produce almost the same diagnostic signals in each particular direction of their body or cover. Typical examples are portable or auxiliary devices where a drive unit is included in a single body with the driven device, e.g. pumps and back-up electricity generators.
- A particular rigid component mounted on a solid basement of large machinery can produce diagnostic signals with a very similar waveform, in the same way as in the previous instance. Examples include solid bearing houses, motor housings and enclosed gearboxes.
- Large-size, low-toughness devices generate a different vibration waveform at almost every spot. The mechanical structure of devices of this kind does not allow an excitation to transfer across the structure.

Compact, solid devices can usually be measured by only one sensor per diagnostic quantity in a particular direction. If we increase the number of identical sensors, the accuracy and resistance to sensor degradation, defects and invalid measurements are improved significantly, due to sensor redundancy [61]. In addition, redundancy is advantageous for suppressing wireless technology shortcomings, e.g. short-term interference or a temporary node failure [62].

When each part of the machinery generates a different waveform of the measured signal, it is necessary to use at least one sensor per component, or even one sensor per axis of the component, to cover all indications of developing failure for such a complex device. Information from other domains can also be applied by using sensors of different types (e.g. temperature sensors, pressure sensors). All these relations among diagnostic signals, signal source and sensing method are summarized in Table 4.1.

Table 4.1: Classification of the type, source and sensing method of diagnostic signals

Diversity of diagnostic signals	**Signal source**	**Sensing method**
similar signals	• compact device • rigid device • one rigid component of a large device	• one sensor per quantity (simplicity) • more than one identical sensor per quantity (redundancy) • various types of sensors (complex information)
different signals	• large-size device • low-toughness device	• more than one identical sensor per quantity (accuracy – it covers more components and directions) • different types of sensors (complex information)

4.2 Diagnostic information transfer through a wireless MCM system

In communication network standards a maximum over-the-medium data rate is usually specified, but due to the overhead of the protocol that is used the actual network throughput is significantly lower. According to Latré et al. [63], the maximum useful IEEE 802.15.4 throughput is not more than 64.9% due to the overhead cost. This means a data rate of 163 kbit/s when no addresses and no acknowledgements are used. The maximum throughput drops further when acknowledgments and an address scheme are used.

4.2.1 Real time data transfer

The simplest diagnostic measurement setup can be realized by a single sensor node wirelessly connected to a gateway via point-to-point access, as was stated in section 2.3.3. Let us assume that we have 12-bit ADC. IEEE 802.15.4 maximum throughput 163 kbit/s provides transfer of uncompressed signal sampled by 13.5 kHz. This is enough for many MCM measurements.

In cases when more than one sensor node is to be used, a centralized star network topology provides an analogy to the conventional wire diagnostic measurement with a central DAQ module or an analyzer [64].

A wireless system based on one or a limited number of sensors connected in this manner is able to work in real-time mode, i.e. the response time is guaranteed [65]. This means that all possible latencies and time delays that occurred during data propagation through the WSN system have to be known and are less than a few milliseconds, because they determine the overall response time of the network.

If 16 bits of address and an acknowledgement are used, IEEE 802.15.4 throughput is 139 kbit/s [63]. If we reduce the ADC resolution to 8 bit and the sampling rate to 8 kHz, two vibration sensing MCM nodes can be connected to a gateway in real time mode without a data compression method. This performance is sufficient for MCM applications where a few signal sources are enough to cover the monitoring task.

4.2.2 Asynchronous data transfer

As was stated in the previous section, WSN technology is not sufficient for real time measurement of rapidly changing signals (e.g. for a control purpose). In addition, real time monitoring consumes energy significantly because of the need for continuous awake node operation. If widespread real time wireless monitoring is needed, a wireless networking technology ensuring high data transfer rate (e.g. Wi-Fi, Bluetooth) has to be utilized. Such network systems were designed to transfer high speed data (e.g. Internet) or transfer data between multimedia devices powered or re-charged from mains electricity. Industrially-used systems need a power line, an induction loop or powerful energy storage for long-life unattended power supply. However, nodes then lose full-independency or become increasingly bulky and too heavy to fall into WSN definition in section 2.3.

However, there are many tasks where accurate monitoring of operational parameters is not required. According to information theory, if we get information about a monitored phenomenon within the normal range of operation where the phenomenon is assumed (e.g. an experience with a device which is permanently used), it does not produce an information value contrary to a warning about an unexpected condition which is significantly more valuable. For example, a warning light on the car instrument panel informing the driver that a system is working properly (e.g. the alternator is charging the battery) is worthless, since the driver assumes that the system should be working properly. The warning light make sense only when it is lightened up when the charging system is corrupted.

Systems able to inform an operator about a faulty device condition, meaning that a part of the device is working improperly but the device is still operable, are called early alert (or warning) monitoring systems. WSN technology is widely used for this purpose. A simple measurement task triggered by an event (i.e. internal/external excitation, RF wake up, etc.), where the guaranteed response time can be relinquished. For example, a fire detection network [66] where a detector node sleeps in the periods between waking up according to an inner timer or an interruption from a thermocouple. An awake node sniffs the ambient temperature, sends data and finally goes back to the sleep mode.

A WSN node and distributed topology has the ability to preprocess data within a network. The fire sensor node from the last example can make a decision on whether the temperature lies within the norm or not. If the measured temperature is approaching the upper threshold, the node is woken up more often, or stays awake sniffing and sending data to a sink node via the network. If the temperature returns to a normal value the wake up period is extended again. We call this data transfer approach asynchronous, because of the variable period between the events. If the time interval between events is enough for sending all data acquired during the previous event, a high sampling rate is feasible.

Benefits of the asynchronous approach that we use in the proposed MCM WSN-based system:

- Each sensor node acquires a diagnostic signal only when an event occurs - communication channel and battery savings.
- The sensor node transmits the preprocessed condition information, not the acquired raw signal - network throughput savings.
- Data in the network are transferred in the order of the event or the processed partial result - precise synchronization is not required.

4.2.3 Extracting signal features

A significant advantage is assured by transmitting extracted signal features instead of raw data records. Signal features are considered as information sources in signal processing in a general way. In condition monitoring, the signal features are defined by international standards, e.g. ISO 10816 operates with the term vibration magnitude, which, depending on the type of machine, can be an RMS of vibration velocity, acceleration or displacement value. The frequency spectrum is used for locating mechanical wear and failures. The crest factor is used for estimating the amount of impact wear in a bearing. We have adopted all these signal features for our scheme scheme.

The novelty of our approach lies in placing feature extraction within the network. Unlike the wired system, which collects all data and performs feature extraction in a centralized unit (usually a PC), the WSN system is able to extract the features as close as possible to a sensor (within one node), see Fig. 4.2.

4.2.4 Data compression

When RMS or the crest factor is used as signal feature, the signal character in one time period is expressed by this indicator. Thus these methods significantly reduce the data themselves, and further compression based on information content is harder to achieve, apart from some lossy compression methods adopted from the IT field. However, the amplitude spectrum produced by FFT is not so reductive as the methods mentioned here. In this case, lossy data compression can be based on a threshold applied to the spectrum. Only significant amplitudes higher than a defined level are transferred to the network.

4.2.5 Redundancy

Generally speaking, wireless data transfer is more susceptible to signal distortion than an interconnection by wire. Present-day communication services providing a

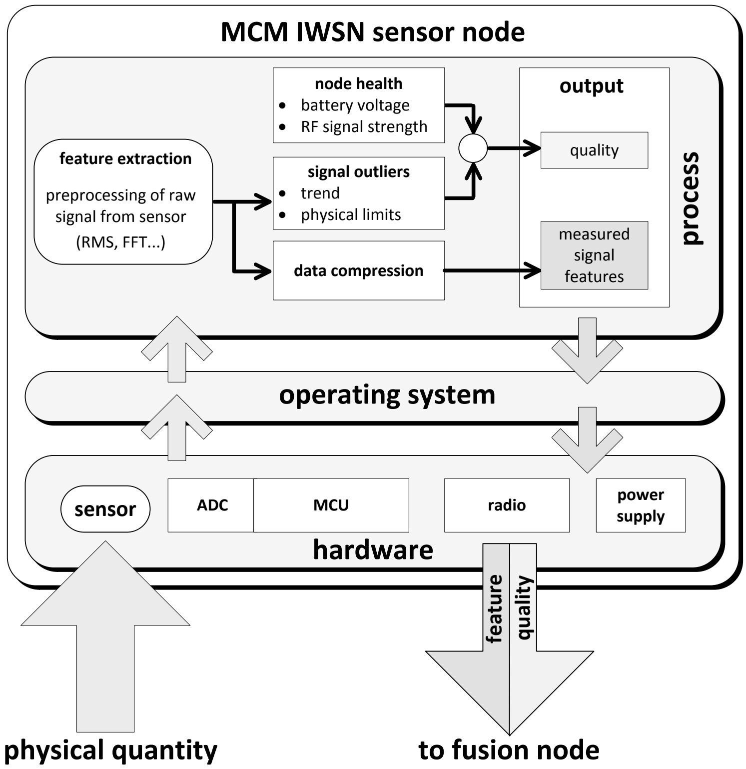

Figure 4.2: Sensor node structure for MCM

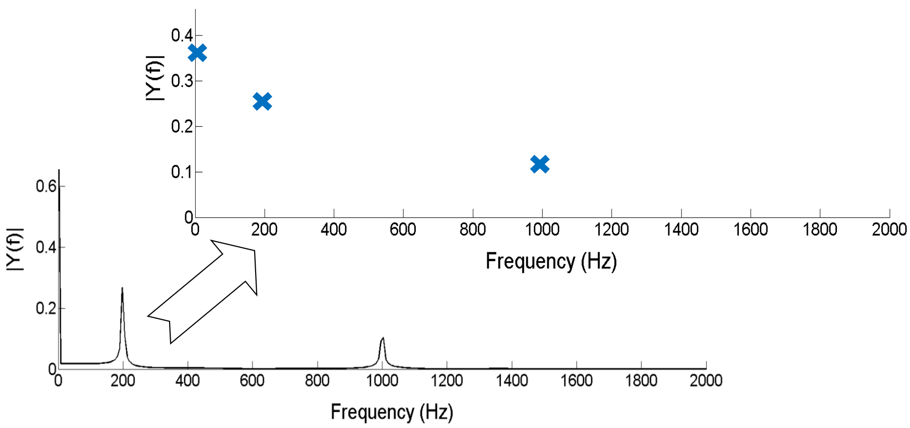

Figure 4.3: Compression of the amplitude spectrum

reliable data stream are sufficient to manage many wireless applications in the same way as if a wire connection was used. However, these communication technologies are mainly developed for hardware designed as mains powered. If energy storage or an energy harvesting system is used as a power supply, many of these sophisticated methods are limited or not applicable because they are too energy-demanding. In recent times, wireless energy-efficient technologies such as WSN have not been designated primarily for a critical system. We propose an enhancement by introducing a distributed data aggregating system.

Unlike enhancing the bottom layers of the OSI model to ensure reliable transmission of data packets between nodes, we focus on the top layers, where data is already linked to physical quantities and where we can use a priori knowledge about signal content to detect faulty data. Our approach is thus complementary, and it is aimed at data degradation that cannot be detected at the lower OSI layers. Theoretical assumptions for dealing with this approach are mainly based on reliability theory. This field of study covers a wide range of knowledge from statistics to economics. We apply a redundancy approach to a number of sensor nodes from this area of study.

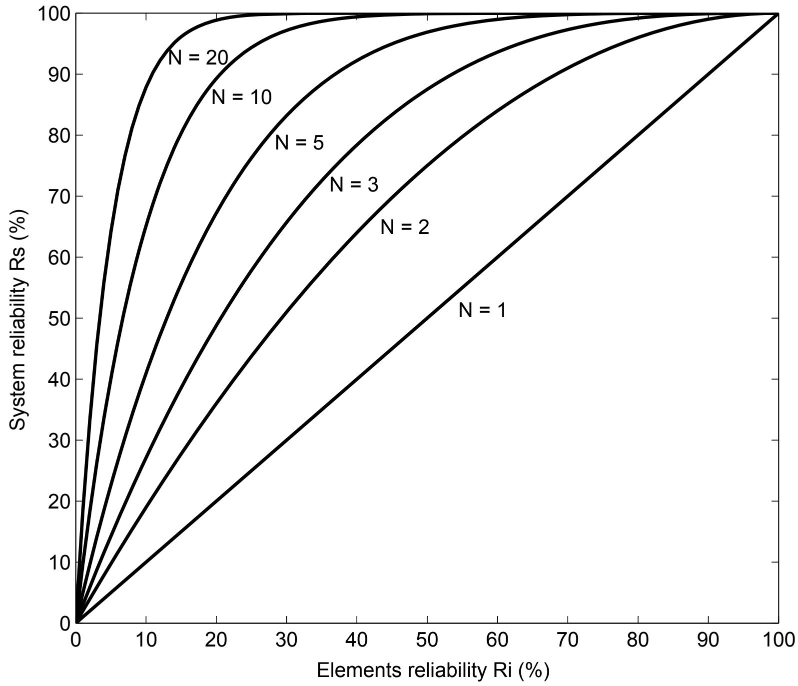

Figure 4.4: Reliability graph of parallel connected components

A sensor node is a complex system including a microcomputer and various peripherals, all controlled by a program code. From the point of view of reliability, a sensor node is a hybrid system formed by serial and parallel blocks (elements) with final reliability $R_i(t)$. If we utilize several sensor nodes sensing the same signal in parallel fashion, the reliability increases significantly, see Fig. 4.4. Failure of such a parallel formation arises when all sensor nodes are faulty. By analogy, the parallel formation is failure-free if one element is working properly. Parallel system reliability R_s is given by equation (4.1), where R_i is the reliability of each element [67].

$$R_s = 1 - \prod_{i=1}^{N}(1 - R_i) \tag{4.1}$$

The parallel components arrangement has the highest reliability of all. Any other system arrangement leads to worse results. The proposed system adopts a parallel arrangement at its input, where more than one sensor node senses the same monitored spot on a device. The sensor node output data are transferred to an aggregation node, see Fig. 4.5. We propose an aggregation node without a backup, and further reliability improvement can be achieved by duplicating the aggregation node or the whole input system, though at the cost of system complexity.

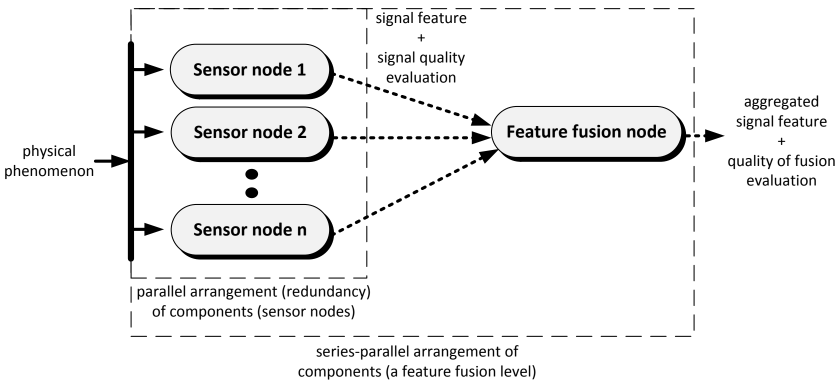

Figure 4.5: Feature fusion level arrangements

4.2.6 Data aggregation

The most powerful signal processing tool in data aggregation systems is information fusion. In this paper, we proceed from the terminology according to Nakamura et al. [68], who introduced fusion into the study of WSN. Nakamura et al. use information fusion and data fusion as a single overall term, with a combination of multiple sources providing improved information (cheaper, higher quality, or greater relevance). The relationship among the input data is used to segregate information fusion into classes (cooperative, redundant and complementary data). In addition, the abstraction level of the manipulated data (raw data/signal, feature, decision) is used to distinguish among fusion processes. Classification on the basis of level of abstraction is preferred, as follows:

- **Raw data fusion**: this type of fusion deals with raw signals acquired by sensors. Proper sampling of rapidly-changing signals, as vibration monitoring requires, produces too many samples for efficient deployment of such fusion within the wireless sensor network. Each passage through a node means a delay, mainly for raw data; the benefit of real-time cannot be achieved, and a changeover to feature extraction is necessary. However, fusion of raw data from sensors connected to one sensor node by wires is reasonable. In our method, we do not utilize raw data as input for the fusion algorithm, though the proposed algorithms are also able to work with raw data.

- **Feature fusion**: this type of fusion involves extracting representative features from the sensor raw data. Many feature vectors are then combined into a single vector. We propose fusion of signal features extracted in the sensor node from the acquired signal. Sensor nodes are placed on machinery in redundancy. This ensures the propagation of valid information through the network and rejection of corrupted data near to the signal source. Feature fusion is driven by a quality parameter composed from the health of the nodes and the descriptors of the measured signal behavior.

- **Decision fusion**: this type of fusion takes into account individual decisions as the input, and combines them to obtain a more confident decision and/or a global decision. If decision fusion is implemented into WSN, we need an efficient computationally undemanding classifier to evaluate the correct/fail state. For example, Hou et al. [69] proposed neural network classifiers for this purpose. A detailed survey of these issues is beyond the scope of this dissertation, but both proposed fusion algorithms can handle an individual decision instead of a signal feature.

A classification of fundamental wireless schemes for MCM is presented in Table 4.2. Each individual category can be combined with each other to create a complex sophisticated sensor network, see Fig. 4.1.

4.2.7 Quality estimation

WSN sensor node functionality can be corrupted externally on its input, its output, or by its own internal malfunction. The sensor node input and its reliability has the same design as other non-wireless devices. Sensors with analog output are connected via ADC. Digital sensors are directly connected to a CPU bus (it can be integrated with CPU(MPU) into one single chip). The internal state of the node is also the same as for other electronic devices, except for significant dependence on a power supply (i.e. low battery level). Wireless output is susceptible to degradation by signal interference.

However, all these issues are solvable by further improving the hardware or data transfer, and we regard WSN as a standardized technology, and available WSN nodes as finished hardware that cannot be changed. It would therefore be very advantageous to have a quality indicator (also referred to in the literature as a symptom) that takes into account system behaviour, and stands for a quantity stating whether or not the sensor data is relevant (i.e. valid/invalid). Especially if this indicator drives the data aggregation process. The quality indicator could be obtained from sensor node resources and measurable parameters:

- power supply voltage,
- the RF strength indicator,
- a signal within the physical limits check (i.e. out-of-range value detection),
- recent trends/slope of the variable (i.e. outlier or jump detection).

Table 4.2: Classification of a signal processing method on the basis of the fundamental WSN systems

Sensing method	**Processing method**	**Fusion method**	**Attribute**
sensor node/s ↓ gateway	• none – raw data • feature extraction • data compression	none raw data fusion (several sensors connected to one node by a wire)	simple strictly limited number of sensors (raw data is transmitted)
several identical sensor nodes (redundancy) ↓ fusion node ↓ gateway	• feature extraction • quality estimation • data compression	feature fusion	improved safety and reliability
several identical sensor nodes / different type of sensor nodes ↓ fusion node ↓ gateway	• classification • quality estimation • data compression	decision fusion (fusion of classifiers)	accuracy complex information about the state of the device

Apart from the last item, which needs a data trend monitoring method, a simple comparison approach for mapping abnormal states of these sources to symptoms can be used, such as a membership function threshold (i.e. a sigmoidal or trapezoidal function). For signal-based abnormal behaviour identification, analysis of previous (i.e. historical) data records over time has to be employed. Evaluating the credibility of the data requires a statistical test for detecting outliers, such as tracing abrupt changes in previous data records. In the case of frequency-domain feature evaluation, the idea is to detect any outliers at the same feature sample in previous records. This idea is depicted in Fig. 4.6. Time-domain records are evaluated in a successive manner along measured periods.

