Enhancing substation condition monitoring through integrated diagnostics, wireless sensor networks and multi-agent systems

Peter C. Baker

A thesis submitted for the degree of Doctor of Philosophy to
Institute for Energy and Environment
University of Strathclyde

2010

Signed:

Date:

Abstract

Due to the increased regulatory and operational demands faced by electrical utilities, the need for effective condition monitoring is a key requirement to avoid equipment failure, ensure equipment uptime, and mitigate business risk in an increasingly competitive marketplace. The addition of new condition monitoring sensors has led to the use of intelligent systems to interpret sensor data, providing automated diagnosis and data management to monitoring engineers. However, typically only the most critical assets are subject to online condition monitoring due to the high cost of instrumenting plant items, so the health of other assets can often go unseen between scheduled maintenance periods, or in the worst-case, until failure.

Utilities face several issues when installing new condition monitoring sensors, ranging from physical deployment constraints to managing and interpreting new complex datasets. By using wireless sensor networks, utilities can extend the coverage of substation condition monitoring systems down to the bay and component level through low-cost, unobtrusive sensors. This thesis investigates enhancing condition monitoring systems through the use of wireless sensors which can host diagnostics at the sensor level to immediately convert sensor data to useful monitoring information at the source.

However, as these devices are discrete and low-power, new approaches to fault detection are required that are suited to operating in this environment. The feasibility of this is demonstrated through a novel low-power sensing technique for electrical insulation health monitoring. This technique, implemented as a wireless sensor, is complemented with embedded diagnostic capabilities operating directly on the sensor node. To demonstrate that this approach can be integrated with existing monitoring systems, a multi-agent approach is proposed. This integrated approach, encompassing data capture, diagnostics and system architectures demonstrates that low-power sensing, sensor networks and multi-agent systems can meet the needs of industry to form the backbone of new substation condition monitoring networks.

Acknowledgements

I would first like to take the opportunity to thank Profs. Stephen McArthur and Jim McDonald for giving me the opportunity to return to the department to undertake a PhD.

Thanks are also due to Martin Judd for the opportunity to branch out my research into the field of RF monitoring, and allow me to get my hands dirty in the high-voltage lab. I would also like to thank everyone in the Advanced Electrical Systems Group, whose intellectual support and endless supply of banter over the past four years has been invaluable to my academic and personal development.

I'd also like to thank my parents for the ongoing encouragement and support. Of course, my heartfelt gratitude goes to Brook Yu, who is forever there to share ideas with no matter how intellectual or trivial. Finally, a special thanks to all my close friends who have taught me so much outside of the world of academia. I dedicate this work to you all for always being there with open arms and making life such an enriching experience.

Contents

List of Figures

List of Tables

Glossary of abbreviations

ACL	Agent Communication Language
AI	Artificial Intelligence
AID	Agent Identifier
AIS	Air-insulated substation
AMPerES	Asset Management and Performance of Energy Systems
AMS	Agent Management System
API	Application Programming Interface
Bluetooth	Personal area network communications protocol
CH_4	Methane
C_2H_2	Acetylene
C_2H_4	Ethylene
CIGRÉ	International Council on Large Electric Systems
CIM	Common Information Model
CBM	Condition Based Maintenance
CM	Condition Monitoring
COMMAS	Condition Monitoring Multi-Agent System
DF	Directory Facilitator (agent)
DFR	Digital fault recorder
DGA	Dissolved Gas Analysis
DSN	Distributed Sensor Network; same as WSN
EIS	Electrical Insulation Systems
EM	Electromagnetic
FFT	Fast Fourier Transform
FIPA	Foundation for Intelligent Physical Agents
FIPA-ACL	The FIPA standard Agent Communication Language
FIPA-SL	The FIPA Semantic Language
GIS	Gas-Insulated Substation or Gas-Insulated Switchgear
GPRS	General Packet Radio Service
GUI	Graphical User Interface

HTTP	Hypertext Transfer Protocol
IEC	International Electrotechnical Commission
IED	Intelligent Electronic Device
IEEE	Institute of Electrical and Electronics Engineers
I/O	Input / Output
JADE	Java Agent Development Framework
J2ME	Java 2 Mobile Edition
J2SE	Java 2 Standard Edition
KIF	Knowledge Interchange Format
MAC address	Media Access Control address
MBR	Model based reasoning
Mote	Common name for a wireless sensor node (North America)
nesC	Embedded programming language for sensor networks
OWL	Web Ontology Language
PD	Partial Discharge
PRPD pattern	Phase-resolved Partial Discharge pattern
PV	Photovoltaic
PEDA	Protection Engineering Diagnostic Agents
SBC	Single Board Computer
SF_6	Sulphur Hexafluoride
SL	Semantic Language
SUMO	Suggested Upper Merged Ontology
SUPERGEN	Sustainable Power Generation and Supply
SVM	Support Vector Machine
TinyOS	Embedded Operating System for Motes
TCP/IP	Transmission Control Protocol/Internet Protocol
UHF	Ultra-High Frequency
VM	Virtual Machine
VT	Voltage Transformer
WiFi	Common name for 802.11 family of wireless protocols
WSN	Wireless Sensor Network
XML	eXtensible Markup Language

Chapter 1

Introduction

“I have never let my schooling interfere with my education.”

- **Mark Twain, 1835-1910**

1.1 Introduction to the research

Over the past 125 years, electrical transmission and distribution networks have grown to form the backbone of modern society, underpinning technological advancements, international trade, communications and industry through the provision of readily available, safe and affordable energy. Electrical networks are some of the largest interconnected systems created by man; spanning countries and continents to reach billions of homes worldwide. Year on year, results published in the United Nations Human Development report have demonstrated that the level of human development is linked to the availability and usage of energy [1], therefore to maintain the high standard of living enjoyed by developed nations, it is necessary to maintain a secure energy infrastructure.

To ensure this, effective monitoring is required so that engineers may understand the ongoing health of electrical plant to create informed maintenance decisions. Without this, failures can occur which can lead to incurred regulatory charges for loss of service, loss of customer revenue, and in more serious cases, catastrophic failure which may lead to complete replacement of an asset.

The requirement for effective condition monitoring on the electrical network is no more crucial than within electrical substations, which form the critical nodes of the electrical network. Historically, within substations, online health monitoring of individual plant items has been limited; only in the past 25 years has condition monitoring of plant seen significant research as the age of plant has approached its original estimated lifespan. In this period, a number of techniques have been developed to monitor high voltage plant, although typically only applied to critical assets due to the associated high costs. Online systems are typically stand-alone and can generate large volumes of data which must be analysed by engineers. This process is time consuming, and due to the complexity of the datasets, trends that may indicate the cause of problems or the onset of potential failures may go unseen.

Intelligent systems techniques have been applied to automate the intensive analytical process and to learn and reason on underlying trends in data. These techniques have been paired with multi-agent systems technology, not only as a platform on which to build distributed systems, but also as a means to integrate disparate systems together into a single architecture to process raw sensor data into a meaningful representation of the equipment condition.

While these systems are able to provide analysis and decision support to engineers, some deficiencies still exist. Typically such systems have only been applied to a single piece of plant that has been instrumented with wired sensors. Widespread deployments of this technology are infeasible, as they require both a suitable communications architecture and ideally require data management to be embedded within the condition monitoring network to control and filter the flow of data. Installing wired communications infrastructure within substations is a non-trivial task, leading to prohibitively high costs from both capital outlay and required outages and also raising a number of safety and operational concerns. This thesis will demonstrate that these issues can be addressed, in part, through the application of wireless sensor networks.

Wireless sensor networks have already seen widespread use within industry in areas such as environmental monitoring and military applications, although to-date limited deployments have been made within the power domain. These devices are well suited to substation applications with strict operational constraints, as they can be used to instrument existing equipment without the need for wired communications links throughout the substation. With the addition of low-power, non-intrusive sensors, plant may be instrumented with little or no downtime; crucial when outages must be planned months or years in advance.