We assume correctness of the data when there is no significant deviation that contradicts a previous data record. There are three possible reasons for an abrupt deviation:

- a fault in the sensor node,
- an abrupt environmental influence affecting the condition of the sensor node,
- a real abrupt change in the condition of the monitored device.

By data trend monitoring we are able to catch many abrupt changes in monitored features, but essentially we are not able to distinguish between them, and mainly to determine their origin. This can make complications to distinguish which data have to be rejected and which transferred further to the sink node. However, we proceed from reliability maintenance theory in section 2.2, which states that an abrupt change of device condition should not occur, because of the expected gradual failure development. In addition, the aggregation process algorithm should transfer data further to the network, if most of the sensor statements are the same (i.e. when they measure the same device place) regardless of the quality indicators. The aggregation quality indicator will in most cases be low. Fundamental rules for the quality indicator:

- The quality indicator has to be a mixture of all resources mentioned here. For data trend monitoring, more than one complementary algorithm can be used.
- An abrupt signal change with high variation in amplitude should be evaluated as low quality, while acceptable minor variations or gradually changing amplitude should be evaluated as high quality.
- One partial quality, mainly from a trend detector, should not by itself make the quality indicator drop near to zero. This should be executed by the out-of-range detector (based on physical limits) or by a combination of several partial qualities.
- The trend monitoring detector should register a drop in quality when a sudden change occurs. If the data resumes an invariable state, the quality should return to a high level as soon as possible.
- Fusion node quality indicator: if most of the sensor node readings record identical data while their quality indicators are near to zero, the output of the fusion node is an aggregation of the readings. Sensors are probably indicating signs of an abrupt failure.

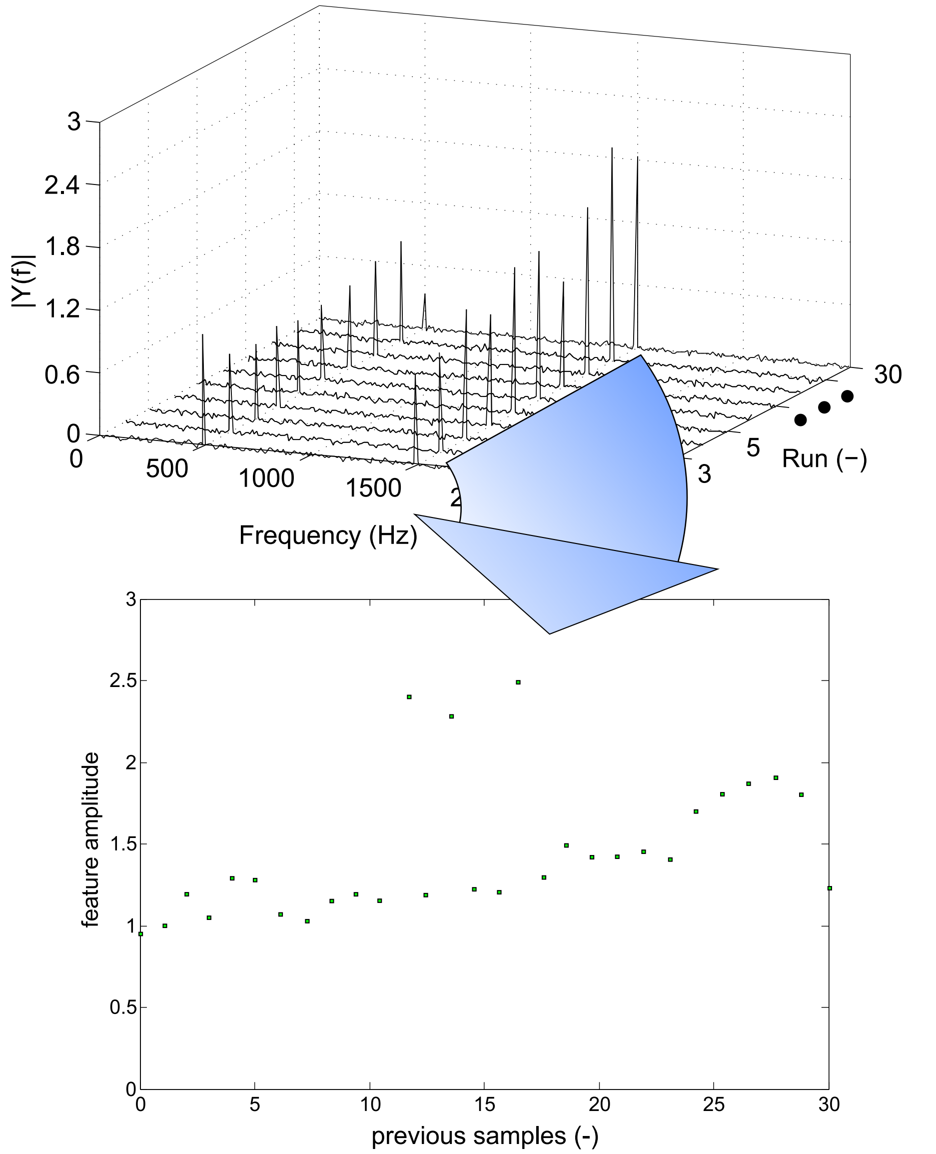

Figure 4.6: A history buffer waveform construction at one feature sample position

There are many statistical methods for data trend monitoring. The limited resources of WSN require computing methods that are simple (i.e. minimally loaded CPU and saving RAM, ROM memory) but efficient. The general scheme for setting the required partial quality indicator is depicted in Fig. 4.7.

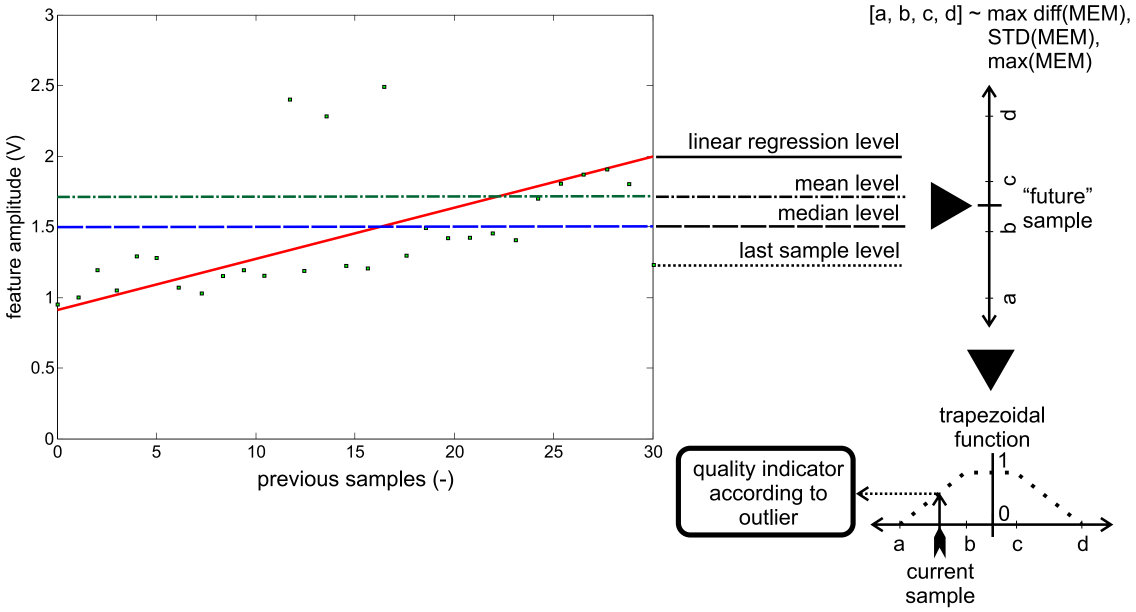

Figure 4.7: Formation of a quality indicator with respect to outliers

First of all, a defined number of previous data records (historical data) are stored in a buffer. Based on a finite set of previous values, the central tendency can be calculated, i.e. the probable value of a "future" sample. We assume that the signal features are normally distributed. In this way, we use the normal distribution as a model for measurement. We therefore propose to use classical well-known statistical methods such as the mean (4.2) and the median (4.3) for calculating a central tendency. As alternative methods, the simple linear regression method (4.6) and the last buffer sample (i.e. simply the previous sample value) are engaged.

The next step is to compute threshold limits. An approach for mapping the current value to the quality indicator by a membership function threshold (e.g. a trapezoidal function) can be used to obtain a multi-value quality evaluation. In addition to these methods, we further take into consideration a standard deviation value (4.4), the maximum difference (4.5) between the values of two consecutive samples and the maximum value. The aim is to find a combination of these functions such that the interval (width) between the "future" sample value and the limit will either be constant with respect to the data range or, more advantageously, such that the width is able to adapt according to the signal variance. The more turbulent the signal, the narrower the limits, and vice versa.

A list of the proposed fundamental functions:

- **arithmetic mean** (find a central tendency, this is influenced by outliers) (4.2),

$$mean(X) = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{4.2}$$

- **median** (find a central tendency, this is more robust in the presence of outlier values) (4.3),

$$median(X) = \text{the middle value of the given numbers (X)} \tag{4.3}$$

- **corrected sample standard deviation** (variation or dispersion from the average) (4.4),

$$STD(X) = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2} \tag{4.4}$$

- **maximum difference** (maximum of the differences between adjacent elements of X along the vector of length m) (4.5),

$$max\_dif(X) = max\left([X(2) - X(1) \quad X(3) - X(2) \ldots X(m) - X(m-1)]\right) \tag{4.5}$$

- **simple linear regression** (fits a straight line (4.6) through the set of n points, so that the sum of the squared residuals of the model is as small as possible).

$$y\_reg(i) = \hat{\alpha} + \hat{\beta}x(i) \tag{4.6a}$$

where $\hat{\beta}$ is:

$$\hat{\beta} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - \sum x_i \sum x_i} \tag{4.6b}$$

and $\hat{\alpha}$ is:

$$\hat{\alpha} = \frac{1}{n}\sum y_i - \hat{\beta}\frac{1}{n}\sum x_i \tag{4.6c}$$

4.3 Information fusion techniques

The roots of information fusion techniques can be found in the late seventeenth and eighteenth centuries, when they were introduced in the works of famous mathematicians such as Bernoulli, Bayes, Lambert, and many others.

We briefly sum up the most widely-used fusion techniques suitable for WSN, and their pros and cons with reference to MCM:

- **Bayesian theory of probability**: in the fusion process of fault diagnosis, Bayesian theory is often adopted. However, Bayesian theory requires probabilities for each question of interest. As a matter of fact, the representation of information absence is badly taken into account by the theory of probability. Indeed, prior and conditional probabilities need to be specified into probabilistic methods. Much hypothesis has been made to simplify the inferring process. In practical applications, these hypotheses will not turn out to be entirely true. In a simplified way, Bayesian methodology works if all necessary information is available. If not, the Dempster-Shafer theory is an alternative to be considered [70, 71].

- **Fuzzy logic**: this method is an effective paradigm for handling imprecision. The method deals with fuzzy uncertainty, which appropriately represents the monitored variable as a linguistic expression (e.g. low, medium, high). Some authors consider fuzzy fusion not to be suitable for a general sensor fusion approach, because the rules are tailored to one specific fusion scenario. However, our proposed fuzzy fusion method attempts to provide complete coverage of the MCM application.

- **The Dempster-Shafer Theory (DST)**: of belief functions – this method is a generalization of the Bayesian theory of subjective probability. Dempster-Shafer Theory was developed by Arthur Dempster, and was then formulated by his student, Glen Shaffer, in a form that is more amenable to reasoning in finite discrete domains [72]. Beliefs in a hypothesis are calculated as the sum of the masses of all of the sets that it encloses. Briefly, DST enables evidence from different sources to be combined and a degree of belief to be arrived at (represented by a belief function) that takes into account all the available evidence. The fundamental scheme of DST, which is computationally excessively demanding due to exponential time in the computation of discernment, has been further developed mainly with the computation savings algorithm presented in [71], [73]. DST implies no relations between the existence and the non-existence of an event. It models only the belief associated to a class, without influencing the belief assigned to other classes. The evidence combination rule of DST provides an operator to integrate multiple items of information from different sources. DST is much more flexible than probability theory. It enables uncertainties, inaccuracies and ignorance to be managed [74].

- **The Kalman filter**: this method is too computationally intensive for WSN, because it involves many matrix multiplications. Abdelgawad et al. [75] have proposed a novel low-power Distributed Kalman Filter (DKF), but its application to larger data volumes is still disputable.

- **Neural networks**: this method is composed of interconnecting artificial neurons. The method is utilized more in classifiers than in data fusion. The main disadvantage is its grey-box like structure, in contrast to fuzzy logic, where the rules can be set up according the experience of an expert. We therefore prefer a fuzzy approach in our proposal.

- **Voting fusion**: this method does not primarily produce information about the quality of the fusion process.

For the reasons presented here, we adopt Dempster-Shafer theory and a fuzzy logic approach to the data fusion node in this thesis.

Chapter 5

Multi-sensor Fusion

The proposed Multi-sensor fusion system is the input section of MCM wireless sensor network, see section 4.5. The inner arrangement of the two cornerstones of this section, the sensor node and the feature fusion node, are described in detail in this chapter. The general WSN idea is a network composed of identical nodes due to overall system complexity and price reduction, see section 2.3. In the same way, the software equipment of each node can be identical. The particular functionality of each node can be selected by a dissemination message from a control unit. The second approach is an Over the Air Programming (OTAP) method, where the nodes are dynamically re-programmed from a remote source (usually the control unit). The chosen method depends on node program memory size and OTAP availability. Imote2 nodes are not OTAP adapted, so we use the first approach, i.e. the unified program code. However, to achieve greater clarity in the following chapters, we describe the sensor node code and the feature fusion node as individual programs.

We adopt available Imote2 techniques to reduce energy consumption. In case of sensor nodes, we use periodical wakening from the deep sleep mode to operational state with CPU speed at 104 MHz. The feature fusion node works at 13 MHz with programmable transmission power. The fusion node sniffs the air for the first wakening packet at low transmission power. If the packet arrives, the fusion node works at higher RF power up to radio silence. A more advanced power management study has already been made, but implementation of its findings lies beyond the scope of this thesis.

5.1 Sensor Node

The proposed sensor node software design is depicted in detail in a block diagram in Fig. 5.1. The initial node settings (referred to as "preset" in the figure) are executed by a dissemination message in connection with the node identification number (ID). The feature extraction method and the invariables are determined. Raw signal data picked up by an integrated sensor or (in this case) by an ADC connected sensor enter the node. The first processing method uses feature extraction. The output of the feature extraction method is an output of the sensor node if the data compression method is not engaged. The same data are saved into the memory buffer as part of overall quality computation based on trend monitoring 4.2.7. Further quality computation sources are the sensor node inner status and checks on whether a feature is within a physical limit or within a predefined pattern.

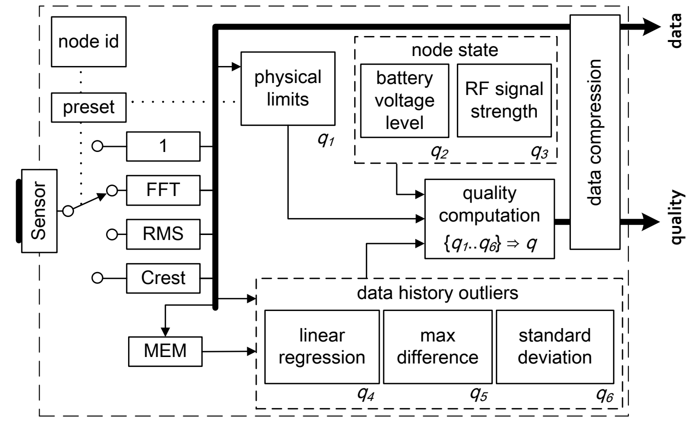

Figure 5.1: Sensor node scheme

5.1.1 Feature Extraction Methods

On the basis of 4.2.3, we adopt RMS (5.1), the crest factor (5.2) and the discrete Fourier transform (DFT) (5.3) of vector x, computed with a fast Fourier transform (FFT) algorithm (5.4). The block labeled as "1" does not perform any extraction method. This option transfers raw time signal data into the node mainly for testing and verification purposes.

$$RMS = \sqrt{\frac{1}{\mathrm{N}} \sum_{i=1}^{\mathrm{N}} \cdot s_i^2} \tag{5.1}$$

$$crest = \frac{peak}{RMS} \tag{5.2}$$

$$X\left(k\right) = \sum_{n=0}^{N-1} x\left(n\right) \exp\left(\frac{-j2\pi nk}{N}\right) \tag{5.3}$$

$$X\left(k\right) = \sum_{n=0}^{N-1} x\left(n\right) W_N^{kn} \tag{5.4}$$

W_N^{kn} is the "twiddle factor", the root-of-unity complex multiplicative constant in the butterfly operations of the Cooley-Tukey FFT algorithm [76], used to combine smaller discrete Fourier transforms recursively. During simulation in Matlab we also use the well-known instruction $y = fft(x)$, based on the Cooley-Tukey algorithm. For HW Imote2 we found inspiration from the developers of AVR embedded systems. Several adaptations of the FFT algorithm have been made for MCU. We use the fixed-point FFT library originally written by Tom Roberts in 1989, with contributions from Malcolm Slaney in 1994 and by Dimitrios P. Bouras in 2006 [77].

5.1.2 Data compression methods

Data compression methods lead to radical bandwidth saving when only the main spectral components are transferred via the network, instead of the original full spectrum. The data compression block provides a list of frequencies and amplitudes of spectral components with amplitudes higher than threshold value T:

$$T = k \cdot RMS \tag{5.5}$$

where the k is a constant depending on the amplitude distribution of the noise, and *RMS* is a root mean square value. We propose the RMS function because of its presence in the sensor node (see the RMS block in Fig 5.1). Another method for setting the threshold could be a level (e.g. a minimal bin) at the position of the first n harmonics, i.e. multiples of the rotating frequency. However, the definition of n requires expert knowledge about the monitored device.

5.1.3 Quality evaluation

The *state of the node* is represented mainly by the *supply voltage* (primarily from a battery) and the strength of the *RF signal* to a neighbor node/s. Both of these quantities have a significant impact on the transferred signal. Although each node is equipped with a voltage regulator, fluctuation in the input voltage and discharging near to low level has a negative influence on the voltage reference of ADC, RF power and all other circuits of the node. RF signal strength could not be used to compute directly in the sensor node because of its absence. The RF received signal strength indication (RSSI) is read directly from the CC2420 Radio and sent with every radio packet. It is possible to use a previous RSSI datum or additionally the current RSSI at the fusion node input.

Data history outliers compare a few previous records with a current value by the trapezoidal membership function (5.6).

$$f(s;a;b;c;d) = \begin{cases} 0, & \mathrm{s}(i) \leq a \\ \frac{s(i)-a}{b-a}, & \mathrm{a} \leq s(i) \leq b \\ 1, & \mathrm{b} \leq s(i) \leq c \\ \frac{d-s(i)}{d-c}, & \mathrm{c} \leq s(i) \leq d \\ 0, & \mathrm{d} \leq s(i) \end{cases} \tag{5.6}$$

The current sample is described as $s(i)$. The parameters a, d locating the 'feet' and b, c locating the 'shoulders' of the trapezoid are computed by a proposed method introduced in table 5.1. Previous data records are collected in the memory block described as *MEM*. If time-domain features are used, the current RMS value or crest factor value is compared with the previous feature record. If frequency domain features are applied, the current sample is compared with the sample at the same position in the previous spectrum record.

The *Physical limits* block contains a simple comparison between the current data and pre-set physical limits. In specific applications where the frequency domain is engaged, simple threshold checking can be substituted for a comparison with an amplitude spectrum pattern.