This thesis explores the issues of extending multi-agent diagnostic systems with wireless sensor networks. The requirements for such a system will be presented, including the need for novel low-power sensors that are suited to wireless, battery-run operation. To illustrate how this may be achieved, a novel diagnostic technique for insulation health monitoring was created that adapts an existing monitoring technique to low-power operation. To complement this, on-sensor diagnosis was achieved through an online diagnostic classifier suited to an embedded environment.

Despite parallels in the nature of multi-agent systems and sensor networks, there are fundamental differences due to energy constraints placed on battery-operated devices. This thesis will discuss the issues of integrating the two systems, using a multi-agent approach for translation that also improves the flexibility and functionality of the monitoring system as a whole.

The multi-agent architecture is demonstrated through three case studies. Through these case studies, we will see that multi-agent systems combined with wireless sensor networks can be a powerful combination for building distributed, extensible substation condition monitoring networks. Intelligent agents operating within a substation sensor network will be able to communicate and collaborate with each other to diagnose faults quicker, whilst being flexible enough to assume new diagnostic capabilities as monitoring techniques evolve.

1.2 Justification for research

Since the advent of Thomas Edison's first commercial electrical power generation scheme at Pearl Street Station in New York, 1882, the worldwide electrical power industry has grown exponentially in size and complexity, although the fundamentals behind even today's network are the same as 125 years ago. More recently, there has been a resurgence in power engineering research due to a greater increase in interest in the domain driven by a number of factors, including: deregulation of electricity markets, the impact of anthropogenic climate change, rising energy prices due to an increased world demand, and the ever-increasing average age of electrical plant.

Electrical plant are expected to continue performing at an optimal ability despite approaching or exceeding their original design lifetime. Unplanned outages and unforeseen replacement of critical assets by utilities is undesirable due to the high capital cost of asset replacement and the effects of running the electrical network in a sub-optimal condition, therefore monitoring engineers should maintain the health of plant through the careful application of condition monitoring technologies that can evolve to meet the ever-changing requirements placed upon them.

With any distributed monitoring system, as more sensors are added, the communications load rises due to an increase in sensory data. The approach used for operational monitoring systems such as SCADA is to transfer all data back to a central point for processing, which has required a significant investment in communication infrastructure to accommodate connections to all sensors. While this may be feasible for power system operations which have to operate in real-time, it is infeasible for condition monitoring systems which typically see limited changes over the lifetime of a plant item as they do not warrant a broad-band communications network. Wired instrumentation of plant items is also prohibitively expensive and could potentially require a planned outage for all affected plant items, therefore it is clear that an alternative, wireless, cost-effective method of instrumenting substations is required; this can be achieved through the application of wireless sensor network technology.

Sensor networks consist of multiple discrete nodes, each with integrated communications, storage, processing and sampling capabilities. Within a substation, they can be used to build a network of condition monitoring sensors which can not only capture monitoring data, but also host diagnostic applications which can perform in-network analysis.

For a utility to implement such technologies, both hardware and software constraints must be considered. Wireless sensing places constraints on the physical sensors that may be attached to sensor nodes, due to energy availability and sampling capabilities. This requires a new, low-power approach to condition monitoring sensors. Effective in-network processing must also be incorporated to manage the flow of additional monitoring data, which in turn requires a suitable software methodology with which to implement it. The multi-agent approach has previously been used in this area, and therefore can be applied, with modification, to the embedded, wireless environment.

To this end, this thesis addresses these requirements through the application of multi-agent systems and wireless sensor networks to substation condition monitoring. Effective low-power sensing is demonstrated through a novel battery-operated PD detector, which is backed by a multi-agent toolkit for wireless sensing applications. On-sensor data management is demonstrated through diagnostic agents which are capable of automated defect diagnosis in real-time. Integration of the sensor network platform with an existing CM system is demonstrated through a flexible agent-based approach. Combined, these individual components represent a blueprint for future substation condition monitoring system based on sensor network and multi-agent technology. The novel aspects of the research can be summarised as follows:

- An investigation of agent-based middleware for wireless sensor networks and their suitability towards condition monitoring applications, including suggested requirements for substation-based sensor networks;
- A novel low-power RF partial discharge detection method, which can distinguish between different defect emissions based upon analysis of the PD pulse frequency spectrum;
- A novel approach to visualising partial discharge frequency spectra;
- An experiment to determine factors affecting frequency spectra of emissions from partial discharge test cells in a laboratory environment. Analysis of the results of this experiment led to:
- The development of a C4.5 classifier for frequency-based partial discharge classification;

- An agent-based classification system for in-sensor-network diagnosis of partial discharge events;
- A multi-agent toolkit for agent-based sensor network diagnostic applications, incorporating;
- An implementation of a sensor network gateway, which translates bit-efficient sensor network messages to FIPA protocol-based messages, providing a standardised agent interface to sensor network functions;
- A case study demonstrating the integration of simple temperature and vibration sensors with an existing multi-agent condition monitoring system;
- A case study showing on-sensor diagnosis of partial discharge defects integrated with an existing multi-agent condition monitoring system; and,
- A case study demonstrating the enhancement of an existing diagnostic system by deploying diagnostic capabilities at the sensor level.

1.3 Thesis overview

This thesis is comprised of 8 chapters including the introductory chapter, plus 3 appendices. Chapter 2 provides background information on substation condition monitoring, including a review of existing condition monitoring techniques.

Chapter 3 investigates multi-agent systems, including; their benefits, the technology that underpins them, and their application to power systems. This format is reflected in Chapter 4, which investigates wireless sensor network technology and its suitability towards substation condition monitoring.

In Chapter 5, low-power substation condition monitoring techniques are investigated, highlighting a need for a low-power approach to electrical insulation health monitoring. In response to this, a novel partial discharge monitoring technique was implemented and verified through laboratory experimentation. To simplify analysis of the experimental results, a novel visualisation method for partial discharge frequency data was implemented.

Chapter 6 presents the key issues regarding the integration of new wireless condition monitoring systems with existing agent-based condition monitoring architectures. This is achieved through the implementation of a sensor network gateway agent which integrates 'SubSense', a sensor network multi-agent platform, with a FIPA multi-agent system.

Chapter 7 includes three deployment case studies, which demonstrate the approaches laid out in the previous chapters through relevant industrial applications. This includes the integration of a wireless temperature and vibration monitor with an existing substation diagnostic multi-agent system, the integration of a low-power partial discharge detector with the SubSense condition monitoring architecture, and an enhancement to an existing condition monitoring application. The last two applications make use of a PD detector simulator, used to simulate PD events on wireless sensor nodes without the need for high-voltage equipment.

The thesis is concluded in Chapter 8, by consolidating the research presented in the previous chapters. It also highlights the opportunities for further research that is outside the scope of this thesis.

The three appendices offer supplementary information on the low-power PD detector design, multi-agent toolkit, and PD detector simulator.

1.4 Associated publications

Publications which have arisen as result of the work described in this thesis are as follows:

- P. C. Baker, S. D. J. McArthur, and M. D. Judd, "A Frequency-Based RF Partial Discharge Detector for Low-Power Wireless Sensing," *IEEE Transactions on Dielectrics and Electrical Insulation*, vol. 17, pp. 133 –140, Feb. 2010
- P. C. Baker, S. D. J. McArthur, and M. D. Judd, "Data Management of On-Line Partial Discharge Monitoring Using Wireless Sensor Nodes Integrated with a Multi-Agent System," in *Intelligent Systems Applications to Power Systems, 2007. ISAP 2007. International Conference on*, Nov. 2007
- P. C. Baker, B. Stephen, M. D. Judd, and S. D. J. McArthur, "Development of an integrated low-power RF partial discharge detector," in *Electrical Insulation Conference, 2009. EIC 2009. IEEE*, pp. 273 –277, May 2009
- P. C. Baker, V. M. Catterson, and S. D. J. McArthur, "Integrating an agent-based wireless sensor network within an existing multi-agent condition

monitoring system," in *Proceedings of: Intelligent Systems Applications to Power Systems, 2009. ISAP 2009. International Conference on*, Dec. 2009

- P. C. Baker, S. D. J. Rudd, S. McArthur, and M. D. Judd, "A conceptual design for a low-power wireless UHF detector," in *Proceedings of: Third UHVNet Colloquium on technologies for future high voltage infrastructure, 2010*, Jan. 2010

Other publications on related topics are:

- M. Zhu, P. C. Baker, N. M. Roscoe, M. D. Judd, and J. Fitch, "Alternative power sources for autonomous sensors in high voltage plant," in *Proceedings of the 2009 IEEE Electrical Insulation Conference (EIC), Montreal Canada*, pp. 36–40, June 2009

- A. J. Mair, R. R. Soman, P. C. Baker, E. M. Davidson, S. D. J. McArthur, S. K. Srivastava, M. Andrus, and D. A. Cartes, "Progress in the development of adaptive control for shipboard power systems through modelling and simulations," in *Grand Challenges in Modeling and Simulation, Proceedings of.*, Society for Modelling and Simulation International, July 2010

Chapter 2

Substation condition monitoring

"Energy is eternal delight."

- **William Blake, 1757-1827**

2.1 Substations

Substations provide a critical function within the electrical network, linking generation, transmission and distribution systems together. They range from small, single voltage level sites that solely perform switching operations to large multiple voltage level sites which can span many acres.

Their main purpose is to provide switching and voltage conversion through primary systems, and protection and monitoring through secondary systems. Voltage conversion is provided by power transformers, which convert between the various transmission and distribution voltage levels. Control and protection is carried out using a number of different schemes, including protective relays and Intelligent Electronic Devices (IEDs) typically connected to high-voltage conductors via voltage or current transformers, which trip circuit breakers when current exceeds a safe operational level. Disconnectors provide circuit isolation for de-energised circuits for maintenance purposes, and busbars and cables link the various pieces of high-voltage plant together.

Air-insulated substations (AIS) are historically the norm, where plant is contained within a wide fenced open area and a large air clearance must be maintained to insulate energised components. An example of an air-insulated substation is shown in Figure 2.1[1].

An alternative to AIS, Gas-Insulated Substations (GIS) have steadily increased in popularity since the first deployment in Germany in 1967 [9]. Within GIS, all high-voltage components including busbars, switchgear and peripheral equipment are housed within Sulphur Hexafluoride (SF_6) filled aluminium enclosures. SF_6 is used as an insulating medium for its high dielectric strength compared to air or vacuum; giving many advantages. GIS are typically 20% of the size of AIS, making them suitable for industrial and residential areas. They are also more reliable than their air-insulated counterparts as they are less susceptible to adverse weather conditions and pollution [10]. Since 2000, there have been more than 20000 GIS bays installed worldwide [11]. Hybrid substations also exist, where equipment is a mix of air-insulated and gas-insulated components. An example of a gas-insulated substation can be seen in Figure 2.2.

Figure 2.1: An example of an air-insulated substation. National Grid Legacy station nr. Wrexham, Wales

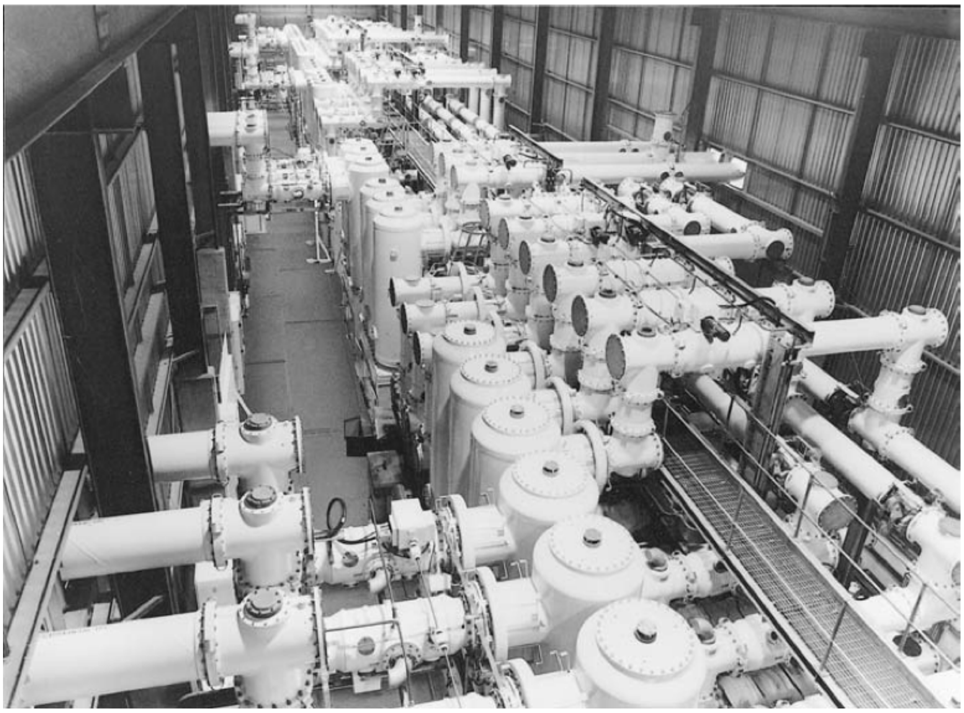


Figure 2.2: An example of a gas-insulated substation [10]

2.2 Condition monitoring

Condition monitoring is defined as a process which monitors an operational parameter of a piece of equipment or machinery, such that changes in that parameter over time are indicative of a fault, to establish the current health of the equipment, and to predict the need for maintenance before a serious failure occurs [12].

In the energy industry, condition monitoring allows utilities to move from periodic maintenance, where plant is taken out of service for maintenance on a periodic basis or based upon its operational lifetime, towards predictive maintenance or condition based maintenance (CBM), where plant maintenance is scheduled based upon the ongoing health of the asset.

A move to CBM allows utilities to move towards a lower-cost model for maintenance, through removing unnecessary outages and discovering incipient faults before they become catastrophic. CBM also increases reliability, allowing the operational lifetime of plant to be extended beyond that of the original design lifetime. In the current deregulated energy market, this is desirable where competition is fierce, and the financial penalties for unexpected loss of supply are severe.

2.2.1 Requirements of new substation condition monitoring systems.

In [12], the four key components of a condition monitoring system are described as:

Sensors: to convert physical measurements into a form which can be measured electronically;

Data acquisition: incorporating analog to digital data conversion and optional signal processing and signal conditioning;

Fault detection: through pattern recognition techniques or AI methods such as model-based or knowledge-based reasoning and data driven classifiers; and,

Diagnosis: when a fault condition has been recognised, it is presented to monitoring engineers along with the likely cause of the fault to aid in the maintenance process.

In addition to this, requirements for new condition monitoring systems include [13, 14]:

Reliability: specifically over many years; the plant under inspection may be expected to operate for 50 years or more so any new monitoring system is expected to operate over a useful proportion of this lifespan;

Low cost: sensors and monitoring systems in comparison to the cost of the asset, with appropriate functionality. Non-invasive methods are preferred, to give an indication of the internal health of an asset while minimising outages required for installation; and,

Compatibility: with new and existing monitoring systems to simplify deployment.

The availability of techniques for electrical plant condition monitoring has increased in recent years [12], with advances in data acquisition, signal processing and AI diagnostics making this possible. However, detection and diagnostic systems are typically discrete from sensing and data acquisition systems, where physical sensors and data capture are separated from interpretation and diagnostic methods. Online diagnostic systems bridge the gap between sensing and diagnosis, although utilities face a number of hurdles to implement fully integrated substation condition monitoring systems.