Quality computation here involves all partial qualities combined into a single overall quality. We model the quality of the physical limit using a two-state logic. If the sample is outside the physical limit, quality q1 is strictly zero. For other qualities we use the geometric mean that single zero partial quality does not drop overall quality q to zero:

Table 5.1: Computation of trapezoidal function parameters

	current sample $s(i) =$ $s(MEM_{last})$; $mean(MEM)$; $y_{regression}(i)$;	
	a	**d**
1	$s(i) - k \cdot max(MEM)$	$s(i) + k \cdot max(MEM)$
2	$s(i) - k \cdot peak(MEM) - diffmax(MEM)$	$s(i) + k \cdot peak(MEM) - diffmax(MEM)$
3	$s(i) - k \cdot peak(MEM) - STD(MEM)$	$s(i) + k \cdot peak(MEM) - STD(MEM)$
4	$s(i) - k \cdot \left\lvert \frac{mean(MEM)}{peak(MEM)} \right\rvert$	$s(i) + k \cdot \left\lvert \frac{mean(MEM)}{peak(MEM)} \right\rvert$
5	$s(i) - k \cdot \frac{max(MEM)}{max(MEM)+STD(MEM)}$	$s(i) + k \cdot \frac{max(MEM)}{max(MEM)+STD(MEM)}$
6	$s(i) - k \cdot \frac{max(MEM)}{max(MEM)+diffmax(MEM)}$	$s(i) + k \cdot \frac{max(MEM)}{max(MEM)+diffmax(MEM)}$
7	$s(i) - k \cdot \frac{max(MEM)}{max(MEM)+\left\lvert 1-\frac{mean(MEM)}{median(MEM)} \right\rvert}$	$s(i) + k \cdot \frac{max(MEM)}{max(MEM)+\left\lvert 1-\frac{mean(MEM)}{median(MEM)} \right\rvert}$
	b	**c**
1	$s(i) - l \cdot max(MEM)$	$s(i) + l \cdot max(MEM)$
2	$s(i) - l \cdot peak(MEM) - diffmax(MEM)$	$s(i) + l \cdot peak(MEM) - diffmax(MEM)$
3	$s(i) - l \cdot peak(MEM) - STD(MEM)$	$s(i) + l \cdot peak(MEM) - STD(MEM)$
4	$s(i) - l \cdot \left\lvert \frac{mean(MEM)}{peak(MEM)} \right\rvert$	$s(i) + l \cdot \left\lvert \frac{mean(MEM)}{peak(MEM)} \right\rvert$
5	$s(i) - l \cdot \frac{max(MEM)}{max(MEM)+STD(MEM)}$	$s(i) + l \cdot \frac{max(MEM)}{max(MEM)+STD(MEM)}$
6	$s(i) - l \cdot \frac{max(MEM)}{max(MEM)+diffmax(MEM)}$	$s(i) + l \cdot \frac{max(MEM)}{max(MEM)+diffmax(MEM)}$
7	$s(i) - l \cdot \frac{max(MEM)}{max(MEM)+\left\lvert 1-\frac{mean(MEM)}{median(MEM)} \right\rvert}$	$s(i) + l \cdot \frac{max(MEM)}{max(MEM)+\left\lvert 1-\frac{mean(MEM)}{median(MEM)} \right\rvert}$

$$q = q_1 \cdot \sqrt{mean\,(q_2 \ldots q_n) \cdot 2ndpercentile\,(q_2 \ldots q_n)} \tag{5.7}$$

Overall quality q is computed from partial qualities q1 to q5. With the exception of quality q1, strictly assigned to zero if the value is outside defined physical limits, other partial qualities are computed by a membership function. Quality q2 representing battery discharging uses a sigmoidal membership function. Other qualities use trapezoidal membership functions. The membership curve depends on scalar parameters.

If RSSI is computed in the fusion node the final quality equation (5.7) takes on the form:

$$q = \sqrt{q_{received} \cdot q_{RSSI}} \tag{5.8}$$

where $q_{received}$ is (5.7) without a readout on RF signal strength.

5.2 Data Fusion Node

The data fusion node consists of a data fusion block with the implementation of a data fusion algorithm. We verified DST-based fusion and fuzzy-based fusion. The idea of a fusion node requires an algorithm that produces a quality estimate of the fusion process in addition to the fusion result. The other blocks correspond to the sensor node structure.

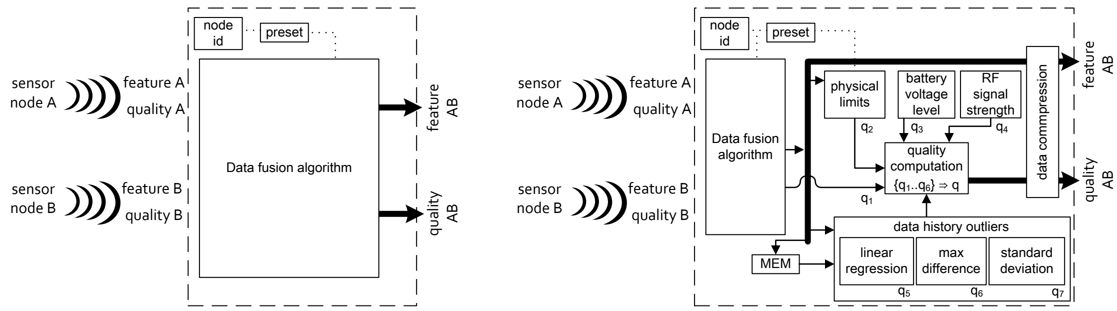

Figure 5.2: Fusion node scheme; left - simple fusion node; right - fusion node enhanced to next fusion process

5.2.1 Dempster-Shafer theory based fusion method

Our interpretation deals with Barnett's algorithm [71], which is a special case of belief functions focused on singletons (one-element subsets), so the computation requires only linear time. Barnett suggests the following strategy:

First, he defines a frame of discernment Θ as a partitioning comprising n elements, i.e., $\Theta = \{i \mid 1 \leq i \leq n\}$. Then, define $\mu_{\{i\}}$ as a basic probability assignment and the orthogonal sum of μ_{ij} as a collection of basic probability assignments that represents evidence in favor of proposition i for each $i \in \Theta$ and j runs through the sets of experiments that confirm proposition i.

Thus μ_i is the totality of the evidence in favor of proposition i, and $f_i = \mu_i(\{i\})$ is the degree of support from this simple support function. Similarly, Barnett defines collection ν_{ij} as evidence against i and in the same way ν_i, $a_i = \nu_i(\neg\{i\})$, where $\neg\{i\} = \Theta - i$.

A combination of all the evidence directly for and against i is the separable support function, $e_i = \mu_i \oplus nu_i$, referred to as the simple evidence function. There are n of them, one for each $i \in \Theta$.

In the case of an WSN-based MCM system, it can be assumed that there does not exist a sensor reading that speaks against a single proposition i, similarly as proposed in [38]. Then basic probability numbers for e_i that are not identically zero ($c_i = 0$, $a_i = 0$) are $p_i = e_i(\{i\}) = \mu_i(\{i\})$, $r_i = e_i(\{\Theta\}) = 1 - \mu_i(\{i\})$, $d_i = 1 - p_i = c_i + r_i = 1 - \mu_i(i)$.

Now, we are able to determine the conflict in the evidence K, the final mass function $m(A)$, and the position of $A_{max(m(A))}$. Finally, the rewritten equations for computing μ_i, K and $m(A)$ where $\forall i \in A$, $A \subseteq \Theta$, $S = \{j | 1 \leq j \leq \mathrm{k}\}$ are as follows:

$$\mu_{ij}(\{i\}) = q_i \times \int_{segment\ i} g(x)dx \tag{5.9}$$

$$\mu_i(\{i\}) = 1 - \prod_S (1 - \mu_{ij}(\{i\}) \tag{5.10}$$

$$K^{-1} = \prod_{i=1}^{|A|} (1 - \mu_i(\{i\})) \left(1 + \sum_{i=1}^{|A|} \frac{\mu_i(\{i\})}{1 - \mu_i(\{i\})}\right) \tag{5.11}$$

$$m(\{q\}) = K \cdot p_q \prod_{i \neq q} d_i \Rightarrow m(\{i\} = K \cdot p_i \frac{\prod_{i=1}^{|A|} d_i}{d_i}) \tag{5.12}$$

The fusion algorithm can be divided into four blocks, according to the main computation tasks, see Fig. 5.3:

1. The input data range is delimited (min, max value) and is fractioned into segments of constant width. Optionally, we establish a dynamic range correction so that the median of all valid values lies in the middle of a segment. Finally, the Gaussian function $g(x)$ is constructed over the value from a node (the Gaussian interprets the distribution of the measurement uncertainty in addition to the fact that the area of the bell curve is always equal to 1).

2. Each item of evidence is represented as a mass function $\mu_{ij}(A)$, the value of which on segment i is the area under the Gaussian bell curve multiplied by the evaluation quality (5.9). Then the basic probability assignment μ_i is computed as the orthogonal sum of μ_{ij} by using (5.10).

3. The combination of evidence is processed by Dempster's rule, so that it produces the final mass function $m(A)$, using (5.12). The quantity $m(A)$, is called A's basic probability number. It represents our exact belief in the proposition represented by A (the most credible segment). In other words, this quantity represents the fusion quality in our scheme.

4. As a result of fusion the most credible segment loses precision, since it is wider than a single input value. We therefore adopt a weighted average to regain precision for the fusion result f according to (5.13).

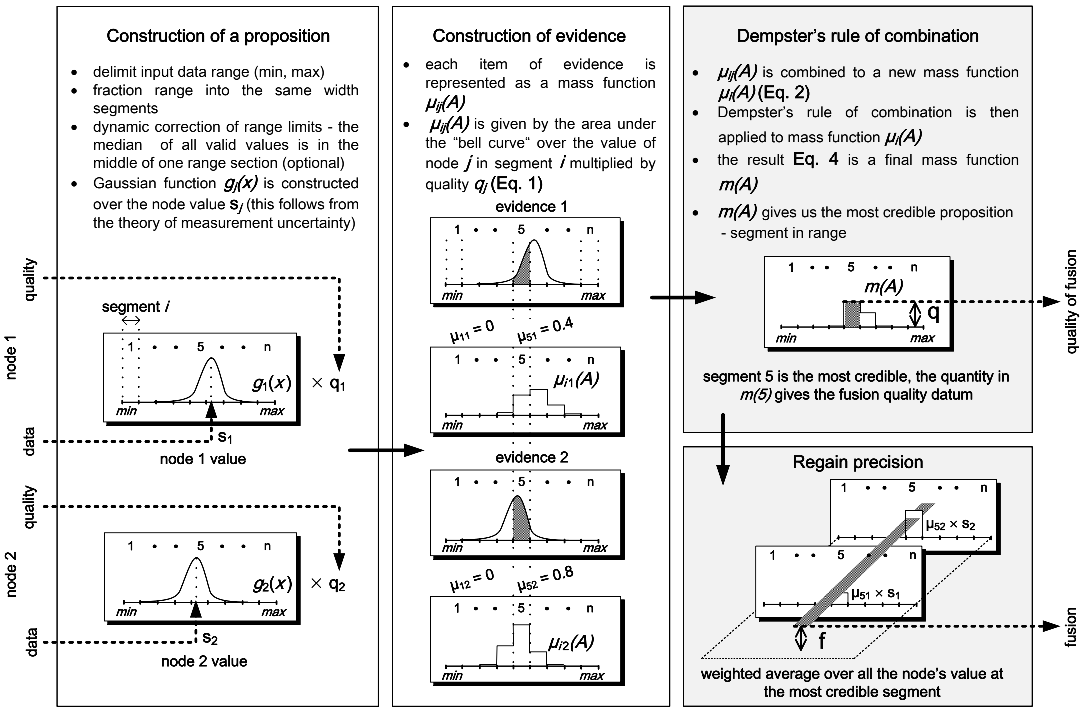

Figure 5.3: The Dempster-Shafer-based data fusion process

$$f = \frac{\sum_S \mu_{ij}(\{i_{\max(m(A))}\}) \times s_j}{\sum_S \mu_{ij}(\{i_{\max(m(A))}\})} \quad S = \{j | 1 \leq j \leq \mathrm{k}\} \tag{5.13}$$

5.2.2 Fuzzy logic-based fusion method

Fuzzy logic-based data fusion is an effective paradigm for mapping an imprecise input space to an output space. The general fuzzy system is defined by inputs, outputs, membership functions associated with a given fuzzy set and a list of IF-THEN statements, called rules.

Our problem has $2j$ input variables (j feature data, j quality data) and two output variables (fusion, quality of fusion). Input feature samples are normalized, i.e. the highest sample value is equal to 1. We use the Mamdani scheme for the fuzzy system in Fig. 5.4. We have proposed two sets of rules. The first set is for the fusion output, and the second set is for the quality output. To create a general set of rules we proceed from a generic truth table (established according to Boolean logic) and simple expert knowledge.

Let us assume two sensor nodes. The value of the first node is high (1), and the second also has a high value (1). The quality indicator of the first sensor node is high (1) and the quality of the second is also high (1). The fusion result and the quality of the fusion have to be high, or both have to be low in the opposite situation. It is evident that a few contradictory statements occur. For example value 1 – high, quality 1 – high and value 2 - low, quality 2 – high. If we repeat this approach for all alternatives we get the complete truth table 5.2. To simplify logic expressions - i.e. to write a minimal Boolean expression representing the required logic - we use the well-known Karnaugh map.

Table 5.2: Generic truth table for 3 nodes

quality node 1	feature node 1	quality node 2	feature node 2	quality node 3	feature node 3	feature fusion moderate	feature fusion radical	quality of fusion moderate	quality of fusion radical
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	1	1	1	1	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	0	1	0	1	1	0	0	1	0
1	0	1	1	0	0	1	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	1	1	1	0	0	1	1	1	1
1	1	1	1	0	1	1	1	1	1
1	1	1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1

We therefore prepare a "moderate" version and a more "radical" version of the rules. The moderate approach, unlike the radical approach, means that in the event of a contradictory statement the result is high. These rules can be verbalized as follows:

Moderate fusion output – is high

- If at least one half of the sensors declare both feature and quality high.
- If a minority of the sensors declare both feature and quality high and at least one half of the other sensors have quality low.
- If a minority of the sensors declare both feature and quality high and the same number of sensors have quality high and feature low.

Radical fusion output – is high

- If more than one half of the sensors declare both feature and quality high.
- If a minority or one half of the sensors declare both feature and quality high, and more than one half of the other sensors have quality low.
- If exactly one half of the sensors declare both feature and quality high and one half of the other sensors have quality low and feature high.

Moderate quality of fusion output – is high

- If at least one half of the sensors declare the quality high and their features have the same value (0 or 1), and if the other exact half of the sensors are not in direct contradiction (f1 – high, q1 – high; f2 – low, q2 – high).

Radical quality of fusion output – **is high**

- If more than one half of the sensors declare the quality high and their features have the same value (0 or 1).

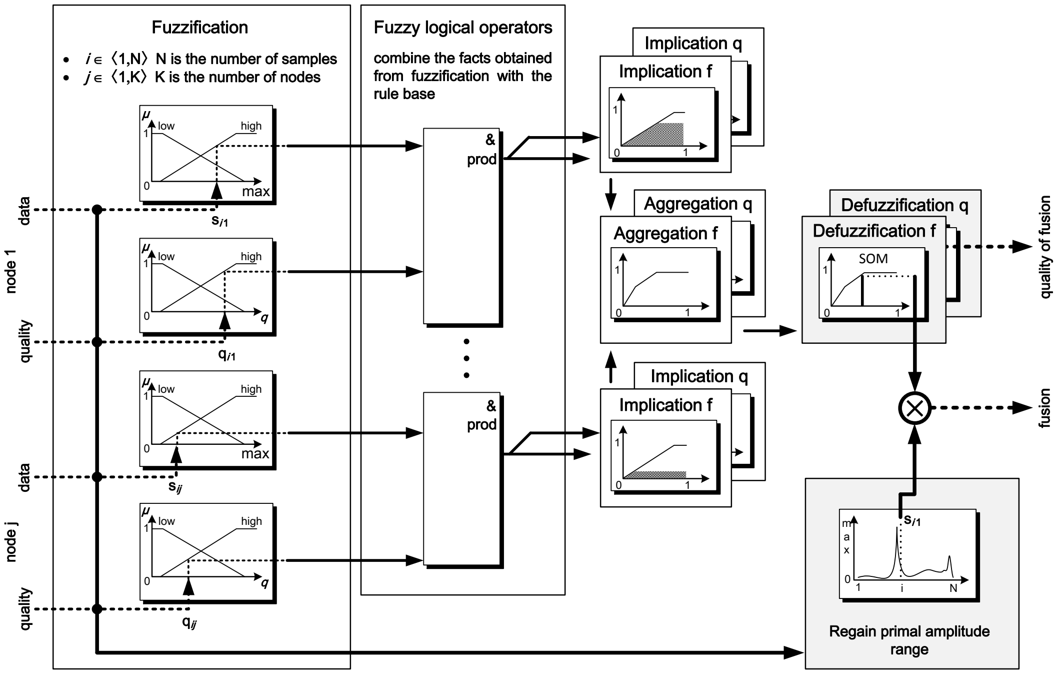

Figure 5.4: The fuzzy logic based data fusion process

The next step is to combine these rules into a rule-based system (e.g. nine rules for the moderate fusion output). We define the membership functions for fuzzification as trapezoidal-shaped for feature input and quality input, and also for both defuzzification outputs.

Finally, both outputs of fuzzy fusion lie in the interval $\langle 0, 1\rangle$. This range is adequate for quality evaluation of the fusion, but the amplitude of the fusion output has to lie within the same range as the feature input. This is arranged by multiplying the highest input feature sample by the resultant sample of fuzzy fusion (see Fig. 5.4).

Chapter 6

Implementation of the proposed methods

At the beginning of this doctoral project the most advanced WSN systems were offered by Crossbow Technology, Inc. This is the same company that developed Mica, one of the first self-contained sensor nodes, see section 2.3.1. Therefore Implementation and algorithm development were therefore fitted to the most powerful Crossbow WSN system of that time. This platform was called Imote2, and Crossbow offered it within a development kit for research purposes. Imote2 is based on Intel PXA 271 XScale 32-bit CPU with 32 MB SDRAM and 32 MB Flash. Today, it would be classified as a high-performance sensor node, according to our node categories in section 2.3.1.

Although the real WSN network might be adequate for carrying out MCM WSN experiments, the proposed ideas should be as general as possible. This means that they should not be focused on a specific hardware. A high-level interpreter language was therefore chosen as an appropriate way to establish processing methods and carry out performance tests on them. Programs in the interpreter language are read, interpreted and executed line by line. This takes a relatively long computation time, but it is more convenient for making prototypes and for verifying proposed algorithms than directly programming an embedded system. In our case, Matlab technical computing language was chosen.

6.1 Developing and optimizing mathematical models in MATLAB

A fundamental Matlab code was created to verify the proposed algorithms (see Fig. 6.1). This code simulates propagating a signal through an MCM WSN system from the sensing node to a middle node. The nodes are created as functions where proposed algorithms are applied. The fundamental code is based on the main test loop. Each run of this main loop starts a set of measurements in the WSN model under test conditions varying together with the increasing loop counter. The test conditions affect the parameters of the proposed algorithms. The set of measurements within the main loop is also fully adjustable. The node health status can be affected, and this changes the quality of the propagated signal. The set of measurements is adequate for running the WSN system from its starting point to a stop at a defined event (e.g. the end value for a loop). One measurement run produces a single measurement by a specified WSN system (e.g. number of sensors) for a defined time and sampling rate. This scheme enables data to be gathered at any stage of the code, and thus at any position in the WSN system. A comparison of partial results serves to find a proper setting for the WSN system and to evaluate the efficiency of the proposed algorithms.