The first step is for sensor and data acquisition systems to be physically connected together. Within substations, operational and safety regulations make cabling an expensive task for utilities, therefore data acquisition systems must be in close proximity to sensor systems. For a condition monitoring system to operate in an online context, the data acquisition system must be linked to the diagnostic system either directly or via a communication link, which would typically reside on a PC either on the corporate network or at the edge of the substation where the same constraints for cabling apply. Utilities, therefore, require a means of effectively integrating condition monitoring components together. In this thesis, we will see that it is possible to achieve this with respect to the requirements laid out above, where low-cost, reliable sensors can be realised that integrate flexibly with existing condition monitoring systems.

2.3 Substation assets

Within substations, the two plant types which usually constitute the largest capital value are transformers and switchgear. To gain an understanding of condition monitoring techniques for these assets, first we must gain an understanding of the fundamentals of these plant types.

2.3.1 Power transformers

Power transformers perform voltage level conversion within the electrical network. Typically, transmission networks operate at voltages at 275kV and above, whereas the distribution network operates at voltages between 132kV and 11kV. A power transformer's function is to convert the voltage level; stepping-up for transmission, and stepping-down for distribution.

Power transformers consists of at least 2 windings, usually referred to as the *primary* and *secondary* windings, which are typically made from paper-insulated copper, wound around a common core made of laminated steel sheets. The voltage ratio between the supply and load is equal to the ratio between the number of turns in the primary and secondary windings. Transformers with more than 2 windings are also used; for instance a tertiary winding is often included to supply auxiliary loads or balance harmonic currents. An example of a 3-phase power transformer is shown in Figure 2.3.

In addition to the core and windings, power transformers may also include a number of peripheral components, such as:

Cooling components: which can be either passive or active. Typically, distribution transformers are cooled through natural thermosiphon forces, whereas transmission transformers are usually fitted with oil pumps to circulate the oil, and radiator fans to dissipate excess heat;

Tap Changers: which can be either on- or off-circuit. Off-circuit tap changers are used to change voltage when the transformer is de-energised. In the case of transmission transformers, on-load tap changers (OLTC) are typically used to modify the load voltage level when the unit is in operation; and,

Bushings: to connect the transformer to the network. These penetrate the transformer tank, connecting the internal windings to a busbar or cable.

In-service transformers typically have a 25-40 year design life, or up to 60 years or more with appropriate maintenance [13]. With rapid expansion of the UK and North American electrical networks occurring between the 1950s and 1970s, it is clear that the assets installed during this period are either approaching or past their original design life. Failures to substation equipment

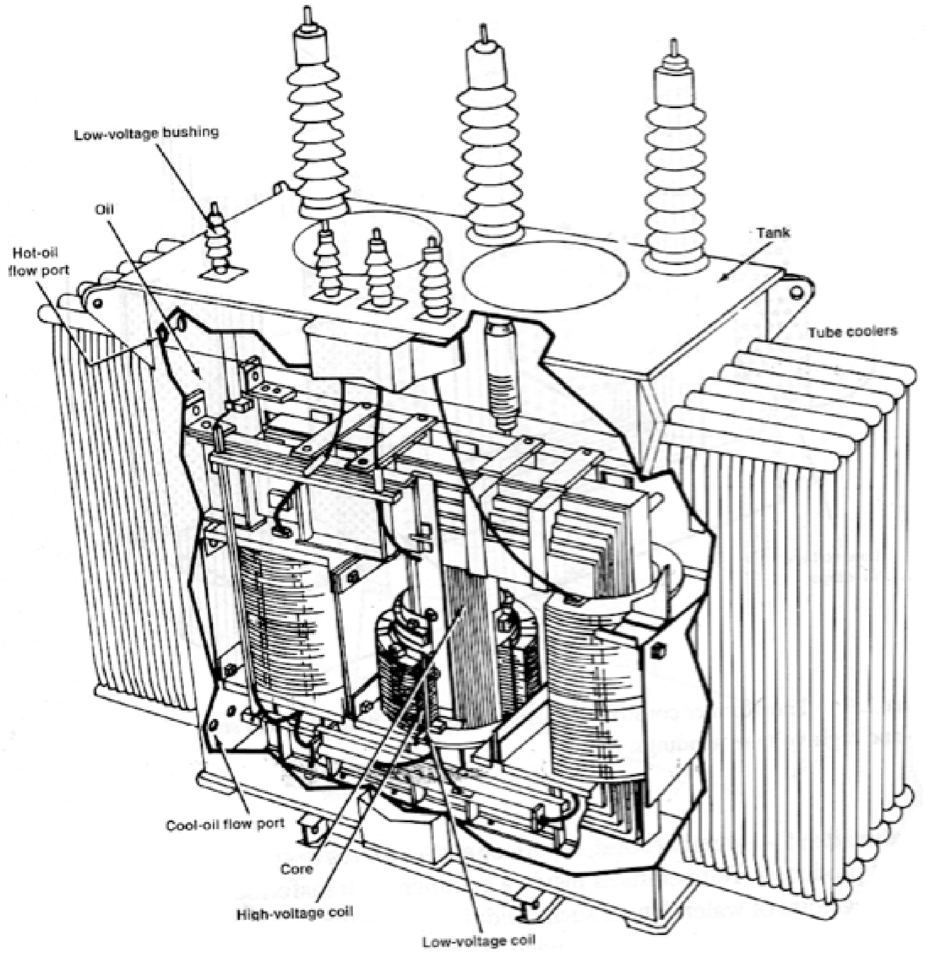
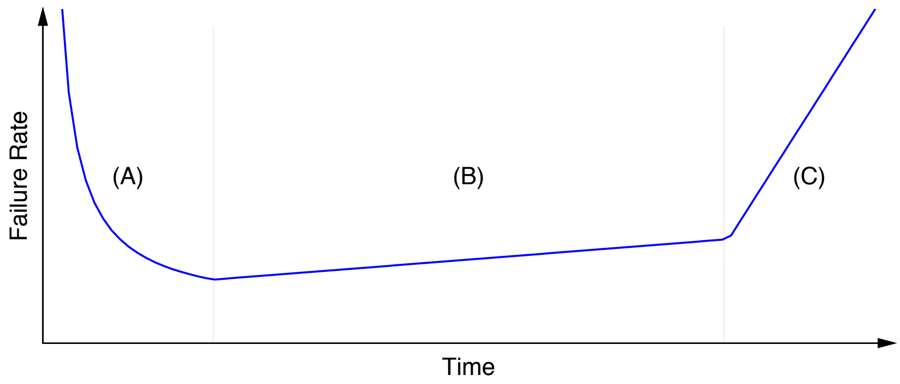


Figure 2.3: A three-phase power transformer [15]

typically follow a bathtub curve, as shown in Figure 2.4. Over equipment lifetime, the function of failure falls into three regions; early failures introduced from manufacturing or installation defects, random operational failures that may be expected over the course of the equipment lifetime, and an increasing failure rate due to equipment ageing. This is true not just of power transformers, but is typical of all power systems assets.

The lifetime of a transformer is commonly recognised as being equal to the life of its insulation system [16]. Over time, the paper and oil insulation system undergoes irreversible ageing, accelerated by increases in temperature typically caused during 'extreme conditions' outwith the expected operational mode of the plant including overloading, and surges from lightning and switching. The fault mechanisms include [13]: applied mechanical forces such as electromagnetic forces from fault conditions which can cause permanent deformation; thermal ageing where chemical reactions in the in-



(A) Dominating early failures

(B) Random failures

(C) Dominating ageing failures

Figure 2.4: The 'bathtub' failure curve

sulating oil and cellulosic paper can generate moisture and fault gases; overvoltages, which can cause arcing or temperature increases leading to thermal faults, and; contamination, either of moisture, oxygen, or foreign bodies introduced during maintenance or peripheral equipment faults, such as metal particles introduced from oil pump failures.

The two top failures for power transformers have been found to be OLTC and winding/insulation faults [17]. For power transformers without OLTC, insulation related defects constitute over a quarter of all defects. Key recommended power transformer monitoring parameters are OLTC operation, insulating oil and paper health, temperature, and operational load [12].

2.3.2 Switchgear

Switchgear is the general term used to encompass all switching components within a substation including circuit breakers (on-load) and disconnectors (off-load). There are a number of different types of switchgear used in operation, with the most common at distribution and transmission levels being oil-, air- and SF_6-insulated.

Within the switchgear family of plant items, the most widely monitored plant type is the circuit breaker. As they have electromechanical parts, the majority of failure modes within circuit breakers relate to the operating mechanism [18]. At the transmission level, circuit breakers may be opened or closed weekly to change the topology of the transmission network, however at the distribution level certain circuit breakers, for instance in residential areas, may not be operated for months or years. This can have a negative effect on the mechanism which may become impeded over time due to mechanical forces such as coagulation of lubricant, or stiction—the static cohesion caused between two resting objects in contact [19]. To overcome this, their operating mechanisms may be periodically exercised manually to ensure that they are in working order.

2.3.2 b) Gas-insulated switchgear

Gas-insulated switchgear use SF_6 as an insulating medium within aluminium enclosures, as shown in Figure 2.2. GIS make up the majority of new installations due to their increased reliability and the superior dielectric strength and arc-quenching capabilities of SF_6 compared to other gases [10]. In SF_6 insulated systems, the primary types of defects are [20]:

- Effects of moisture and SF_6 decomposition products;
- Mobile metallic particles; and,
- Fixed metallic particles and protrusions.

Moisture is managed within GIS through the use of special absorbers, which can be replaced periodically if required. In normal operation, the moisture content and SF_6 decomposition products have no effect on the long-term performance of GIS. In fact, SF_6 is extremely stable over time, although the presence of particles and protrusions can cause partial discharge (PD) to occur during operation which can result in a build-up of decomposition gases.

2.3.2 b) Other switchgear types

Other switchgear types use oil, vacuum, and air as their insulation mechanism. Vacuum insulated switchgear are designed to be sealed throughout their operational lifetime, and oil- and air-insulated circuit breakers that remain in service typically must undergo regular manual periodic inspection to ensure their operation [21]. Integrated monitoring schemes for circuit breakers are discussed later in this chapter in Section 2.4.4.

2.4 Online monitoring techniques

In addition to standard online voltage and current measurement used for operational monitoring, a number of online condition monitoring techniques have been developed to detect and diagnose the fault conditions and ageing mechanisms detailed previously. The majority of these monitoring methods are concerned with the health of the electrical insulation system of high-voltage plant, however additional tests are employed for specific plant components.

2.4.1 Chemical tests

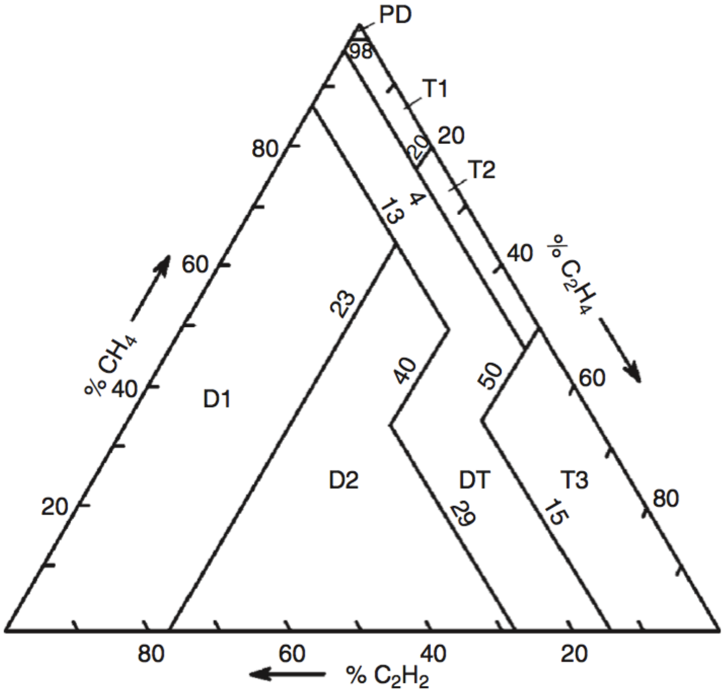
2.4.1 a) Dissolved gas analysis

Dissolved gas analysis (DGA) of transformer insulating oil is the most common method for transformer insulation health monitoring. DGA assesses the levels of fault gases dissolved in transformer oil which are primarily caused by overheating, electrical discharge and PD. Depending on the severity of the fault, different gases are produced, due to a breakdown in the insulating oil creating free radical molecules which recombine as fault gases in varying ratio depending on the temperature of the fault; for instance, hydrogen is produced from low-energy fault conditions, whereas acetylene is produced by the highest energy fault conditions.

Online gas monitors are available, such as the Kelman Transfix and GE Hydran. In the past, on-line monitors have typically only been installed on the most critical assets which are either known or suspected of undergoing some fault condition, however more recently they are becoming more widespread for routine use.

Analysis of fault gases can be performed by comparing their ratio; there are a number of published methods for this, including IEEE and IEC standards [22, 23]. One commonly used diagnostic method is Duval's triangle [24], which uses a triangular plot to represent the ratio of certain fault gases (See Figure 2.5). This visualisation method allows monitoring engineers to plot fault gas ratios for methane (CH_4), ethylene (C_2H_4) and acetylene (C_2H_2), with the different regions representing different suspected fault conditions. For instance, gas ratio results falling in the T3 region indicate a thermal fault in the insulating oil.

While DGA is the standard mechanism for oil-filled transformer monitoring, it does have some drawbacks. The presence of fault gases may not indicate



PD	Partial discharge
T1	Low-range thermal fault (below 300 °C)
T2	Medium-range thermal fault (300-700 °C)
T3	High-range thermal fault (above 700 °C)
D1	Low-energy electrical discharge
D2	High-energy electrical discharge
DT	Indeterminate - thermal fault or electrical discharge

Figure 2.5: Duval's triangle with fault codes [24]

a serious fault condition as gases are naturally present in mineral oil and contamination may occur from gases produced by normal arcing processes in the OLTC. Therefore, for online DGA monitors, only step-changes in dissolved gas levels are indicative of the onset of faults [25].

2.4.1 a) SF_6 gas analysis

Gas analysis methods have been developed to detect the presence of defects in SF_6 insulated systems. As with the DGA method for oil-filled transformer fault gas analysis, the ratio of different fault gases generated by arcs and discharges in GIS can be used as an indicator of the type of defect present. Gas chromatography is used to measure levels of SF_6 decomposition products, however in practice fault gases are so greatly diluted by SF_6, gas analysis methods are rendered too insensitive [26]. Therefore, PD measurement is the preferred method for fault detection in GIS [27].

2.4.2 PD detection

Partial discharge arise as a result of localised ionisation in the insulation medium surrounding a high-voltage conductor, and can occur in solid, liquid or gaseous insulation, consequently occurring in all high-voltage equipment. These defects may be introduced during the manufacturing process or alternatively occur throughout the operational lifetime of the plant. Occurring at the high voltage or earth conductors, PD can indicate the presence of faults long before equipment failure. They can increase in both intensity and frequency over time, eventually leading to internal arcing which in some cases can result in the loss of the asset.

A number of different insulation defects are known to generate PD. Typical defects in GIS include (See Figure 2.6) [28]:

Protrusions: Sharp metallic protrusions on a conducting surface;

Particles: fixed to the surface of an insulator;

Free metallic particles: within the GIS enclosure, which are mobile due to the alternating EM field; and,

Electrically floating parts: conducting components that have become disconnected from a conductor so that they have a capacitive (floating) potential.