6.1.1 Input data to the model

When fundamental variables defined at the beginning of the main code are initialized, the signal generator block/function creates input data for the sensor node. This function is designed to obtain a signal suitable for verifying all proposed signal processing methods. The harmonic signal created by the sum of a defined number of sinusoids varying in amplitude and frequency was engaged. The harmonic signal is advantageous as the input signal for the next section, due to its well-calculable signal features, and in this way the output fluctuation of each WSN section can easily be compared.

6.1.2 Data passing within the model

Signal sensing, analog-digital conversion and radio transfer in WSN can easily lead to signal interference, as has already been pointed out. The program code reflects this property by controlled data degradation at positions where data are passed between the function blocks of the node. Signal errors are created by the signal jammer function.

The harmonic signal produced by the signal generator function goes through the signal jammer function, where it is randomly degraded by long-term noise, short-term noise and loss of connection. Other properties influencing the signal in the time domain, e.g. time delay, phase shift and others, have been omitted due to signal feature transfer instead of a raw signal (see 4.2.3).

The signal jammer function is based on a pseudo random number stream activating the signal degradation. The RandStream constructor is used to ensure that there is the same pseudo-random number stream during each run of the code. The pseudorandom number stream is controlled by the seed parameter. The seed parameters were determined according to the richness of the signed degradation.

6.1.3 Sensor node

The sensor node processes described by the block diagram presented in section 5.1 were converted into the Matlab function. A simplified function code flowchart is presented in Fig. 6.2. The input to this function comprises data from the signal generator function representing the input signal from a real sensor or ADC. Information about the feature extraction method that will be used and a declaration of the fundamental variables are defined at the beginning of the m-code. The signal feature and the quality evaluation form the output of the sensor node function.

The first step in the run of the code is a case structure for switching between preprocessing methods constructed as individual functions. The case structure produces an array containing the computed feature (e.g. the frequency spectrum from an input signal). The next step is a check on whether the memory block (MEM) exists or not. A structure data type is used to create the MEM block. The MEM structure is the global variable, and besides buffering previous data, it stores the

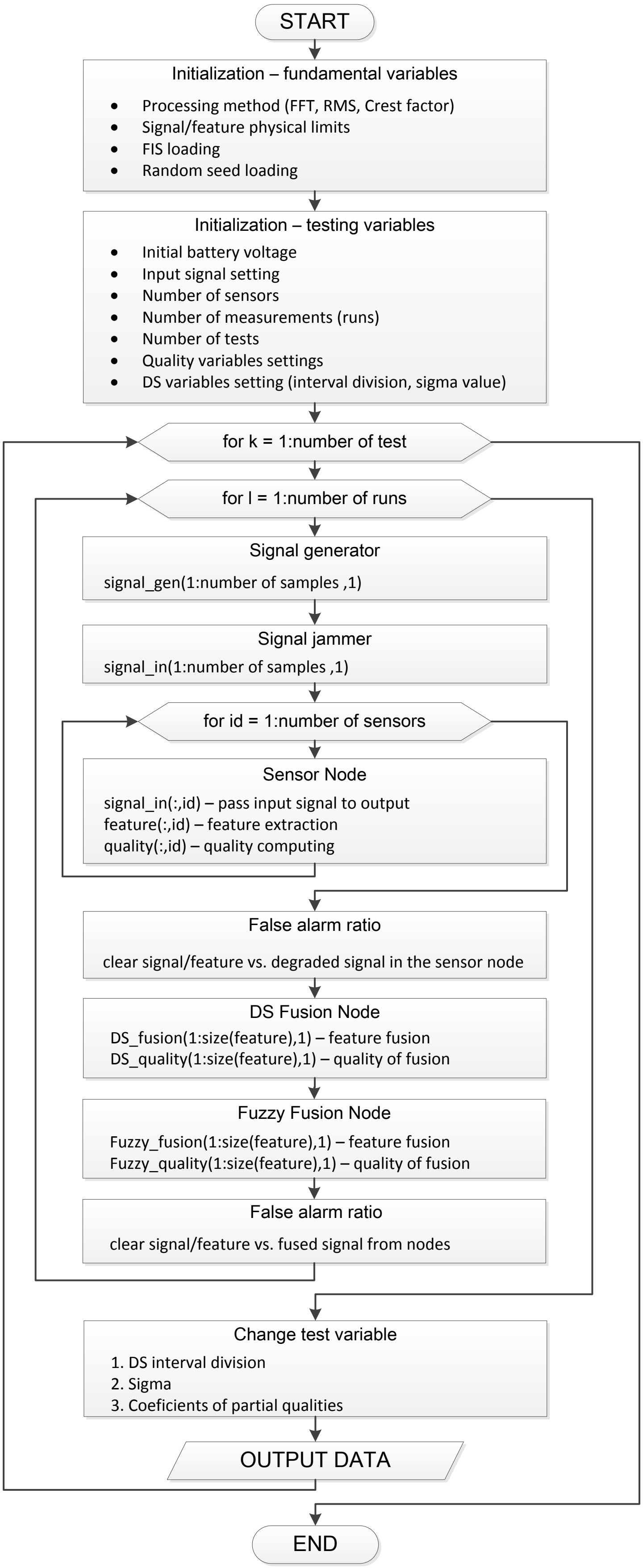

Figure 6.1: Fundamental WSN flowchart in Matlab

variables required for subsequent verification or testing. If the structure does not exist, a new structure is constructed. Otherwise, the program uses the existing structure.

The next section is a quality evaluation accomplished by the equations presented in section 5.1.3. The matrix operations for computing these functions were used modestly, because the high-level Matlab algorithms were subsequently to be transferred into C language. Similarly, the math functions were mostly re-written in C code instead of using embedded Matlab functions.

At the end of sensor node function, actual data for subsequent computation in the main code are saved to return variables. Data requiring storage are saved to relevant fields at the end of MEM structure.

6.1.4 Data fusion node

The data fusion node function is mainly based on two individual functions making data fusion. The Matlab code of the first function according to Dempster Shafer theory is a direct analogy to the Barnett algorithm depicted in Fig. 5.3. It begins by dividing the input signal range using the median subfunction. After that, the bell curve with the center at the input sample value is created by the normal probability density function. The appropriate surface area is computed by numerical integration and then the final mass function is computed. The DST fusion function also contains simple arithmetic operations, and finally it saves the data to output variables.

The second function, fuzzy-logic based fusion was created by Matlab Fuzzy Logic Toolbox in relation to the theories presented in section 5.2.2. This toolbox offers an intuitive interface for creating a fuzzy-logic based solution in a convenient manner. The proposed function is graphically formed into the complete fuzzy system with step by step selection of all parts of the fuzzy system: type of system, fuzzification setting, definition of rules, implication method, aggregation method and defuzzification setting. The result is saved into an .fis file, which is loaded into the main code, and it works as a function mapping the input to the output according to the fuzzy system that was prepared in the toolbox.

6.2 Implementation into real WSN hardware

Implementation into a real embedded system was based on the WSN development kit Imote2. The latest version of the Imote2.NET network was purchased in order to test the proposed system. The main difference from earlier types, which used conventional operation systems (e.g. TinyOS), was in the .NET software platform utilization. The main parameters of the Imote2 network are:

- Intel/Marvell PXA271 XScale Processor at 13-416 MHz,
- MMX DSP Coprocessor,
- 256 kB SRAM, 32 MB FLASH, 32 MB SDRAM,
- integrated 802.15.4 radio
- I/O: 3 × UART, 2 × SPI, I2C, AC97

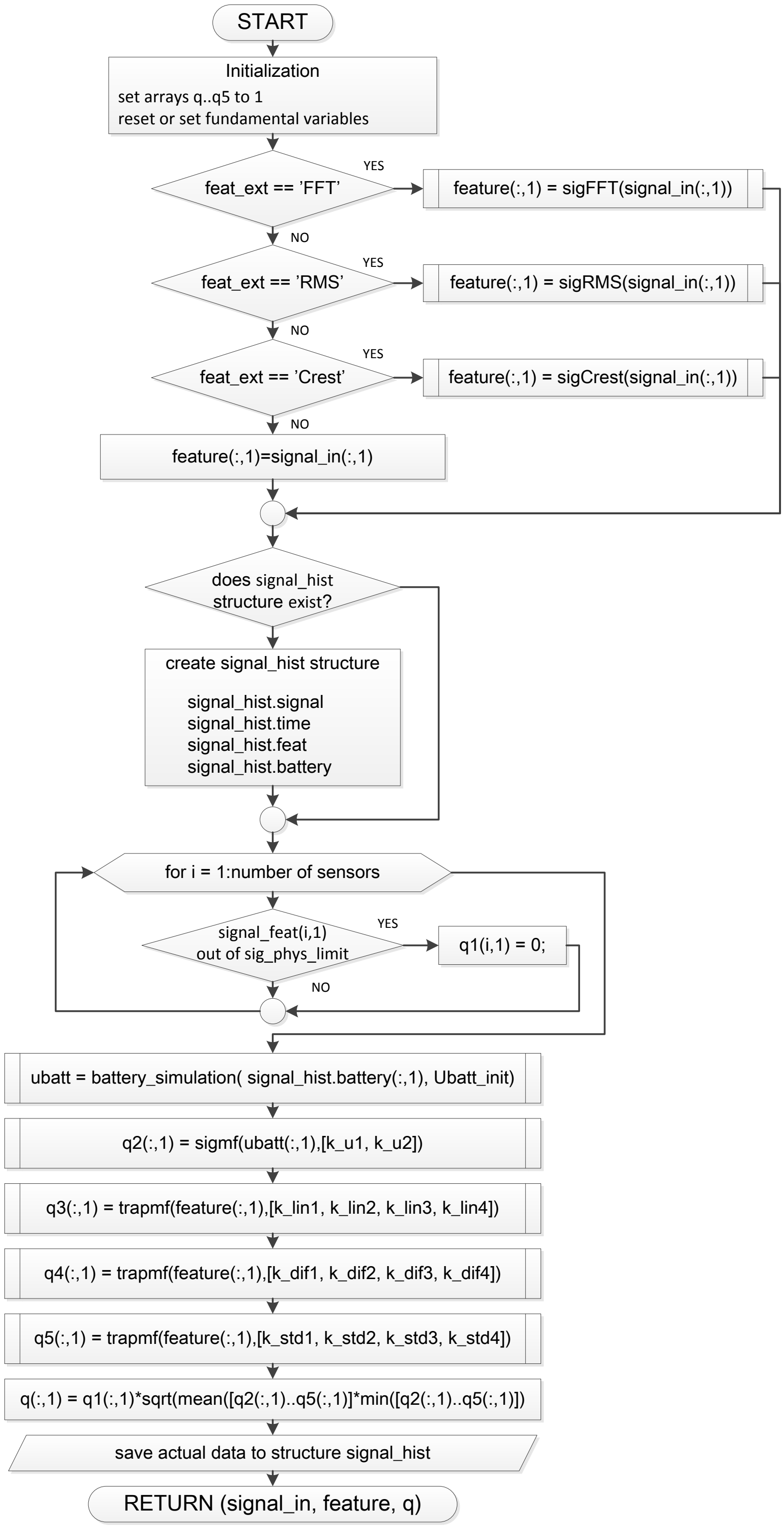

Figure 6.2: Sensor node function flowchart in Matlab

- built-in battery charger (Li-Ion, Li-Poly)
- pre-installed .NET Micro Framework

The datasheet parameters of this advanced WSN platform refer to wide abilities for many applications, sufficient also for our purposes. Micro Framework is the smallest version of .NET (a Microsoft operating system platform that incorporates applications, a suite of tools and services) for very resource-constrained devices. It offers a complete and innovative development and execution environment that brings the productivity of modern computing tools to embedded programming (see section 2.3.2). This was the main reason for using Imote2 hardware for the proposed experiments. Putting the Imote2 platform into operation was a task carried in a master project [78] supervised by the author. Working on this task led to the unpleasant discovery that the micro framework is not properly connected to the hardware resources. Many hardware features were inaccessible due to deficient, corrupted or completely missing libraries. Crossbow offered poor support. Unfortunately, the promised update solving these issues has never been published, and all WSN activities at Crossbow were later terminated.

According to other users' experience of this system, the pre-installed .NET micro framework was replaced by applications written under TinyOS, described in section 2.3.2. TinyOS has implemented almost the complete hardware layer of Imote2, including peripherals and sensor board. The most significant disadvantage is that debugging is almost impossible. Only three LEDs can help to monitor code behaviour. The latest TinyOS 2.1.2 version enables "printf" function to Imote2. It extends the ways to control applications and debug them. There is no debug mode allowing step-by-step running of code or memory monitoring, which is available for other embedded systems. A detailed description of how to install TinyOS and how to adopt some settings useful or required for this project was published in another master thesis supervised by the author [79] and in a recently published technical report [C1].

6.2.1 Real WSN system

The sensor node of the Imote2 network is assembled from stackable boards: the battery board, the MCU board IPR2410, and the sensor board ITS400. Programming and serial communication with a node is provided by interface board IIB2400, see Fig. 6.3. Sensor board ITS400 is equipped with basic onboard sensors to sense temperature, humidity, acceleration and light intensity, and a general purpose ADC. ADC MAX1363 provides four 12bit channels with voltage range 0 - 3 V. It is used as the sensing input of the sensor node. No other peripherals of this board are used in our experiments.

The maximum sampling rate of ADC is around 7.5 kHz when the 32 kHz timer is engaged and the CPU speed is increased to 104 MHz mode. Higher CPU frequencies require higher core voltage, and were therefore not tested.

6.2.2 Input data

To evaluate the proposed algorithms we used two independent sources. The first source, an arbitrary signal generator, was used for simple test execution, mainly for

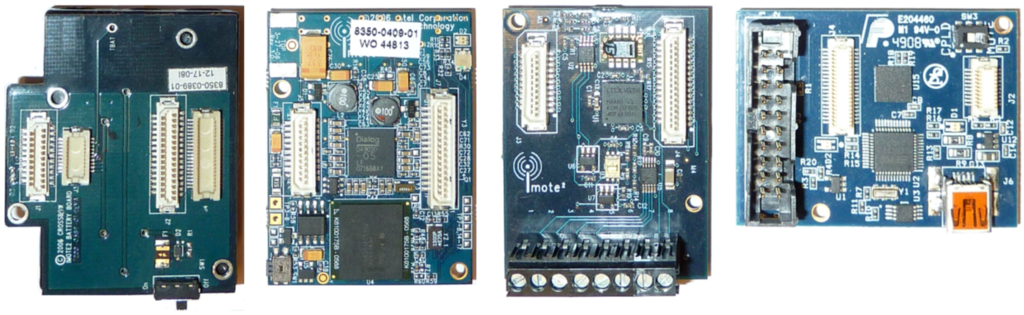

Figure 6.3: Imote2 stackable boards; from left: battery board for 3 × AAA Accu, MCU board IPR2410, sensor board ITS400 and interface board IIB2400

feature extraction evaluation. The second source was analog output card 9264 with a 9162 USB carrier from National Instruments. The fully programmable output of this card allows the same tests to be performed as in a WSN Matlab simulation, and also enables real data acquired at a mechanical device to be replayed by an NI 9234 card, see Fig. 6.4.

6.2.3 Network topology

A tree topology with the acknowledgement function to control loss of data frames was selected. There are more sophisticated communications protocols, but this solution ensures that no other data manipulation implemented in a lower communication layer influences the results. In addition, during the performance test of the fusion algorithms, the data produced by each node was sent to the sink node separately in time.

6.2.4 Sensor node

The program code under TinyOS is written in NesC language. This is an event-driven extension to the C programming language described in [48,80]. A nesC application consists of one or more components linked together to form an executable. A component provides and uses interfaces. These interfaces are the only point of access to the component and are bi-directional. An interface declares a set of functions called commands that the interface provider must implement, and another set of functions called events that the interface user must implement. There are two types of components in nesC: modules and configurations. Modules provide application code, implementing one or more interface. To support larger computations, TinyOS provides tasks. A task is a function which a component tells TinyOS to run later, rather than now. The application code comprises event handlers, commands, tasks, and local procedures.

The simplified flowchart in Fig. 6.5 presents the fundamental application code idea for the sensor node. The code begins with the inclusion of the header files and the subroutines, a declaration of the used interfaces, a declaration of the used variables and the setting of the fundamental constants. The header files declare packet, timer and ADC settings. There has to be an interface for Timers, ADC read, PMIC (power management), SplitControl (to switch between the on and off power states of the component, e.g. ADC and radio) in addition to common resources. All of the user code should start from the signal that all components are initialized

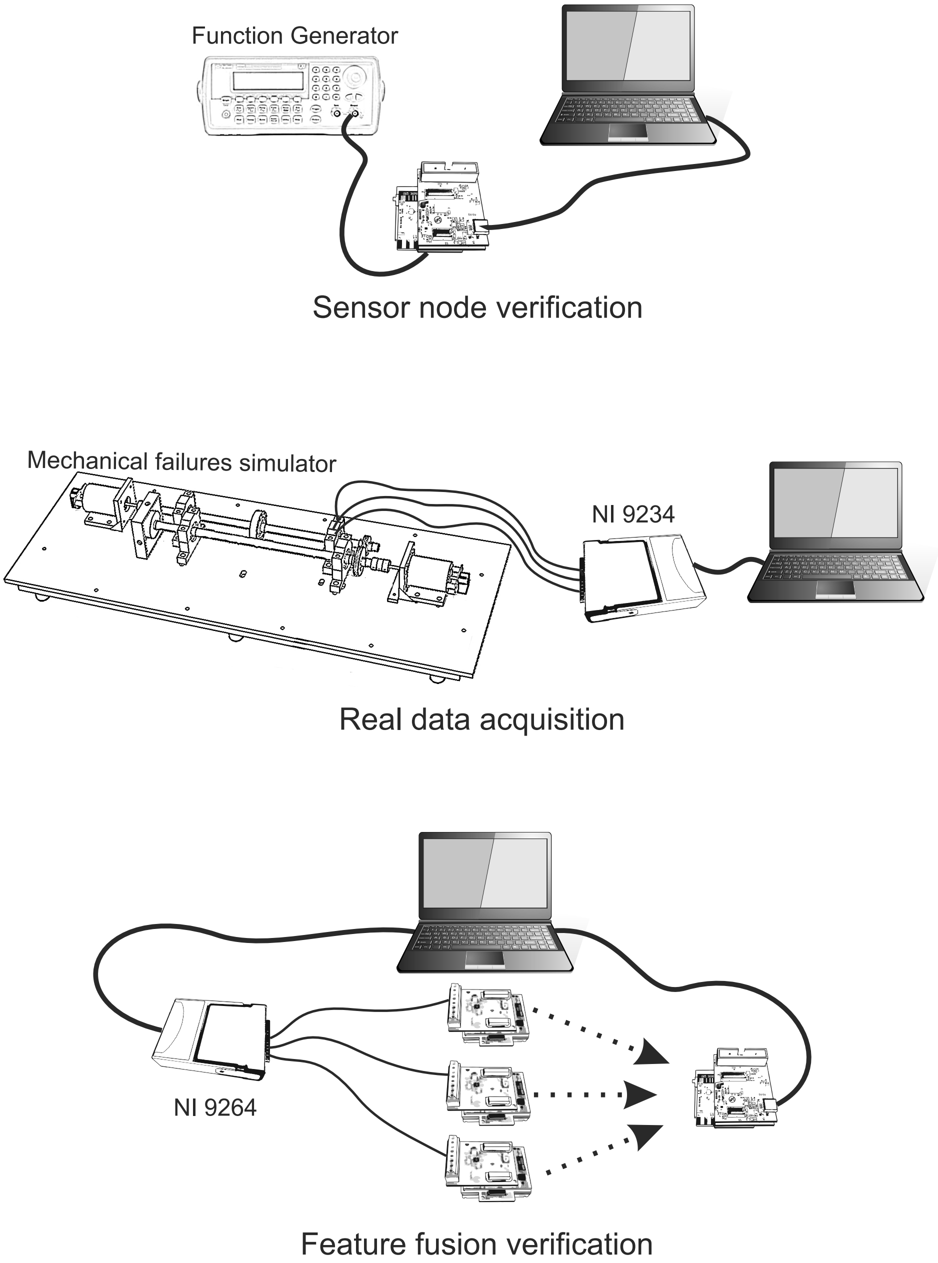

Figure 6.4: Verification of proposed methods

and working. This signal is the event *Boot.booted()*. All events are able to signal an error during their execution. Errors are handled by a LED blinking.

If all components are booted, a condition to start the measurements has to be implemented. We use a slower "Timer 1" with single firing. If the sensor (in our case ADC) is engaged, the sampling loop is started. This loop is controlled by a fast "Timer 0", adjusted according to the required sampling rate, to gather a continuous signal sequence for a defined time. The power supply voltage checking command can be included in this loop to obtain the battery voltage value together with each sample. However, any kind of code included in this loop has a considerable effect on the maximum sampling rate.

When the input signal is acquired, a signal feature is computed according to a switch pre-set to one feature method. The first i runs of the program save the feature waveforms into the memory buffer. After the buffer is filled, partial qualities for determining the final quality are computed per feature sample. At the end of this loop, the actual data are saved to the buffer and are sent to the fusion node. Finally, "Timer 1" is already started for repeated sensor node measurements.

6.2.5 Data fusion node

The application code for the fusion node is based on receiving messages from the sensor nodes and saving them into a buffer. If the buffer contains relevant messages from all sensor nodes the radio is switched off and data fusion is executed. Finally, the result of the fusion algorithm is sent to the sink node. Two individual approaches are possible, the first when one fusion is executed per sample, and the second when one fusion is executed per complete measurement (batch processing). This choice depends on the communication protocol. For the performance test, we chose batch processing. Fusion is executed when the whole measurement sequence is received. However, the first option was also successfully tested together with data transfer based on the acknowledgement function.

DST fusion is computed according to the algorithm described in detail in section 5.2.1.

The fuzzy fusion algorithm is a direct translation of the FIS Matlab file into C code. It was carried out by the Arduino FIST: MATLAB Fuzzy Inference System Converter by Peter Geiger and Karthik Nadig [81].

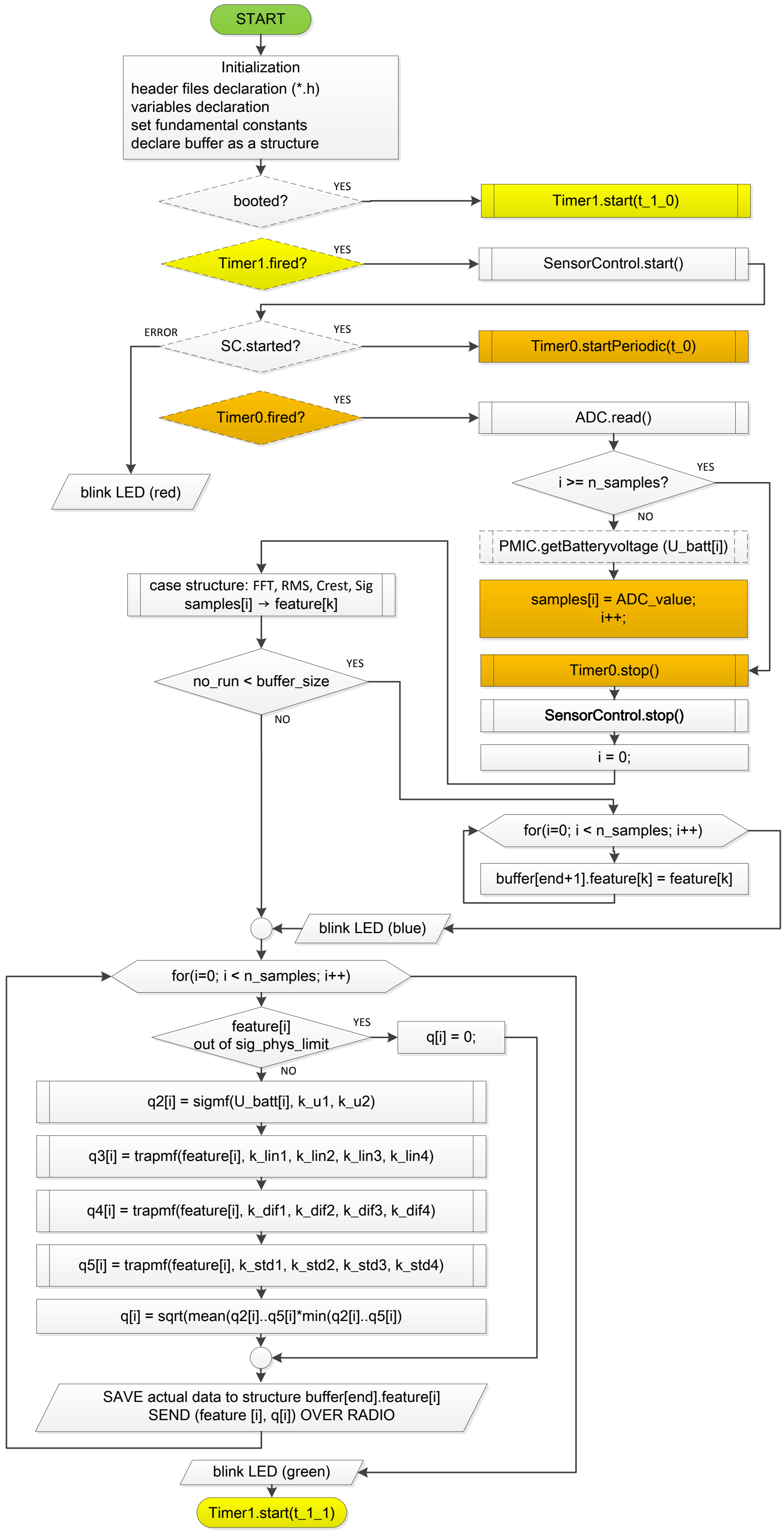

Figure 6.5: Sensor node flowchart in TinyOS

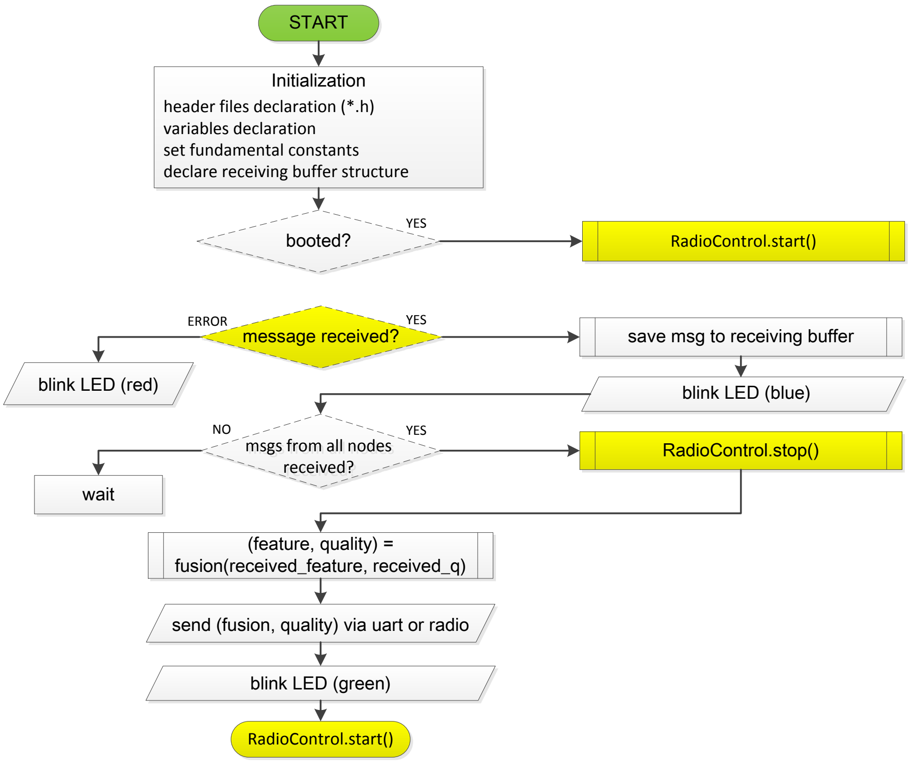

Figure 6.6: Fusion node flowchart in TinyOS

Chapter 7

Results

This chapter presents the results produced by techniques introduced in the previous chapters for verifying the proposed methods, mainly featuring the fusion section of a WSN-based MCM system. The model in Matlab was used to determine the values of the experimental variables and to verify the performance of each method. This verification was based on progressive modification of influencing factors. Results from a real test in the Imote2 WSN system are presented for comparison with relevant simulation results in Matlab.

7.1 Performance tests within the model in Matlab

7.1.1 Quality evaluation in the sensor node

To set the quality indicator according to outliers in the signal / feature waveform (see section 4.2.7), abrupt signal changes with high variation in amplitude have to be evaluated as low, while minor variations or gradually changing amplitude are evaluated as higher. All proposed methods are based on the "current" data point (i.e. the "future" data point from the point of view of the MEM buffer) mapping to the quality indicator by a membership function to gain a multi-value quality evaluation. We use a trapezoid as the membership function (see the section 5.1.3.

Artificial signals with some imperfections were prepared (see the red ringed blue line in Fig. 7.1) in order to check how well the proposed methods find outliers. All proposed formulas for computing the "future" sample and trapezoid parameters (summarized in table 5.1) were tested on signals of this type. The applicable methods and combinations of methods are as follows:

Computation of the expected position of the "future" sample:

- *last sample in MEM*: This is no computing method in fact. It is the simplest method for tracing data. If the last value is an outlier this method produces an invalid result.

- $mean(x)$: This is the arithmetic average of all values (x) in the MEM buffer. The computing demands of this method are low. Computation is based on a summation and a division. Essentially it is a low pass filter - the response is slow.

- $median(x)$: This method is the middle value from all arranged values (x) from the lowest value to the highest value in the MEM buffer. This method is not sensitive to outlier values. Computationally, this method is based on sorting. It is more demanding than mean computing but it produces very similar results.

- *linear regression* (y_reg): This method has acceptable computing demands based on multiplication, summation and division. The essential idea of this method is based on the linear regression line fitted across the MEM waveform records. If an abrupt change occurs this approach can produce an overshoot due to the interpolation by the linear line. Significantly, overshoots are created if the MEM buffer has not enough size. The method produces the best ratio in curve tracing performance and computing demands.

Computation of [a,b,c,d] parameters of the trapezoid:

- $k \cdot max(MEM)$: This is very simple to compute. It comprises the simplest detector for detecting an abrupt rising/falling edge (i.e. jumps, glitches etc.) in connection with the *last sample in MEM* method. A drawback of utilizing this method is that it widens the limits when a significant outlier comes to the MEM buffer. In this case the quality monitoring is too rough.

- $k \cdot peak(MEM) - diffmax(MEM)$: The function has opposite behaviour to that of $k \cdot max(MEM)$. The higher the maximal difference between neighbour values in *MEM*, the more strictly the limits are set. In an extreme case, the limits are set to zero. A simple computing method based on comparison.

- $k \cdot peak(MEM) - STD(MEM)$: This produces a smoother waveform than *diffmax(MEM)* because it utilizes a statistical index per whole *MEM* buffer record.

- $k \cdot \left|\frac{mean(MEM)}{peak(MEM)}\right|$: Essentially, this is the reciprocal of the crest factor (RMS is replaced by the mean), the limits are sometimes too strict.

- $k \cdot \frac{max(MEM)}{max(MEM)+STD(MEM)}$: This function produces the required trapezoid parameters successfully. The benefit is that the division by zero issue is reduced (i.e. if $STD(MEM) = 0$, this function is $\neq 0$). In addition, *STD(MEM)* can be substituted by other deviation factors, e.g.:

  - $diffmax(MEM)$: maximal difference between neighbour values in *MEM*,
  - $\left|1 - \frac{mean(MEM)}{median(MEM)}\right|$: a skewness detector,

  however, the results do not differ significantly among these deviation factors and a specific choice depends more on their computing demands.

Common characteristics:

- the buffer size must be correctly defined,

- the larger the MEM buffer size, the better the accuracy, but the more sluggish the response to a change

- linear regression is more agile in response to a change than other methods, but it produces overshoots if MEM has an improper size (usually too small)

- the *last sample in MEM* method detects abrupt changes, unlike methods based on the median, the mean or linear regression, which are more sensitive to trend divergence
- most statistical methods face the division by zero issue. It occurs in limit cases when a signal crosses 0, i.e. an invariable signal value for a long time. In such cases the program code has to replace a non-available value by another value, according to the desired function.

On the basis of the results, we engaged the edge detector and the trend monitor based on the *last sample in MEM* method (7.1) and the *linear regression* method (7.2), respectively (their interconnection is the green line in Fig. 7.1). The quality indicator serves satisfactorily for detecting imperfections.

$$q_{last}(i) = trapmf(s(i); a; b; c; d) \quad \text{(see 5.6)} \tag{7.1}$$

where *s(i)* is the "current" sample and parameter a is:

$$a = last\_sample(MEM) - k \cdot \frac{max(MEM)}{max(MEM) + STD(MEM)} \tag{7.1a}$$

parameter b is:

$$b = last\_sample(MEM) - l \cdot \frac{max(MEM)}{max(MEM) + STD(MEM)} \tag{7.1b}$$

parameter c is:

$$c = last\_sample(MEM) + l \cdot \frac{max(MEM)}{max(MEM) + STD(MEM)} \tag{7.1c}$$

parameter d is:

$$d = last\_sample(MEM) + k \cdot \frac{max(MEM)}{max(MEM) + STD(MEM)} \tag{7.1d}$$

$$q_{reg}(i) = trapmf(s(i); a; b; c; d) \quad \text{(see 5.6)} \tag{7.2}$$

where *s(i)* is the "current" sample and parameter a is:

$$a = y\_reg(MEM) - k \cdot \frac{max(MEM)}{max(MEM) + STD(MEM)} \tag{7.2a}$$

parameter b is:

$$b = y\_reg(MEM) - l \cdot \frac{max(MEM)}{max(MEM) + STD(MEM)} \tag{7.2b}$$

parameter c is:

$$c = y\_reg(MEM) + l \cdot \frac{max(MEM)}{max(MEM) + STD(MEM)} \tag{7.2c}$$

parameter d is:

$$d = y\_reg(MEM) + k \cdot \frac{max(MEM)}{max(MEM) + STD(MEM)} \tag{7.2d}$$

Coefficients k and l are experimentally determined with respect to signal amplitude. In our experiments $k = 1.5$ and $l = 0.3$.

Additionally, the Grubbs test (also referred to as the ESD method - Extreme Studentized Deviate) a well-known method for detecting outliers, was used for a comparison with the proposed method. If we know the population mean and STD from historical data, we can calculate the Z ratio (see eq. (7.3), i.e. a number which quantifies how far the outlier is from other values. Since 5% of the values in a Gaussian population are more than 1.96 standard deviations from the mean, we can consider that the outlier comes from a different population if Z is greater than 1.96 (see the violet line in Fig. 7.1). It can be seen that the proposed detector produces more accurate results than the Grubbs test.

$$Z = \frac{|mean - value|}{STD} \tag{7.3}$$

7.1.2 Evaluation of fusion algorithms

The structure of the tests for finding optimal values and dependencies of the algorithm variables was based on our own evaluation approach, which detects outliers using envelope thresholds, in addition to the typical Signal-to-Noise Ratio (SNR), computed as (7.4).

$$SNR = \left(\frac{A_{signal}}{A_{noise}}\right)^2 \tag{7.4}$$

While information fusion is a well investigated subject in the literature, there is no objective benchmark for a redundancy-based fusion algorithm comparison [82]. We used envelope thresholds around a simulated feature waveform (a summation of two sine waves 200 and 1000 Hz). Thus two-sided thresholds were created. The distance between the threshold and a “clear” waveform is given by a circle. The centers of the circles are placed at the waveform. The tangent points perpendicular to the waveform create the threshold (see Fig. 7.2). The diameter of the circles has to be set to a value such that authentic diagnostic information in the spectrum is maintained (in our case approximately 10% of the signal).

If the waveform crosses the higher threshold, we detect a type I error - false presence. The reverse case is a type II error - false absence, the waveform crosses the lower threshold. The frequency of occurrence of these errors is referred to as the false alarm rate (FAR).

The test evaluation is performed by comparing the average FAR of the raw data from the sensor with the FAR of the fusion output data. The simulation repeats a pseudo-random sequence with a given number of measurements. When this sequence is executed, the tested parameter is changed and the sequence is performed again (see section 6.1.4). The parameters of the pseudo-random signal sequence are gathered in table 7.1.

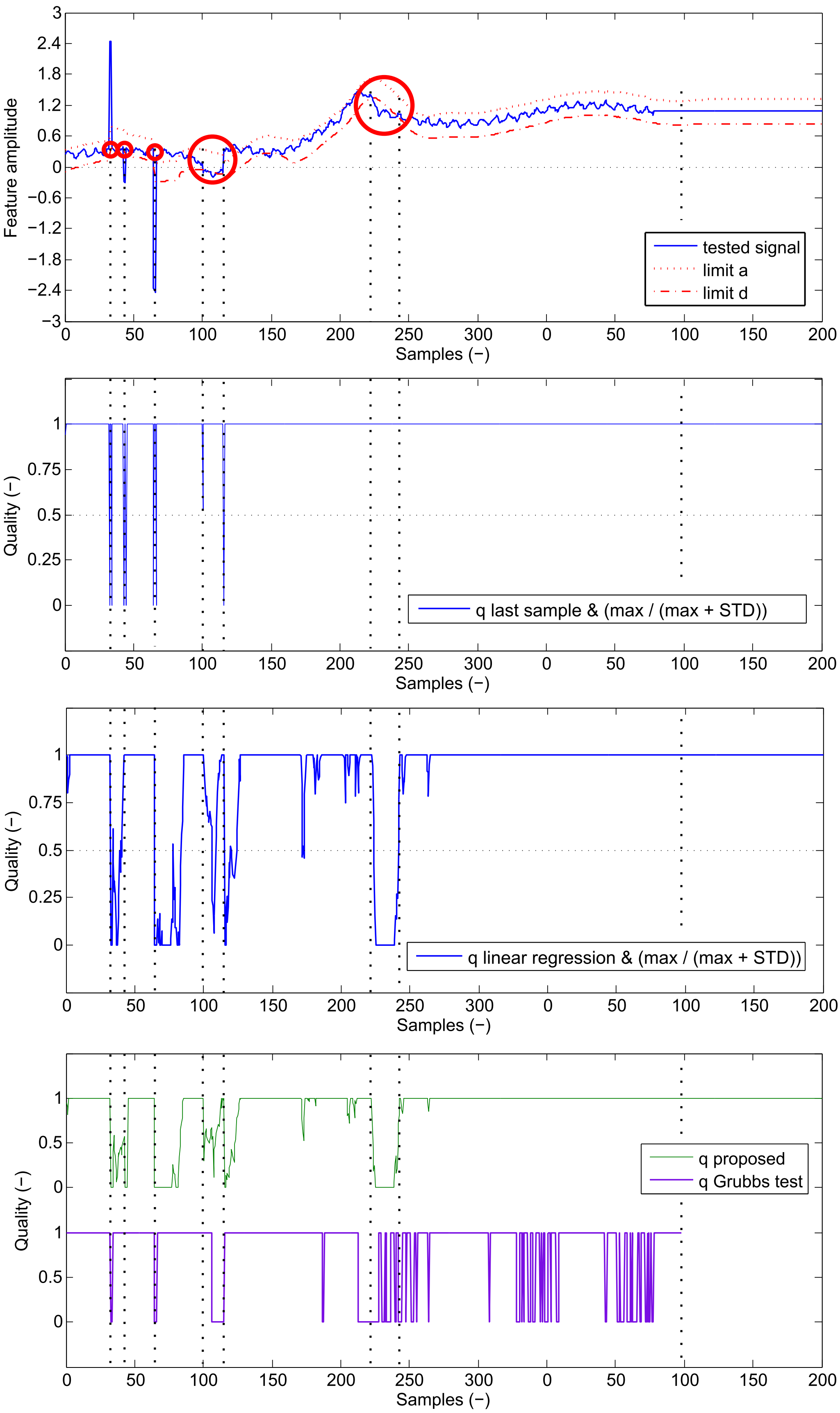

Figure 7.1: Comparison of quality indicators

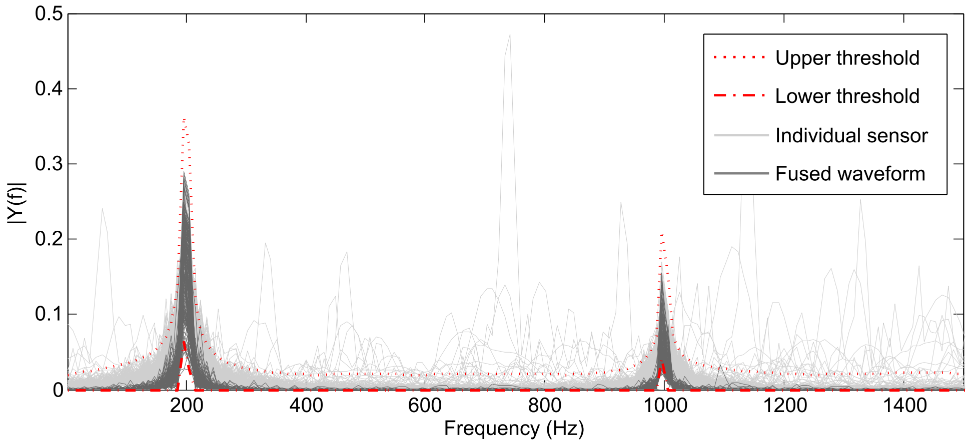

Figure 7.2: Envelope thresholds for determining the false alarm rate (FAR); frequency spectrum cutout - input signal: summation of two sinus waves (200 Hz and 1000 Hz), sampling rate 5 kHz, NFFT = 1024; upper limit cross mean FAR type I; lower limit cross mean FAR type II