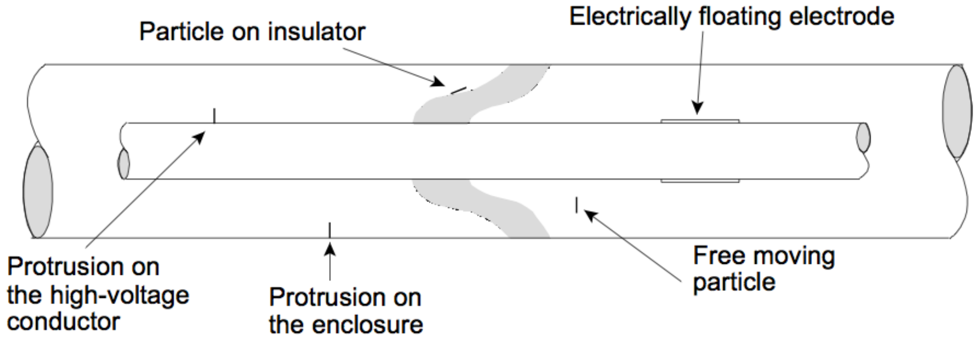


Figure 2.6: Typical PD-generating insulation defects in GIS [28]

In transformers, the insulation system is more complex, therefore there are a wider range of defects which are known to generate PD, including [29]:

Protrusions: sharp metallic protrusions on a conducting surface;

Floating electrodes: where a metallic component has become electrically disconnected from a conductor;

Bad contacts: intermittently floating metallic parts, such as electrostatic field foil components;

Free moving particles: transported within the insulating oil, migrating around the tank;

Fixed particles: attached to the surface of paper insulation;

Surface discharges: where moisture from cellulosic degradation or ingress water has formed on the surface of paper insulation; and,

Void discharges: where gaseous voids occur in solid insulation, or bubbles occur within liquid insulation.

When measuring the occurrence of partial discharges, it is not the actual discharge that is measured but rather the PD pulse which is a 'shockwave' emitted from the defect site. This can be measured by means of the apparent charge induced at nearby terminals, or the acoustic or RF wave propagating from the PD source (both methods are discussed later in this section). There are a number of established measurement and diagnostic techniques used by utilities, of which the most commonly used are described below.

2.4.2 b) Electrical Method: IEC 60270

IEC standard 60270 is an international standard for PD measurement in high voltage equipment [30]. PD pulse measurement is achieved by attaching a capacitively coupled sensor to the terminals of a single phase of the plant, and measuring the 'apparent charge' equivalent of the PD pulse, expressed in picocoulombs (pC). The apparent charge is representative of the PD pulse: if it was injected across the terminals of the plant under observation it would generate the same measurement as the PD pulse itself. A number of different PD pulse distributions can be created from IEC data for the purpose of diagnosis [28], based upon:

ϕ: the electrical phase.

q: the apparent charge.

n: the number of pulses occurring within a particular phase window.

These include: ϕ-n, which is a 2-dimensional distribution of PD occurrence against phase; ϕ-q, which is a distribution of apparent charge against phase, and; ϕ-q-n, which combines the previous two methods to give a 3-dimensional view of activity, with apparent charge plotted against phase angle across a 1-second period [31]. The ϕ-q-n plot is typically referred to as a phase-resolved partial discharge (PRPD) plot.

A number of automated diagnostic techniques have been applied to PRPDs, including the use of fuzzy classifiers [32], support vector machines (SVM) for identification of multiple PD sources [33] and a knowledge-based approach [34].

The IEC technique is best suited to off-line testing in a laboratory or at the manufacturing stage to verify the insulation strength before new plant is dispatched to customers, as the level of electrical noise commonly present within substations means that the technique does not perform well in the field (although this has been addressed in part through noise rejection methods [35]).

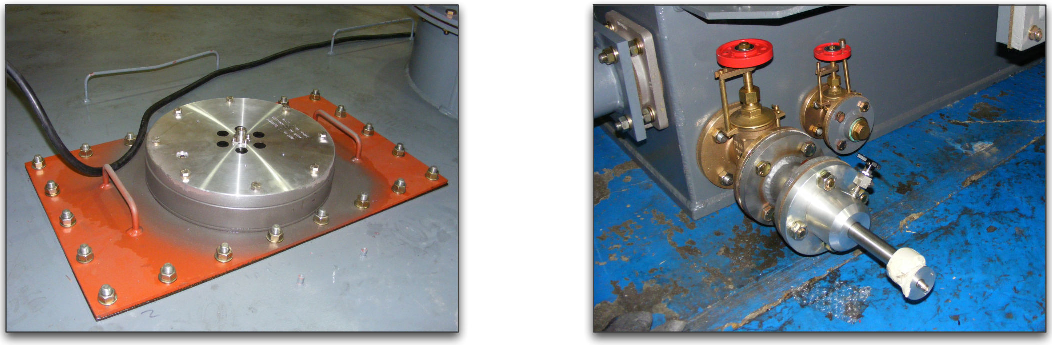
2.4.2 b) Acoustic method

The acoustic PD detection method measures PD pulses in the ultrasonic range, between 20kHz and 1MHz. This method, first employed over 30 years ago [36], is not subject to the electrical noise problems of the IEC method because the measurable ultrasonic wave emission is not affected by strong electric fields, however piezoelectric sensors can also be subject to large amounts of environmental noise within substations. Recent advances in the technique have used advanced signal processing to reduce susceptibility to noise [37, 38]. Time-of-flight information from multiple sensors can be used for defect location in transformers, although issues exist due to the variation in velocity of the ultrasonic wave, as propagation in metallic structures is faster than in insulating oil which can lead to errors in the localisation process.

2.4.2 b) UHF method

The UHF technique captures RF emissions from PD in the UHF band (300MHz to 3GHz), resulting in less susceptibility to noise than the IEC and acoustic methods as the electrical interference spectrum is concentrated at lower frequencies. However, despite being more immune to noise than the IEC and acoustic methods, RF PD pulse measurements are still affected by several factors including the relationship between PD source and sensor in terms of geometry and distance, and signal reflection, refraction and attenuation. These factors must be considered when applying the UHF diagnostic method.

The UHF method uses externally mounted sensors attached to plant at specific points, creating an electrical aperture through which EM emissions can be measured. In GIS, sensors can be attached to dielectric barriers or retrofitted onto visual inspection ports, however new GIS installations are more commonly provided with integrated UHF sensors. In transformers, UHF sensors can either be fitted onto existing inspection hatches or alternatively inserted through oil ports [39], examples of which can be seen in Figure 2.7.



(a) An example of a UHF sensor mounted on top of a transformer

(b) An example of a UHF probe sensor mounted on a gate valve

Figure 2.7: Examples of UHF sensors on power transformers

The UHF technique was initially applied to GIS [40], resulting in an online monitor capable of identification and location of PDs made possible by the waveguide effect of the GIS coaxial bus ducts [26, 41]. This method was subsequently applied to oil-filled transformers [42, 43], where the use of multiple sensors and time-of-flight measurement of PD pulses resulted in accurate location of defects [44]. The versatility of this technique has been demonstrated through its application to HVDC reactors [45], and has been widely adopted for continuous online monitoring of GIS with one particular commercial implementation installed in over 80 transmission level substations worldwide [46].

Some interpretation methods are similar to that of the IEC method; for example phase-resolved plots can be used to aid in diagnosis. However, the UHF method does not measure apparent charge; in this case, PD signal magnitude is measured, and phase-resolved plots consist of relative signal magnitude plotted against phase and cycle for a 1-second period (See Figure 2.8).