Table 7.1: Pseudo-random signal sequence parameters

Parameter	**Value**			
MEM buffer	50 samples			
Number of sensors	3			
Length of sequence	150 measurements			
avg. SNR(-) (s1, s2, s3)	32.51	28.57	23.42	
FAR(-) t1/t2 (s1, s2, s3, avg.)	724/42	235/81	718/24	12.16

7.1.3 DST fusion settings and evaluation

According to the theory of the proposed DST fusion algorithm (see 5.2.1), data fusion performance is affected by two conditions. The first is the length of a segment in the input value range, i.e. the number of segments above which the fusion process constructs evidence to select the most credible segment as a partial result. The second condition is the standard deviation of normal distribution (σ), i.e. the shape of the Gaussian ("bell curve") constructed above each input value as a density function. Additionally, we propose input range segmentation based on the median of the relevant input values. This should ensure that the center of the final distribution function will not lie between two segments. This will improve the final precision due to the higher diversity among the segments. The impacts of all these factors were evaluated in the Matlab model by the testing sequence described above. We used 150 gradually changing steps of the tested value.

Input value range segmentation with σ as a parameter

It is evident from the DST fusion scheme that an excessive number of segments increases the computing time. Unlike the situation when few segments are used, fusion works improperly. We start the evaluation of a proper number of segments with parameter σ at 1, and we decrease it to obtain the best results (see Fig. 7.3). The time demand of the DST fusion algorithm is depicted in Fig. 7.4.

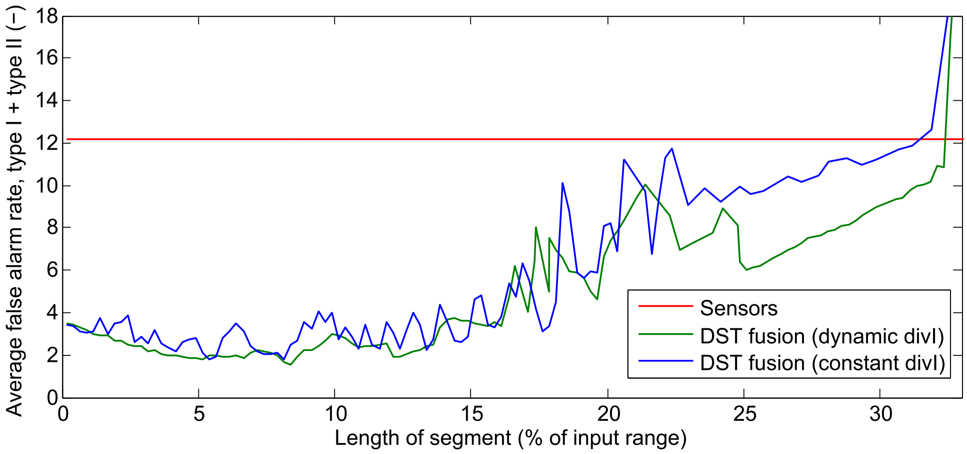

Figure 7.3: Dependance of FAR on segment size; red line - average FAR from sensors; blue line - DST fusion with constant segmentation; green line - DST fusion with dynamic segmentation

Dynamic value range segmentation by the median

The difference between the green line and the blue line in Fig. 7.3 shows that the proposed dynamic value range segmentation is able to eliminate sharp edges at two segment transitions. The increase in computing time is almost negligible, see Fig. 7.4.

Through these experiments, we are able to achieve average FAR (type I and type II together) 0.86, while the average FAR from the sensors is 12.16 per one measurement. The signal to noise ratio is 56.6. The DST fusion algorithm produces theses results when sigma is 0.06 and a segment is 5% of the full input range. One fusion iteration (i.e. one resultant sample produced by the DST fusion of samples

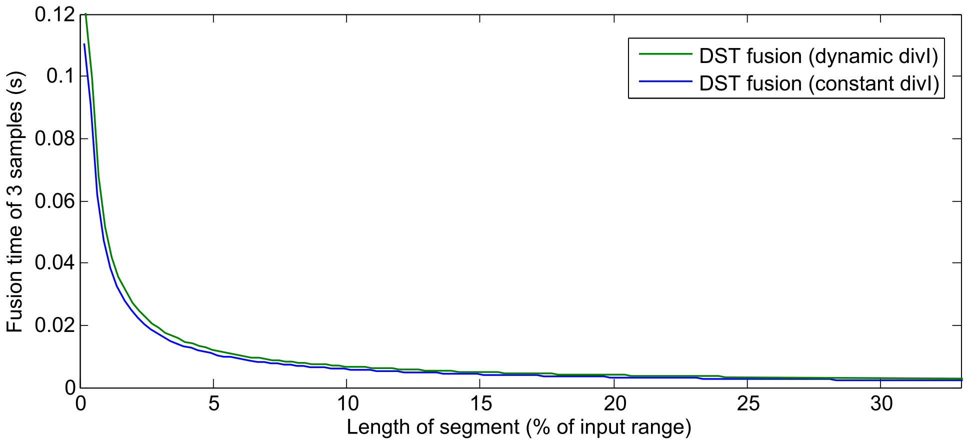

Figure 7.4: Dependence of computing time consumption on size of segment

from 3 sensor nodes) takes 0.012 seconds.

7.1.4 Fuzzy logic-based fusion settings and evaluation

Fuzzy logic-based fusion is determined by rules, by the type of inner method that is used (fuzzification, defuzzification etc.) and the membership function settings. The rules are invariable, i.e. one setup, one setting and also the membership function settings for each input have to be the same. The type of membership functions and the inner computing method can be changed. The best results were obtained with the configuration presented in table 7.2. Average FAR = 1.64, SNR = 65.2, and one fusion iteration takes 0.004 seconds if the radical approach is used. With average FAR = 1.91, SNR = 60.4, the time consumption is the same if the moderate approach is used. The moderate/radical difference is defined in section 5.2.2.

Table 7.2: Optimal fuzzy-logic based fusion settings

Part of fuzzy system	**Computing method**
And method	product
Or method	not used
Implication	min
Aggregation	sum
Defuzzification	mom

7.2 Performance tests using real WSN hardware - Imote2

7.2.1 Simple point-to-point data transfer

Firstly, we used direct transfer between the sensor and the gateway node, without a middle node (simplest WSN structure, see section 4.2.1). Imote2 WSN nodes enable sampling up to 5 kHz by integrated 12-bit ADC without any adaptation of the HW. These parameters are adequate for many MCM applications (e.g. detection of imbalance, misalignment, etc.) For example, if we acquire one vibration signal per 2 seconds we get 10000 samples. Feature extraction significantly reduces this number. RMS and the crest factor produces one figure per selected period (e.g. one figure per one second of sampling). For FFT with N selected to 1024 (a compromise between spectrum resolution and computing demands), the output data will be reduced to $(N/2) + 1$, i.e. 513 samples.

Although WSN systems of this kind are already produced, we had to evaluate the fundamental cornerstone of the proposed fusion system, i.e. data transfer between sensor node and sink node. This means that the feature extraction and quality computation program codes work properly according to the Matlab simulation results. The evaluation was performed manually using a programmable signal generator (Agilent 33220A) and an oscilloscope (Tektronix MSO2014) with the FFT function. Partial degradations were arranged by fine changes in power supply, using a potentiometer, and RF attenuation by varying the distance and the shielding of the nodes. Fig. 7.5 compares the RMS spectrum of the input signal computed by the oscilloscope and the output amplitude spectrum produced by the node.

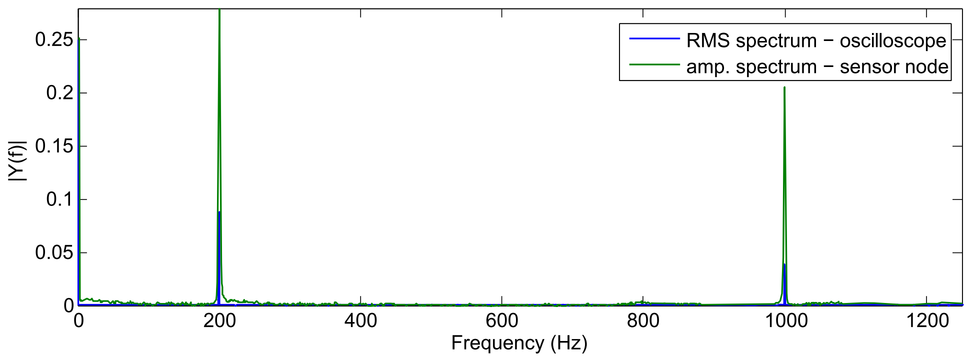

Figure 7.5: Frequency spectrum comparison (a cutout) - input signal: summation of two sinus waves (200 Hz and 1000 Hz), sampling rate 5 kHz, NFFT = 1024; blue line - amplitude spectrum (RMS) of the input signal to the sensor node (computed by MSO2014 oscilloscope); green line - amplitude spectrum computed by the sensor node

7.2.2 Data compression

To evaluate the proposed data compression method, we prepared a measurement on a mechanical vibrodiagnostic simulator to obtain a real signal (see Fig. 7.6). This signal was used as an example to find coefficient k in the proposed RMS threshold compression method.

We experimentally defined $k = 0.007$. Although we send approximately 30% of the original number of FFT samples, the main frequency bins rest and the diagnostic information is transferred further.

The second approach for setting the threshold mentioned in section 5.1.2 is based on expert device identification. For example, a vibrodiagnostic simulator is a simple mechanical device - a shaft driven by a DC brushless motor. The shaft is mounted via two ball bearings, and it is connected with the motor by a jaws coupling. If we want to monitor the health of this device, we need to check that the frequency bin corresponds to each individual component. Firstly, this involves computing the rotating frequency, and then this value determines the frequency bins of all other

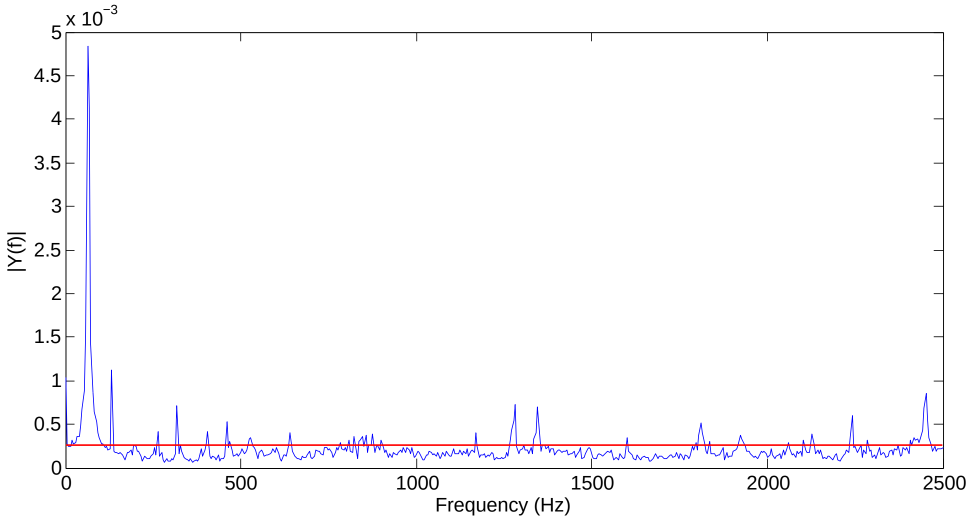

Figure 7.6: Data compression threshold setting (red line)

components. Fig. 7.6 shows the rotating frequency at 66 Hz (i.e. 4000 RPM) and the ball bearing frequencies correspond to the outer and inner race (327 and 476 Hz). The threshold value is then set on the basis of expert knowledge. The value is usually set according to the condition of a new or fault-free device.

7.2.3 Evaluation of fusion algorithms

National Instruments measuring and testing equipment

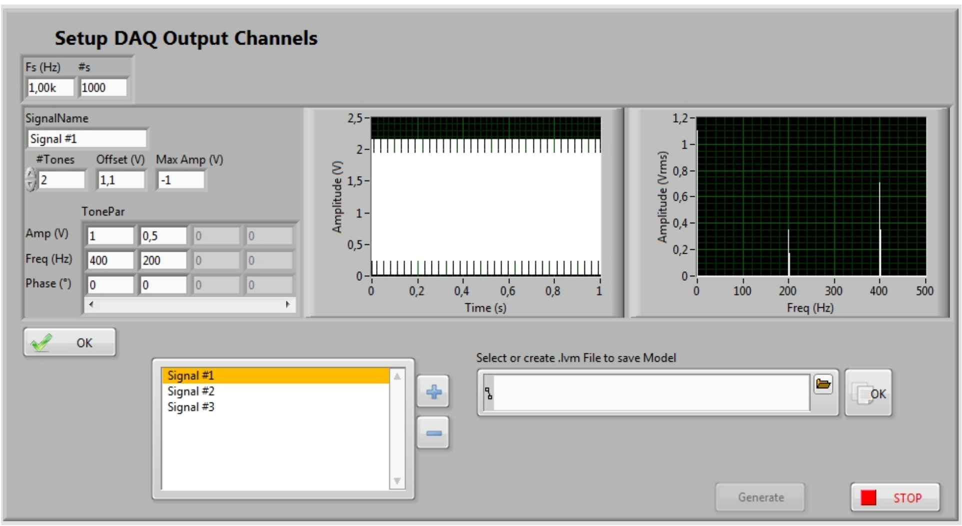

Figure 7.7: NI control panel of our own application to fusion node evaluation

NI measurement equipment was used for evaluating the program codes (see section 6.2.2). A control application in LabView was created to set the required signals to feed the sensor node inputs, similarly as for the Matlab evaluation. The signal is created by summation of the sinusoids, or it can be loaded from a file. Some artificial imperfections can be added to the signal, e.g. amplitude changes, breakdowns, steps, jumps etc.) The results are summarized in table 7.3.

Table 7.3: Fusion node evaluation by artificial signals

Parameter	Value
MEM buffer	50 samples
Number of sensors	3
Number of measurements	150
avg. SNR(-) (s1, s2, s3)	31.51 36.21 33.42
FAR(-) t1/t2 (s1, s2, s3, avg.)	338/110 811/113 541/84 13.31
avg. SNR(-) DST fusion	42.29
FAR(-)	2.12
time consumption per iteration (s)	0.05
avg. SNR(-) fuzzy-logic fusion (radical)	49.54
FAR(-)	3.34
time consumption per iteration (s)	0.021
avg. SNR(-) fuzzy-logic fusion (moderate)	47.15
FAR(-)	3.48
time consumption per iteration (s)	0.021

Vibrodiagnostic simulator

The "real" test is based on data from a mechanical vibrodiagnostics simulator, see Fig 7.8.

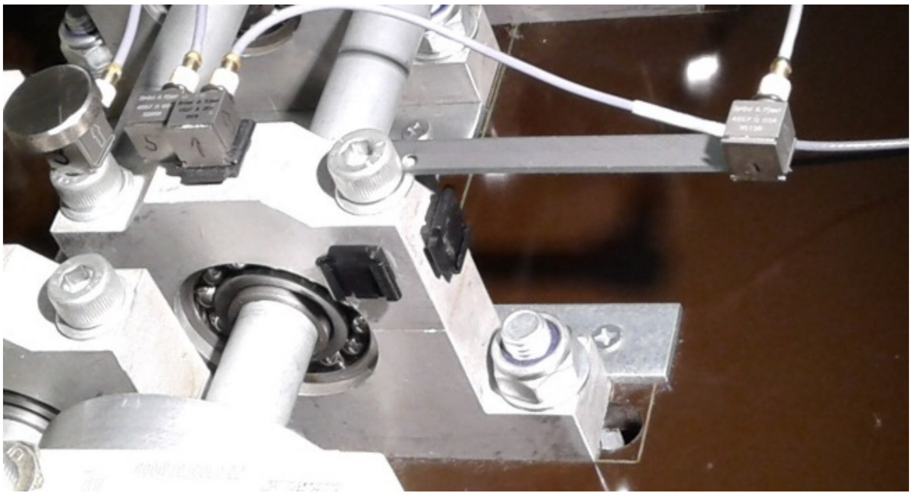

Figure 7.8: Placement of the accelerometers in the vibrodiagnostics simulator; accelerometer fitting from the left - freely placed (weighted by a small weight), fixed by two-sided tape, correctly fitted by a mounting clip as a reference, fitted via a hardened metal strip

We arranged the measurement for one run of the simulator at 4000 RPM:

- monitored place - the z axis (Cartesian coordinates) of bearing house no. 4,
- four BK 4507 accelerometers, one as a reference, the rest for each sensor node,
- accelerometers connected via a Nexus 2693 conditioning amplifier, amplifier set as band pass from 0.1 Hz to 1000 Hz,

- reference accelerometer connected via NI card 6211 to a PC, sampling rate 5 kHz,
- reference accelerometer correctly mounted via UA 1475 mounting clips,
- sensor node accelerometers mounted in a "sloppy" manner to create incorrect measurements,
- 50 measurements were created, one measurement record takes 1 second, sampling rate of the sensor nodes 5 kHz, 4000 RPM means more than 60 periods acquired per one second,

The comparison between reference and fusion output was made by the standard deviation at each sample position over all 50 measurements. The results are depicted in Fig 7.9 and summarized in table 7.4.