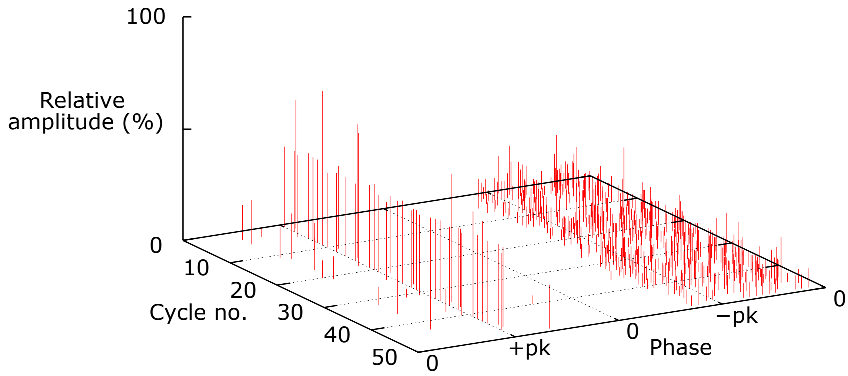


Figure 2.8: A phase-resolved PD pattern based on UHF signals

While experts in PD analysis can manually diagnose the type of defects from visual inspection of a phase-resolved plot, the volume of PD data generated from online diagnostic systems makes the task infeasible across an entire fleet of plant. A number of intelligent diagnostic techniques have been applied to aid in the diagnostic process. These include the use of data-driven classifiers [47] and knowledge-based systems [34].

The chromatic technique has also been applied to RF partial discharge diagnostics [48], which uses three filters on the original source signal represented by the three additive primary colours: red, green and blue. The relative hue

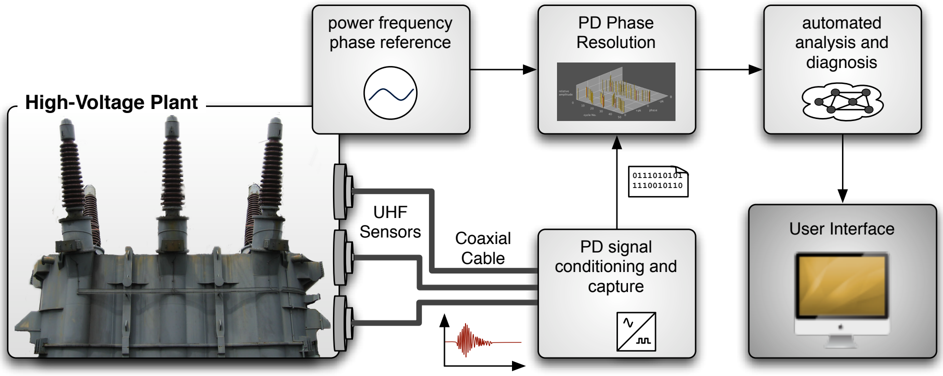


Figure 2.9: Principles of an RF PD monitoring system

(H), lightness (L) and saturation (S) of these signals are then plotted on two polar diagrams (H-L, and H-S), which can be used to establish patterns that may be indicative of, in this case, partial discharge. The chromatic method has also been successfully applied to dissolved gas and circuit breaker analyses [49, 50], however compared to other methods, it is primarily used for visualisation and identification of interesting features rather than a diagnostic tool.

Localisation of PD sources can also be achieved through the analysis of time-of-flight data using multiple UHF sensors [44]. This has obvious benefits to utilities as it allows defects to be diagnosed and assessed in-service without taking the plant out of service and performing an invasive internal inspection.

In addition to analytical methods based upon phase-resolved PD plots, other approaches have been applied to UHF signal interpretation. The presence of multiple concurrent PD sources presents a problem in analysis for phase-resolved plot based methods as distinguishing features can be obscured because the two data sets are overlaid. Possible pre-processing solutions have been proposed, such as spectrogram analysis and envelope analysis [51], which could allow separation of the data streams.

The UHF method, coupled with advanced diagnostic capabilities, represents the state of the art in PD diagnostics. End-to-end UHF PD monitoring systems comprise the 4 components of a condition monitoring system discussed previously: sensors, data acquisition, fault identification and defect diagnostics. This architecture, shown in Figure 2.9, captures, stores and diagnoses PD pulse data providing automated support to maintenance engineers. Field demonstrations of this type of system have successfully diagnosed and located PD faults within in-service transformers prior to an outage [52].

Portable PD monitoring units are available[2] which can be transported be-

[2]From manufacturers such as DMS (`http://www.dmsystems.co.uk/`), Doble Lemke

tween sites to perform on-site diagnostics, however such systems are expensive, require a human operator on-site, and require a mains power source for online operation. Despite such systems being portable, the lack of a monitoring communications infrastructure often prohibits widespread permanent installations in remote sites. Therefore, to-date, deployments of the UHF technique on transformers are typically limited to ad-hoc installations on critical plant items which are suspected or known to be undergoing a serious level of PD. This demonstrates an industry need for low-power, distributable PD monitors that can take advantage of the RF technique while being deployed in-situ within substations.

2.4.2 b) RF frequency analysis

RF PD emissions are known to exhibit different frequency characteristics for different defects [28, 53]. Based upon this property, it is possible to analyse the measured frequency spectrum of PD emissions to extract defect-specific information which may be used for classification. As mentioned previously, tank geometry and propagation affects affect the measured EM pulse and hence the observed frequency components of PD emissions. In oil-filled transformers, defect and sensor position have a significant effect due to the complex structure of the internal components.

While these propagation factors influence UHF signals, certain defects exhibit significantly different frequency spectra. For instance, in [53], free particle defects were found to have a high intensity and high amplitude across the full frequency spectrum, while protrusions were found to have a low amplitude and a higher intensity at low frequencies, with floating electrode defects having a significantly higher intensity than particle and protrusion defects. The frequency spectrum of a PD emission was also found to be consistent over time. Further work on the frequency-based analysis of PD pulses is detailed in Chapter 5.

2.4.3 Other physical monitoring

As well as online tests for insulation monitoring, other techniques can be applied to measure the different parameters relating to plant health such as temperature and vibration. These types of sensors are generally suited to plant that

(`http://www.doble-lemke.eu`) and Power Diagnostix (`http://www.pd-systems.com/`)

undergo mechanical or thermal stresses, such as busbars, transformer cores, fans and pumps.

2.4.3 c) Temperature monitoring

Temperature monitoring within substations has many applications, and has been recommended as a prerequisite for all substation condition monitoring installations [14]. Transformer temperature monitoring is in widespread use, as temperature rises in transformers can cause chemical processes in the insulation system which can lead to the onset of PD. These temperature rises can be caused by overloading, or through problems in insulating oil circulation, possibly caused by pump or fan faults or the presence of foreign contaminants in the oil.

Within power transformers, the 'hot spot temperature' is the hottest part of the winding of a transformer, and is the limiting factor for loading the transformer. The hotspot temperature of a transformer should not exceed $110\,^{\circ}\mathrm{C}$ to minimise premature ageing through overheating of the paper-oil insulation system [54].

The hotspot temperature can be measured directly using fibre-optic temperature sensors [55], although this method is expensive and can only practically be installed during manufacture [14]. The most commonly used method uses externally mounted sensors to measure top- and bottom-oil temperature, which can then be combined with load current data as inputs to a standard thermal model [56, 57] . In addition to measuring transformer oil temperature, it is also necessary to measure ambient temperature as unexpected thermal conditions seen in transformers may be caused by adverse environmental conditions.

2.4.3 c) Vibration monitoring

Within power transformers, the magnetic forces caused by fault currents can cause core and winding deformations, which represent up to 15% of all transformer faults [14]. To identify incipient faults introduced by these deformations, vibration monitoring can be applied. Diagnostic methods have included the use of artificial neural networks (ANNs) to estimate vibration levels and automate classification of winding defects [58], and model-based reasoning, where a model was used to calculate transformer deformations based upon the comparison of observed and expected vibration [59, 60]. Vibration monitoring has also been applied to on-load tap changers (OLTC) monitoring [61]. Other components including fans and pumps are also monitored for vibration [62], but this practice is not widespread.