Table 7.4: Evaluation of the fusion node by real vibrodiagnostic signals

Parameter	Value		
MEM buffer	50 samples		
Number of sensors	3		
Number of measurement	150		
std (reference)	8.58×10^{-5}		
std (s1, s2, s3)	2.2×10^{-4}	2.9×10^{-4}	2.6×10^{-4}
std DST fusion	6.12×10^{-5}		
std fuzzy-logic fusion (radical)	5.17×10^{-5}		
std fuzzy-logic fusion (moderate)	5.84×10^{-5}		

7.3 Summary of results

The achievements of this doctoral project are well summarized in Fig. 7.10. The ball bearing housing is measured by three WSN sensor nodes (accelerometers). All sensors are mounted at one place in redundancy fashion. All sensors make improper measurements of physical phenomena (i.e. vibration) to simulate imperfections in the wireless measurement chain. This means that the redundant sensors are improperly mounted, temporarily terminated, overloaded, etc. Raw signals, in the top row in the figure, demonstrate invalid measurements in the time-domain. WSN technology is limited in bandwidth (250 kbps), which means that it is not feasible to acquire raw data samples and transfer them to a central unit to analyse all of them. Therefore each sensor node extracts a signal feature (the middle row of the figure). In this case, it is an amplitude spectrum, where the data are reduced whereas MCM useful information is maintained. However, imperfection of individual measurements persists. This is solved by the next level of the nodes, where the sensor nodes transfer their data. We have called this level the feature fusion level, where the data are aggregated. The output of the feature fusion level node is demonstrated in the bottom row of the figure. We have proposed two independent aggregation methods

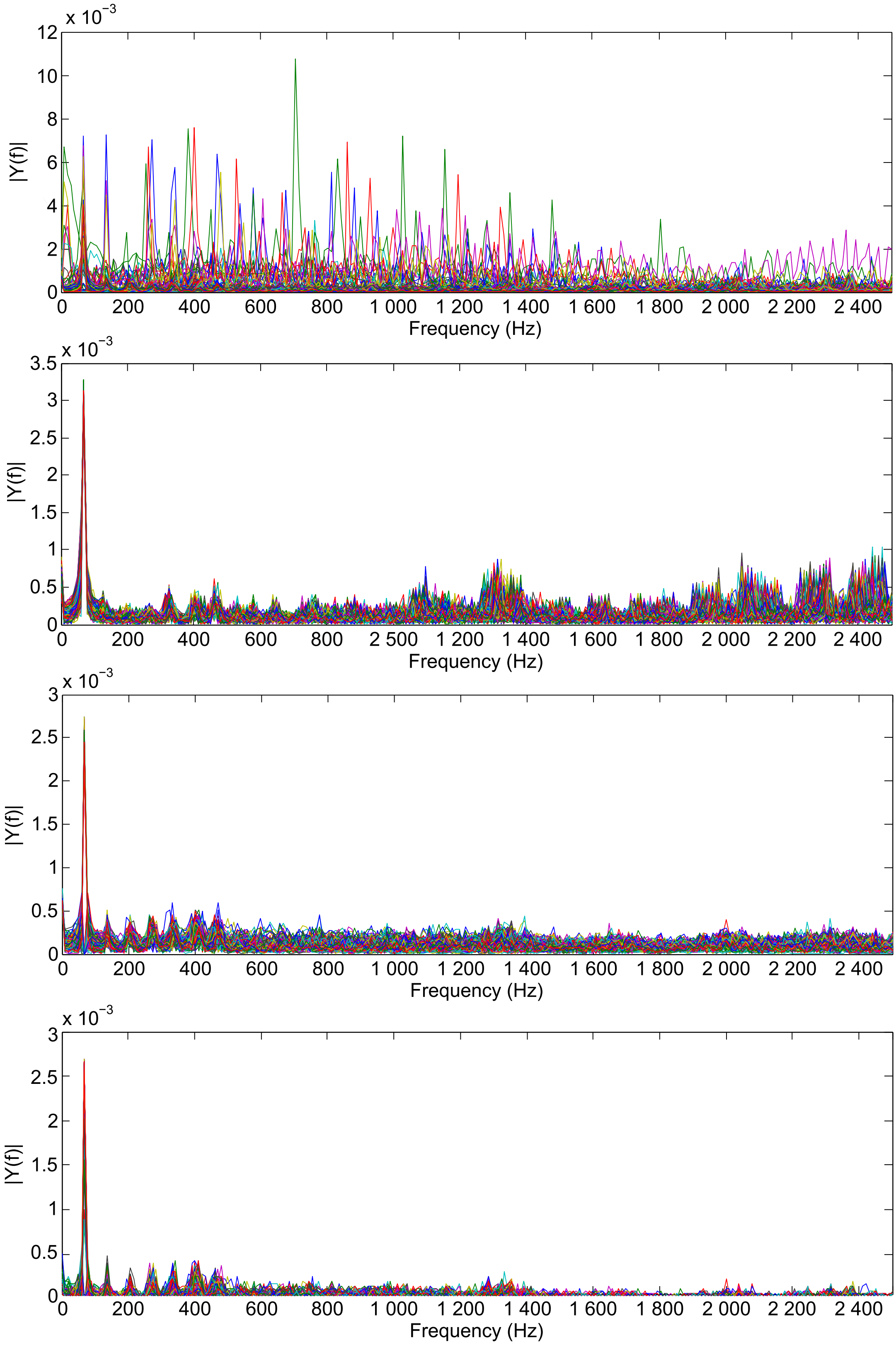

Figure 7.9: Fusion efficiency - real measurement; the graphs demonstrate amplitude spectra measured during one run of the vibrodiagnostic simulator device: top - all improperly mounted sensors, second - reference, third - DST fusion, bottom - fuzzy-logic fusion

based on information fusion. The first is based on Dempster-Shafer theory (we refer to it as DST fusion), while the second is based on fuzzy-logic (we refer to it as fuzzy-logic fusion).

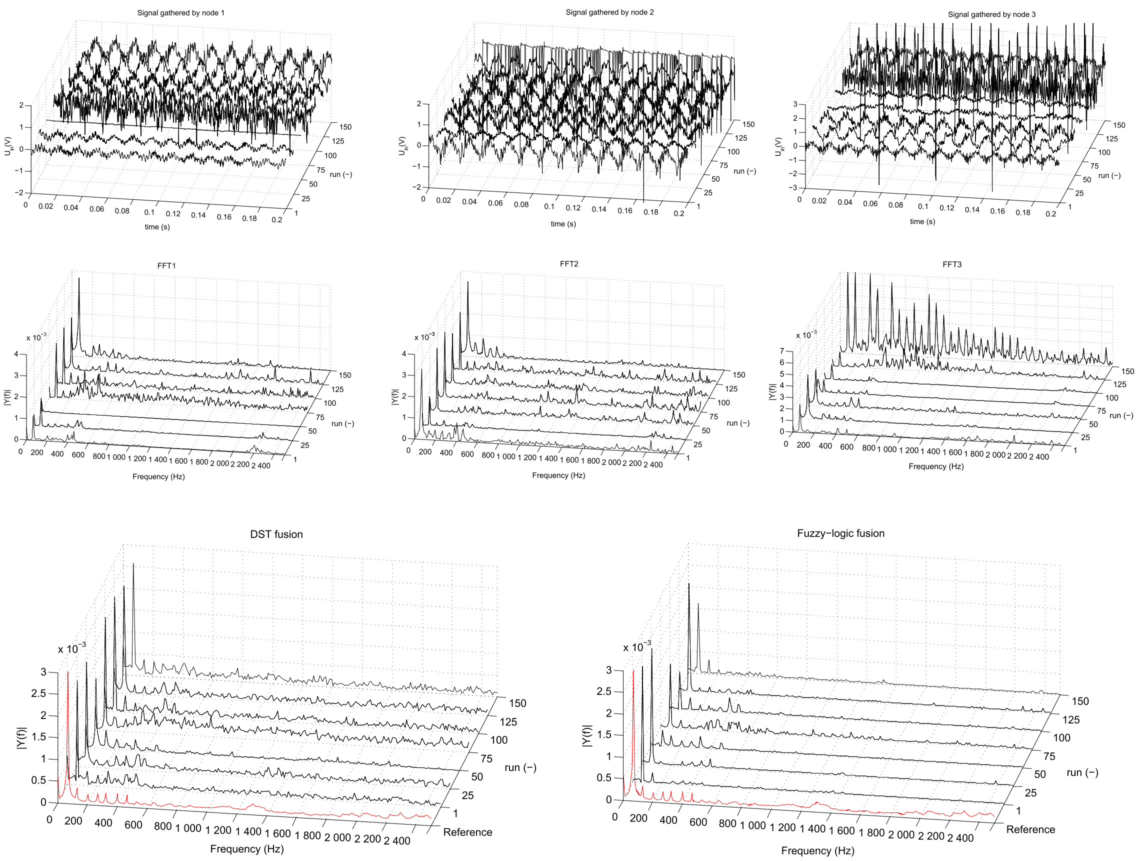

Figure 7.10: Data fusion from three sensor nodes (top row – degraded time-domain signals on the input of the nodes, middle row – amplitude spectrum produced by the nodes, bottom – fused spectrum produced by the fusion node. The red line is the reference from a sensor connected by wire).

Both algorithms making information fusion are driven by a quality indicator produced by the sensor nodes according to the validity of the acquired data and the health of the sensor node. This concept was first published by Hermans et al. [38] to improve temperature monitoring. We adopted this idea for signals using high sampling rates, and we improve it for signal feature extraction and quality indicator heuristics. We have introduced a new fuzzy-logic based fusion algorithm. All proposed methods have been verified via simulation and via real WSN experiments numerically summarized in table 7.5.

All experiments summarized in table 7.5 demonstrated rapid signal improvement when multi-sensor fusion was applied. In the case of the first two experiments, where an artificial signal with artificial imperfections (cut off, steps, jumps, overloading etc.) was used, the best result was for the DST fusion method when FAR evaluation was engaged. In the case of the SNR evaluation, however, both fuzzy-logic approaches had better results. The better fuzzy-logic SNR results were due to the extremely low level of amplitude outside the effective frequency bins because of signal normalizing between 0 and 1, followed by mapping logic where the low amplitude levels are near to zero and vice versa. The FAR results are of greater importance for further feature analysis. They indicate whether correct information

Table 7.5: Summary of results

	SNR(-), FAR(-), std		
	Simulation	**Simulation HW**	**Real test**
Raw feature data (average)	28.16, 12.16, -	33.71, 13.31, -	- , - , 2.57×10^{-4}
Fuzzy-logic fusion (moderate)	60.4, 1.91, -	47.15, 3.48, -	- , - , 5.84×10^{-5}
Fuzzy-logic fusion (radical)	61.2, 1.64, -	49.54, 3.34, -	- , - , 5.17×10^{-5}
DST fusion	56.6, 0.86, -	42.29, 2.12, -	- , - , 6.12×10^{-5}

is present or not. By optimizing the DST fusion parameters (i.e. sigma and length of segment) we achieve a very low value of error occurrences in the features that are produced.

Real signals / features were compared by the average standard deviation of each sample position over 150 waveforms. The improvement ratio (approximately five times) is very similar to the results from the simulations.

Algorithmic implementation of the proposed fusion approaches is simple and computationally inexpensive for use in wireless sensor network nodes. Both algorithms take a similar amount of memory - approximately 30 KiB of ROM without optimization (the sensor node code takes 50 KiB of ROM). DST fusion parameters should be adapted to the measured signal / features. Fuzzy-logic based fusion is more general-purpose, and mainly requires a definition of the logic rule strictness to define what type of input is acceptable to transfer as output. However, an improper setting of the rules or another part of the fuzzy-logic algorithm leads to false results. DST fusion is more robust to an improper setting of the input parameters.

Both proposed algorithms evaluate their own results. Information about the quality of the signal and the condition of the network can be transferred to a further network level. The presence of a quality indicator in the data fusion node output also enables redundancy at this level. Thus very reliable wireless systems are feasible, although high sampled signals are monitored.

Finally, we attempted to increase the number of sensor nodes to obtain a DST fused spectrum without any error (within both thresholds). This situation for artificial signal sequence 7.1 occurred when 8 sensors nodes were engaged. Obviously, the more redundant the sensors were, the higher the resistance to disturbances and imperfections in the transferred signal. The experiments carried out in this thesis engaged three sensor nodes in redundant fashion. Both proposed fusion algorithms produce valuable results, while the complexity of the system remains reasonable.

The number of redundant sensor nodes that is used should be a compromise between robustness and network size, based the demands for particular applications. The main limitation of WSN multi-sensor fusion system proposals, which has to be taking into account, is the bandwidth of the IEEE 802.15.4 systems.

Chapter 8

Conclusion

This thesis has investigated WSN performance in terms of reliability, with a view to adopting this technology as a cornerstone of the wireless early warning monitoring system. The selected application area is the aircraft industry, particularly condition monitoring of the power plant and accessories of light aircraft. Our work has improved the reliability by introducing multi-sensor fusion as an HW-independent method complementary to the networking in the bottom OSI layers.

8.1 Summary

This thesis has worked on feature level fusion in the WSN-based early warning monitoring MCM system to improve reliability. Feature level fusion deals with multi-sensor fusion, where a group of sensor nodes measure the same physical phenomena at the same place in redundant fashion. Instead of raw data, the sensor nodes transfer extracted features such as diagnostic information contained in the measured signal. Together with the features, each sensor node produces an uncertainty evaluation of the transmitted information (referred to as quality). The sensor node data are received by a fusion node, where the data are aggregated to the most appropriate result with respect to the quality of the received feature. The fusion node also evaluates its output by the quality indicator.

The design described here has been verified by means of simulation and real WSN experiments. For three independent nodes sensing the same randomly degraded signal, the improvement in SNR is higher when fuzzy fusion is applied. The radical approach produces slightly better results than the moderate approach. The improvement is almost 1.5 $\times$, due to the extremely low level of amplitude outside the effective frequency bins. Fuzzy fusion consumes less than 50% of the computing time for DST fusion. However, DST fusion is more efficient for preventing false alarms (in our experiment 6 $\times$ fewer errors on an average). Moreover, DST fusion does not contain inner variables (rules, membership function settings, etc.) that strongly influence the results, as in fuzzy-logic based fusion. DST fusion is influenced by the length of the segment and by the sigma value, but this method produces appropriate results in a relatively broad range of value settings between the apparent limits.

8.2 Accomplishment of the aims of the thesis

The aim of our project was to propose a quality-based data fusion approach to propagate maximal information content picked up from a device by a wireless sensor network to a sink node, while retaining reasonable system complexity. This objective was addressed in the following way:

WSN-based MCM system design

- making a review of the literature on aircraft MCM techniques and WSN technology, summarising key issues for the establishment of a WSN-based early warning monitoring system,
- establishing an essential system design, taking into account the character of the diagnostic signals, mainly vibrations, acquired from various mechanical devices by conventional wire systems,
- applying distributed signal processing methods, mainly information fusion, see Fig. 4.1,
- focusing on the feature fusion section of the scheme,
- designing the structure of a sensor node and a fusion node,

Improving WSN reliability

- utilizing redundancy - an information fusion scheme referred to as multi-sensor fusion, as developed in the thesis,
- proposing a quality indicator to drive the fusion based on signal imperfections and the health of the acting WSN node,
- composing a suitable heuristics for finding signal imperfections, i.e. for identifying samples varying in comparison with previous records,
- establishing a fusion algorithm based on Dempster-Shafer theory able to produce data aggregation and quality evaluation of fusion,
- proposing a new fusion algorithm based on fuzzy logic in addition to the DST fusion algorithm,

WSN bandwidth savings

- proposing an asynchronous monitoring method that applies defined time delay or event-based wake-up,
- dealing with an enormous amount of raw data in the case of signal monitoring with a high sample rate, using:
    - extraction of features from the signal,
    - an FFT spectrum compression method executed by a threshold limit driven by the RMS value,

Verification

- simulating the feature fusion WSN level in Matlab,
- evaluating sensor node heuristics that achieve the required performance,

- establishing a False Alarm Rate (FAR) for evaluating the correctness of the data,
- examining DST and fuzzy-logic based fusion algorithms influencing the parameters,
- optimizing factors influencing feature fusion,

Performance evaluation

- arranging the proposed multi-sensor fusion system by implementation into real Crossbow Imote2 WSN hardware,
- conducting an experimental test showing the proper functionality and feasibility of the proposed multi-sensor fusion in the WSN system.

The results of this thesis have been published in two international journals *IEEE Transactions on Industrial Electronics*, *IEEE Transactions on Industrial Informatics*, and presented at three international refereed conferences. In addition, some partial results have been presented locally, see the list of publications.

8.3 Future work

The weakest point of this thesis is the absence of a large-scale validation campaign. The results of the proposed methods have been shown only in the form of case studies on artificial signals and on one real device. This problem is due to the difficulty in obtaining a large amount of real data.

To complete the whole monitoring system, it is necessary to work on the decision fusion part. This will involve proposing a suitable fault classifier working in a constrained embedded system, as is required for WSN nodes.

Classification methods should be deployed in two places in the decision level stage of the WSN monitoring system. One place is the feature fusion output, where a partial decision should be made, and the other is the decision fusion output, where the final decision will be produced. Based on the features obtained in the feature level stage of the system, there are several suitable classification techniques. When the features are based on the model of the monitored systems it is suitable to use simple thresholds. However when the features are more complex, there is a need to train a classifier. There is usually no information about faulty states of a monitored subject; therefore it is necessary to use a special kind of classification technique called one-class classification. In one-class classification, the classifier model is trained to recognize a certain behavior of a monitored subject that is different from normal operation.

Classification methods suitable for WSN systems need to be elaborated similarly as information fusion in this thesis. We believe that the results of this thesis will be helpful for future investigators, since all fusion methods proposed here are able to work with partial decisions as well as with signal features.

List of Publications

Work and publications related to the thesis

International Journals

[A1] O. Kreibich, J. Neuzil, and R. Smid, "Quality-based multiple sensor fusion in an industrial wireless sensor network for mcm," *Industrial Electronics, IEEE Transactions on*, 2013. online access.

[A2] J. Neuzil, O. Kreibich, and R. Smid, "A distributed fault detection system based on iwsn for machine condition monitoring," *Industrial Informatics, IEEE Transactions on*, 2013. online access.

International Conference Proceedings

[B1] O. Kreibich, J. Neuzil, and R. Smid, "Application of wireless sensor networks in condition monitoring of rotating devices," *CM 2012 / MFPT 2012 - The 9th International Conference on Condition Monitoring and Machinery Failure Prevention Technologies*, London, UK, 2012.

[B2] J. Neuzil, O. Kreibich, and R. Smid, "Advanced signal processing in wireless sensor networks for MCM," *CM 2012 / MFPT 2012 - The 9th International Conference on Condition Monitoring and Machinery Failure Prevention Technologies*, London, UK, 2012.

[B3] J. Neuzil, R. Smid, and O. Kreibich, "Rotary machine condition monitoring using one-class classification in wireless sensor networks," *CM 2011 / MFPT 2011 - The 8th International Conference on Condition Monitoring and Machinery Failure Prevention Technologies*, Cardiff, UK, 2011.

[B4] R. Smid and O. Kreibich, "Information fusion in wireless sensor networks for MCM," *CM 2011 / MFPT 2011 - The 8th International Conference on Condition Monitoring and Machinery Failure Prevention Technologies*, Cardiff, UK, 2011.

[B5] J. Neuzil, R. Smid, and O. Kreibich, "Distributed classification in wireless sensor networks for machine condition monitoring," *CM 2010 / MFPT 2010 - The 7th International Conference on Condition Monitoring and Machinery Failure Prevention Technologies*, Stratford-upon-Avon, UK, 2010.

Local Publications

[C1] O. Kreibich, "Imote2 installation procedures within WSN based MCM project," tech. rep., CTU in Prague, 2013.

[C2] 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, 2013.

[C3] O. Kreibich and J. Neuzil, "Wireless sensor networks (in Czech)," *MM Prumyslove spektrum*, vol. 11, pp. 38–39, 2011.

[C4] J. Neuzil and O. Kreibich, "Wireless sensor networks for machine condition monitoring," *POSTER 2010 - Proceedings of the 14th International Conference on Electrical Engineering*, Prague, CZ, 2010.

[C5] O. Kreibich, "Wireless sensor network in vibro-diagnostics," *POSTER 2009 - Proceedings of the 13th International Conference on Electrical Engineering*, Prague, CZ, 2009.

[C6] O. Kreibich, "Vibration simulator for experiments in vibro-diagnostic," *POSTER 2007 - Proceedings of the 12th International Conference on Electrical Engineering*, Prague, CZ, 2007.

Other Work

[D1] "Safe and safety elements in aerospace and space technology," Member of the research team of a project no. SGS12/193/OHK3/3T/13, funded by the Student Grant Agency of CTU in Prague. 2013 - 2014.

[D2] "Advanced methods of data and signal processing for diagnostics," Member of the research team of a project no. SGS12/155/OHK3/2T/13, funded by the Student Grant Agency of CTU in Prague. 2012 - 2013.

[D3] "Digitization, synchronisation and signal processing in sensors and sensor networks," Member of the research team of a project no. SGS10/207/OHK3/2T/13, funded by the Student Grant Agency of CTU in Prague. 2010 - 2011.

[D4] "Sensors and intelligent sensor systems," Member of the research team of a project no. 102/09/H082, funded by the Czech Grant Agency. 2009 - 2011

[D5] "An enhancing of vibrodiagnostics education - wireless approach into vibrodiagnostics," Principal investigator of a project no. G1 1901/2009, funded by FRVS. 2009

[D6] O. Kreibich and R. Smid, "Control module for a vibrodiagnostic simulator device," prototype. 2010.

[D7] O. Kreibich and R. Smid, "Power control unit for a DC motor," prototype. 2008.

[D8] O. Kreibich and R. Smid, "Vibration simulator for experiments in vibrodiagnostics," prototype. 2008.

Work and publications not directly related to the thesis

International Journals

[E1] 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*, vol. 58, no. 5, pp. 1027–1036, 2011.

[E2] M. Kubinyi, O. Kreibich, J. Neuzil, and R. Smid, "Novel S-transform information fusion for filtering ultrasonic pulse-echo signals," *Przeglad Elektrotechniczny*, vol. 87, no. 1, pp. 290–295, 2011.

International Conference Proceedings

[F1] O. Kreibich and R. Smid, "E-learning tools for education and training in diagnostics and machine condition monitoring," in *Proceedings of the 2nd International Conference on Computer Supported Education - CSEDU 2010*, Valencia, Spain, 2010.

[F2] M. Kubinyi, M. Reinstein, and O. Kreibich, "GNS ambiguity resolution techniques," *CUAS 2008 - Conference Proceedings*, Prague, CZ, 2008.

Local Publications

[G1] L. Tomes, P. Malcik, J. Skopal, J. Curdova, J. Senk, and O. Kreibich, *Popularization Technical Standards in Industry, Bussiness and Education (PRTN III) (in Czech).* Czech Office for Standards, Metrology and Testing, 2009.

Other Work

[H1] O. Kreibich, R. Smid, and V. Nadvornik, "Power driver of inductive load for electromechanical vibration transducer," prototype. 2008.

References

[1] "Aviation investigation report a10o0101," tech. rep., Transportation Safety Board of Canada, May 2010.

[2] H. Bai, M. Atiquzzaman, and D. Lilja, "Wireless sensor network for aircraft health monitoring," *Broadband Networks, 2004. BroadNets 2004. Proceedings. First International Conference on*, pp. 748–750, 2004.

[3] "Annual safety review 2009," tech. rep., European Aviation Safety Agency, 2009.

[4] "Annual safety review 2010," tech. rep., European Aviation Safety Agency, 2010.

[5] "Transport safety performance in the EU - a statistical overview," tech. rep., European Transport Safety Council, 2003.

[6] A. Riley, B. Aardema, P. Vosbury, M. A. Eiff, H. Frautschy, R. Serkenburg, D. Shaffer, T. Wild, and T. Michmerhuizen, *Aviation Maintenance Technician Handbook*. U.S. Department of Transportation: Federal Aviation Administration, 2008.

[7] J. Knezevic, *Systems Maintainability*. Systems effectiveness, Springer, 1997.

[8] Airbus, *AMM Engine vibration indicating*, training manual 77-32-00 ed., 2010.

[9] A. Linke-Diesinger, *Systems of Commercial Turbofan Engines: An Introduction to Systems Functions*. Springer-Verlag Berlin Heidelberg, 2008.

[10] C. Scheffer and P. Girdhar, *Practical Machinery Vibration Analysis and Predictive Maintenance*. Practical Machinery Vibration Analysis and Predictive Maintenance, Elsevier Science, 2004.

[11] G. O. Chandroth and N. E. Sharkey, "Cylinder Pressures and Vibration in Internal Combustion Engine Condition Monitoring," *in Proceedings 'Comadem 99*, pp. 294–297, 1999.

[12] B. Lu and V. C. Gungor, "Online and Remote Motor Energy Monitoring and Fault Diagnostics Using Wireless Sensor Networks," *IEEE Transactions on Industrial Electronics*, vol. 56, pp. 4651–4659, Nov. 2009.

[13] X. Xue, V. Sundararajan, and W. Brithinee, "The application of wireless sensor networks for condition monitoring in three-phase induction motors," *Electrical Insulation Conference and Electrical Manufacturing Expo, 2007*, pp. 445–448, 2007.

[14] D. Miorandi, E. Uhlemann, S. Vitturi, and A. Willig, "Guest Editorial: Special Section on Wireless Technologies in Factory and Industrial Automation, Part I," *IEEE Transactions on Industrial Informatics*, vol. 3, pp. 95–98, May 2007.

[15] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Commun. ACM*, vol. 43, no. 5, pp. 51–58, 2000.

[16] H. G. Chiwewe, T.M., "A Distributed Topology Control Technique for Low Interference and Energy Efficiency in Wireless Sensor Networks," *IEEE Transactions on Industrial Informatics*, vol. 8, pp. 11–19, Feb. 2012.

[17] P. T. A. Quang and Dong-Sung Kim, "Enhancing Real-Time Delivery of Gradient Routing for Industrial Wireless Sensor Networks," *IEEE Transactions on Industrial Informatics*, vol. 8, pp. 61–68, Feb. 2012.

[18] L. Ferrigno, V. Paciello, and A. Pietrosanto, "Experimental Characterization of Synchronization Protocols for Instrument Wireless Interface," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, pp. 1037–1046, Mar. 2011.

[19] J. Jang, D. Berdy, J. Lee, D. Peroulis, and B. Jung, "A Wireless Condition Monitoring System Powered by a Sub-100 μW Vibration Energy Harvester," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, pp. 1082–1093, Apr. 2013.

[20] K. Al-Agha, M. H. Bertin, T. Dang, A. Guitton, P. Minet, T. Val, and J. B. Viollet, "Which wireless technology for industrial wireless sensor networks? The development of OCARI technology," *IEEE Transactions on Industrial Electronics*, vol. 56, pp. 4266–4278, Oct. 2009.

[21] P. Neumann, "Communication in industrial automation - What is going on?," *Special Issue on Manufacturing Plant Control: Challenges and Issues INCOM 2004 11th IFAC INCOM'04 Symposium on Information Control Problems in Manufacturing*, vol. 15, pp. 1332–1347, Nov. 2007.

[22] P. Neufeld, U. Meier, L. Rauchhaupt, and M. Kratzig, "A unified approach for the assessment of industrial wireless solutions," *Emerging Technologies & Factory Automation (ETFA), 2011 IEEE 16th Conference on*, pp. 1–4, 2011.

[23] V. C. Gungor and G. P. Hancke, "Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches," *IEEE Transactions on Industrial Electronics*, vol. 56, pp. 4258–4265, Oct. 2009.

[24] I. Silva, L. A. Guedes, P. Portugal, and F. Vasques, "Reliability and availability evaluation of Wireless Sensor Networks for industrial applications," *Sensors*, vol. 12, no. 1, pp. 806–838, 2012.

[25] Y. Han and Y. H. Song, "Condition monitoring techniques for electrical equipment - a literature survey," *Power Delivery, IEEE Transactions on*, vol. 18, no. 1, pp. 4–13, 2003.

[26] A. Starr and B. K. N. Rao, *Condition Monitoring and Diagnostic Engineering Management*. Elsevier Science, 2001.

[27] P. Zhang, Y. Du, T. Habetler, and B. Lu, "A survey of condition monitoring and protection methods for medium voltage induction motors," *Energy Conversion Congress and Exposition, 2009. ECCE 2009. IEEE*, pp. 3165–3174, 2009.

[28] R. Isermann, *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. Springer-Verlag Berlin Heidelberg, 2006.

[29] "ISO 13374-1:2003 Condition monitoring and diagnostics of machines part 1: Data processing, communication and presentation: general guidelines."

[30] "ISO 13374-2:2007 Condition monitoring and diagnostics of machines part 2: Data processing."

[31] F. S. Nowlan and F. H. Heap, "Reliability-centered maintenance," tech. rep., United Airlines San Francisco, California, 1978.

[32] B. Keeter and D. Plucknette, "The seven questions of reliability centered maintenance," *RCM-2008 Reliability Centered Maintenance Managers' Forum.*, 2008.

[33] GE Aircraft Engines Business Group, *CF6-80C2 PMC Models Basic Engine and Systems*, training manual, special edition ii ed., 1990.

[34] R. Finch, *Converting Auto Engines for Experimental Aircraft*. Finch books, Finch R. Publishing Company, 1998.

[35] "SKF multilog on-line system WMX." Web. 02 Oct. 2013. <http://www.skf.com/binary/12-19873/Multilog-On-line-System-WMx-data-sheet.pdf>.

[36] "Status Check wireless condition monitoring system." Web. 02 Oct. 2013. <http://www.timken.com/>.

[37] R. C. Luo and O. Chen, "Mobile sensor node deployment and asynchronous power management for wireless sensor networks," *IEEE Transactions on Industrial Electronics*, vol. 59, pp. 2377–2385, May 2012.

[38] F. Hermans, N. Dziengel, and J. Schiller, "Quality estimation based data fusion in wireless sensor networks," *IEEE 6th International Conference on Mobile Adhoc and Sensor Systems, 2009. MASS '09.* , pp. 1068–1070, 2009.

[39] M. Wang, J. Cao, B. Chen, Y. Xu, and J. Li, "Distributed Processing in Wireless Sensor Networks for Structural Health Monitoring," *Ubiquitous Intelligence and Computing* (J. Indulska, J. Ma, L. Yang, T. Ungerer, and J. Cao, eds.), vol. 4611 of *Lecture Notes in Computer Science*, pp. 103–112, Springer Berlin Heidelberg, 2007.

[40] J.-F. Martínez, A.-B. García, A.-M. Sanz, L. López, V. Hernández, and A. Dasilva, "An Approach for Applying Multi-agent Technology into Wireless Sensor Networks," *Proceedings of the 2007 Euro American Conference on Telematics and Information Systems*, EATIS '07, (New York, NY, USA), pp. 22:1–22:8, ACM, 2007.

[41] "The sensor network museum." Web. 11 Oct. 2013.
<http://www.snm.ethz.ch/snmwiki/Main/HomePage>.

[42] "Mica datasheet 6020-0060-04 rev a," Crossbow Technology, Inc ., 2009.

[43] "The evolution of wireless sensor networks," tech. rep., Silicon Laboratories, Inc., 2013.

[44] "Em35x system-on-chip datasheet." Web. 12 Oct. 2013.
<http://www.silabs.com/products/wireless/zigbee>.

[45] B. Warneke and K. Pister, "MEMS for distributed wireless sensor networks," in *Electronics, Circuits and Systems, 2002. 9th International Conference on*, vol. 1, pp. 291–294 vol.1, 2002.

[46] C. He, M. E. Kiziroglou, D. C. Yates, and E. M. Yeatman, "A MEMS self-powered sensor and RF transmission platform for WSN nodes," *Sensors Journal, IEEE*, vol. 11, pp. 3437–3445, Dec 2011.

[47] "Crankshaft torque wireless transmission." Web. 12 Oct. 2013.
<http://www.irtelemetrics.com>.

[48] P. Levis and D. Gay, *TinyOS Programming*. Cambridge University Press, 2009.

[49] "Nano-RK: A wireless sensor networking real-time operating system." Web. 12 Oct. 2013.
<http://www.nanork.org>.

[50] "SOS 2.x home page." Web. 12 Oct. 2013.
<http://projects.nesl.ucla.edu/public/sos-2x/doc/>.

[51] "Open source stuff for DASH7." Web. 12 Oct. 2013.
<http://www.indigresso.com/wiki/doku.php?id=opentag:main>.

[52] Daintree Networks Inc., *Getting Started with ZigBee and IEEE 802.15.4*, tech. rep. 2013.

[53] "Wireless sensor networks open new frontiers for a smarter world." Web. 12 Oct. 2013.
<http://www.ti.com/lit/ml/sszy008/sszy008.pdf>.

[54] "ANT Message Protocol and Usage," tech. rep., Dynastream Innovations Inc., 2013.

[55] "Wireless Sensor Networks - Dust Networks." Web. 12 Oct. 2013.
<http://www.linear.com/products/wireless_sensor_networks>.

[56] "Personal area networks MIW protocol," tech. rep., Microchip Technology Inc., 2013.

[57] M. Becker, A. Timm-Giel, K. Murray, C. Lynch, C. Görg, and D. Pesch, "Comparative Simulations of WSN," *ICT-MobileSummit 2008* (P. Cunningham and M. Cunningham, eds.), 2008.

[58] M. Korkalainen, M. Sallinen, N. Karkkainen, and P. Tukeva, "Survey of Wireless Sensor Networks Simulation Tools for Demanding Applications," *Networking and Services, 2009. ICNS '09. Fifth International Conference on*, pp. 102–106, 2009.

[59] "NesCT: A language translator." Web. 12 Oct. 2013. <http://nesct.sourceforge.net/index.html>.

[60] E. L. Deitch, *Learning to Land: A Qualitative Examination of Pre-Flight and In-Flight Decision-Making Processes in Expert and Novice Aviators.* Ph.D. thesis, Virginia Polytechnic Institute and State University, 2001.

[61] G. Yong, W. Kui, and L. Fulu, "Analysis on the redundancy of wireless sensor networks," *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, (San Diego, CA, USA), pp. 108–114, ACM, 2003.

[62] K. K. Tan, S. N. Huang, Y. Zhang, and T. H. Lee, "Distributed fault detection in industrial system based on sensor wireless network," *Industrial Networking Standards for Real-time Automation and Control*, vol. 31, pp. 573–578, Mar. 2009.

[63] B. Latre, P. Mil, I. Moerman, N. Dierdonck, B. Dhoedt, and P. Demeester, "Maximum Throughput and Minimum Delay in IEEE 802.15.4," *Mobile Adhoc and Sensor Networks* (X. Jia, J. Wu, and Y. He, eds.), vol. 3794 of *Lecture Notes in Computer Science*, pp. 866–876, Springer Berlin Heidelberg, Jan. 2005.

[64] X. Dai, K. Sasloglou, R. Atkinson, J. Strong, I. Panella, C. L. Yun, H. Mingding, W. A. Chee, I. Glover, J. E. Mitchell, W. Schiffers, and P. S. Dutta, "Wireless Communication Networks for Gas Turbine Engine Testing," *International Journal of Distributed Sensor Networks*, vol. 2012, 2012.

[65] A. C. Lima-Filho, R. D. Gomes, M. O. Adissi, T. A. Borges da Silva, F. A. Belo, and M. A. Spohn, "Embedded System Integrated Into a Wireless Sensor Network for Online Dynamic Torque and Efficiency Monitoring in Induction Motors," *IEEE/ASME Transactions on Mechatronics*, vol. 17, pp. 404–414, June 2012.

[66] B. Son, Y. Her, and K. J., "A design and implementation of forest-fires surveillance system based on wireless sensor networks for South Korea mountains," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 6, pp. 124–130, 2006.

[67] R. Teoste, "Digital Circuit Redundancy," *IEEE Transactions on Reliability*, vol. R-13, no. 2, pp. 42–61, 1964.

[68] E. F. Nakamura, A. A. F. Loureiro, and A. C. Frery, "Information fusion for wireless sensor networks: Methods, models, and classifications," *ACM Comput. Surv.*, vol. 39, no. 3, p. 9, 2007.

[69] L. Hou and N. W. Bergmann, "Novel Industrial Wireless Sensor Networks for Machine Condition Monitoring and Fault Diagnosis," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, pp. 2787–2798, Oct. 2012.

[70] J. Gordon and E. H. Shortliffe, "The Dempster-Shafer theory of evidence," in *Rule Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project* (G. G. Buchanan and E. H. Shortliffe, eds.), Redwood City, CA: Addison-Wesley, 1984.

[71] J. A. Barnett, "Computational methods for a mathematical theory of evidence," *Classic Works of the Dempster-Shafer Theory of Belief Functions* (R. Yager and L. Liu, eds.), vol. 219 of *Studies in Fuzziness and Soft Computing*, pp. 197–216, Springer Berlin Heidelberg, Jan. 2008.

[72] G. Shafer, "Perspectives on the theory and practice of belief functions," *Int. J. Approx. Reasoning*, vol. 4, pp. 323–362, Oct. 1990.

[73] R. Kennes and P. Smets, "Fast algorithms for Dempster-Shafer theory," *Uncertainty in Knowledge Bases* (B. Bouchon-Meunier, R. Yager, and L. Zadeh, eds.), vol. 521 of *Lecture Notes in Computer Science*, pp. 14–23, Springer Berlin Heidelberg, Jan. 1991.

[74] A. Bellenger and S. Gatepaille, "Uncertainty in ontologies: Dempster-Shafer theory for data fusion applications," *CoRR*, vol. abs/1106.3876, 2011.

[75] A. Abdelgawad and M. Bayoumi, "Low-power distributed Kalman filter for wireless sensor networks," *EURASIP J. Embedded Syst.*, vol. 2011, pp. 1–11, 2011.

[76] J. Cooley and J. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," *Mathematics of Computation*, vol. 19, no. 90, pp. 297–301, 1965.

[77] T. Roberts, M. Slaney, and P. D. Bouras, "Fixed-point Fast Fourier Transform." Web. 5 Aug. 2013.
<http://www.dbouras.eu/>.

[78] J. Vysoky, "Signal Processing Method Implementation into WSN Nodes (in Czech)," Master's thesis, Czech Technical University in Prague, Faculty of Electrical Engineering, 2011.

[79] G. F. Mugurel, "Data processing in Wireless Sensor Networks implemented on Imote2 Sensor Nodes with ITS400 Sensor Board," Master's thesis, Transilvania University of Brasov - Faculty of Electrical Engineering and Computer Science, 2012.

[80] “TinyOS home page.” Web. 12 Oct. 2013. <http://www.tinyos.net/>.

[81] P. Geiger and K. Nadig, “Arduino FIST: MATLAB Fuzzy Inference System to Arduino C Converter.” Web. 24 Jun. 2012. <http://www.makeproto.com/projects/fuzzy/matlab_arduino_FIST>.

[82] N. Poh, T. Bourlai, J. Kittler, L. Allano, F. Alonso-Fernandez, O. Ambekar, J. Baker, B. Dorizzi, O. Fatukasi, J. Fierrez, H. Ganster, J. Ortega-Garcia, D. Maurer, A. Salah, T. Scheidat, and C. Vielhauer, “Benchmarking Quality-Dependent and Cost-Sensitive Score-Level Multimodal Biometric Fusion Algorithms,” *Information Forensics and Security, IEEE Transactions on*, vol. 4, pp. 849–866, Dec. 2009.