2.4.4 Integrated circuit breaker monitoring

Several commercial online monitoring systems for circuit breaker monitoring and diagnostics have been developed that integrate a number of sensors into a single Intelligent Electronic Device (IED). These are designed to measure a number of circuit breaker parameters to enable monitoring engineers to identify incipient faults. To illustrate the types of measurements used for circuit breaker monitoring, consider the circuit breaker parameters measured by the Areva T&D CBWatch-2 circuit breaker monitoring system [63], which represent a significant subset of the possible circuit breaker parameters which may be measured:

- SF_6 pressure, temperature, density and leakage (for SF_6 insulated plant);
- close and open operating times;
- primary contact separation speed;
- mechanical performance deterioration;
- hydraulic or spring operating mechanisms; and,
- I^2t—a function of the accumulated current during arcing and the duration of the arc.

IEDs deployed in the field for circuit breaker monitoring have been shown to reduce costs by eliminating routine tests, allowing circuit breaker maintenance to move towards a condition-based scheme [64]. Online circuit breaker monitoring systems such as CBWatch-2 generate extensive monitoring data, however they do not necessarily provide diagnosis of the actual fault condition which a circuit breaker may be undergoing, or may undergo in the future. This is a recognised issue which has been addressed previously in the literature.

Strachan *et al.* [19] have implemented a circuit breaker diagnostic method which used five features of trip coil current signals along with a number of statistical classification techniques to generate clusters representing various breaker conditions. This knowledge was subsequently encoded into a decision tree which may be used for on-line diagnosis of circuit breaker conditions.

In [65], the authors implement a new type of IED for circuit breaker monitoring which transmits a combination of 15 analog and digital signals to a concentrator PC running a diagnostic system comprising an advanced signal processing and an expert system. The authors' method is capable of classifying circuit breaker events, verifying circuit breaker operation, and identifying breaker fault conditions. The system uses broadband wireless communications to communicate between the IEDs and concentrator PC running the diagnostic system. This minimises instrumentation wiring, however as each IED requires 1.4Mbs^{-1} bandwidth for continuous monitoring of the 15 data streams, the system may have issues with scaling in large deployments which could be addressed by performing diagnostics directly on the IED.

2.5 Sensors

As highlighted previously in Section 2.2, sensors are an integral part of any condition monitoring system. It follows that, for new equipment, sensors should ideally be included during the manufacturing stage so that early onset failures may be identified quickly, and the state of plant health may be monitored from day one. However, as one of the main benefits of condition monitoring is the postponement of asset replacement, it also follows that as condition-based maintenance is increased, the majority of condition monitoring sensors will—and must—be fitted to in-service assets. For this to be practical, new condition monitoring sensors must be minimally intrusive, where sensors can be attached to the periphery of plant on-site, either on or through the outer wall at convenient access points. An example of this is UHF PD sensors installed through transformer oil drain valves (See Figure 2.7(b)), as recommended by CIGRÉ [66].

A different approach to this issue has been to use remote wireless sensing for PD diagnostics, where PD emissions are detected at a central point for an entire substation using a four-antenna array. In [67], this method successfully located a defect in the tertiary winding of a transformer, however it could not identify the specific phase in which the fault was located.

For utilities, this approach to PD monitoring is advantageous as, by being completely non-intrusive, it provides wide coverage at a lower cost than individual PD monitors. However, this method is limited in identifying the exact location of defects and therefore acts as an early warning system for an entire substation identifying items of plant which may need closer investigation, for instance using other techniques such as the UHF method.

2.6 Discussion

In this chapter, a number of substation condition monitoring techniques commonly used by electrical utilities have been described. If utilities are to extend condition monitoring coverage further in the future, a number of issues must be addressed: firstly, how can condition monitoring systems effectively scale-up if sensor coverage is to increase, avoiding 'data overload' to both communications networks and monitoring engineers, and secondly; how can multiple heterogeneous sensors and monitoring techniques be integrated into a cohesive, unified view of substation health, where automatic interpretation capabilities convert complex datasets into useful monitoring information.

In the next chapter, we will see that multi-agent systems and artificial intelligence (AI) techniques have been successfully applied to these issues to create flexible, extensible diagnostic systems. In Chapter 5, an industrially relevant application will be investigated to establish whether low-power diagnostic techniques required for wireless sensor network applications are feasible.

2.7 Chapter bibliography

The following sources were used as bibliographic sources for this chapter, in addition to the inline references:

- D. Warne, *Newnes electrical power engineer's handbook*. Elsevier, 2nd ed., 2005
- T. K. Saha, "Review of Modern Diagnostic Techniques for Assessing Insulation Condition in aged Transformers," *IEEE Transactions on Dielectrics and Electrical Insulation*, vol. 10, pp. 903–917, Oct. 2003
- M. J. Heathcote, *The J&P Transformer Book*. Newnes, 12th ed., 1998
- M. Wang, A. Vandermaar, and K. Srivastava, "Review of condition assessment of power transformers in service," *IEEE Electrical insulation magazine*, vol. 18, no. 6, pp. 12–25, 2002

- R. James and Q. Su, *Condition Assessment of High Voltage Insulation in Power System Equipment*. Institution of Engineering & Technology (IET), 2008

Chapter 3

Multi-agent systems

"The future is here. It's just not widely distributed yet."

- **William Gibson, 1948 -**

3.1 Overview

Software agents are an often misunderstood or misrepresented subject, therefore when discussing them it is necessary to define key concepts to create a common point of reference. The IEEE Power & Energy Society Multi-Agent Systems Working Group has done just that, producing a two-part guide to the technology behind multi-agent systems, and their application within the power domain [70, 71]. This chapter summarises and extends the contents of these papers, focusing on the aspects of multi-agent systems that are key to substation condition monitoring applications with a view to extending these systems down to the sensor level with wireless sensor networks.

3.2 Software agents

Fundamentally, software agents are a software development methodology that provides suitable high-level abstractions with which to build complex distributed software. However, the key attribute that differentiates agents from other software methodologies is the concept of *autonomy*. Wooldridge and Jennings, two pioneers of the agent research field, define software agents as a piece of software that operates within an environment, which is able to react autonomously to the environment to meet its design goals [72].

Extending this concept further, Wooldridge defines an *intelligent agent*, as one that exhibits 'flexible autonomy', which can be can be recognised by three fundamental characteristics:

Reactivity: where agents react to changes in their environment, either from physical changes observed through sensors, or through communication with other agents;

Pro-activeness: where agents exhibit goal-directed behaviour; an intelligent agent will strive to execute one or more goals to fulfil its design requirements; and,

Social ability: where agents operate as part of a community, acting cooperatively via message-based communication.

3.3 Multi-agent systems

A multi-agent system (MAS) is defined as a software system consisting of two or more agents. Within an agent society, agents are programmed to have individual goals, however, through social interactions the collective individual goals of agents may serve to attain an implicit global goal. For example, within a substation, the goal of each agent may be to monitor a separate individual plant item. Together they serve to monitor the entire substation, however this goal is never explicitly defined. This has certain advantages; as software agents are modular they can be easily deployed in different configurations depending on the application. They can be duplicated or replaced easily without impacting on the global operation of the system. For diagnostic applications, these attributes make MAS technology a natural abstraction for integrating multiple intelligent techniques within a single system.

3.3.1 Agent platforms

FIPA, the Foundation for Intelligent Physical Agents, is an IEEE Computer Society organisation working for the standardisation and dissemination of agent-based technology [73]. FIPA publishes reference standards defining interfaces and components required to build interoperable multi-agent systems, also specifying inter-agent communication protocols, patterns and semantic languages. FIPA-compliant multi-agent systems have become popular as the use of well-defined and well-supported standards fosters interoperability and simplifies the design and construction of industrially robust systems.

An agent platform is a standard model within which software agents operate [74]. The FIPA agent management specification defines a standard logical architecture for agent platforms, and the mandatory and optional components that should be included to ensure compatibility between systems. Shown in Figure 3.1, the FIPA agent management reference model includes:

Agent management system: A mandatory component of the agent platform, the agent management system (AMS) provides supervisory control over agents operating within the agent platform;

Directory facilitator: An optional component providing 'yellow pages' services. A directory facilitator provides a directory of agent services, allowing agents to discover each other without any prior knowledge, by function rather than hardcoded address; and,

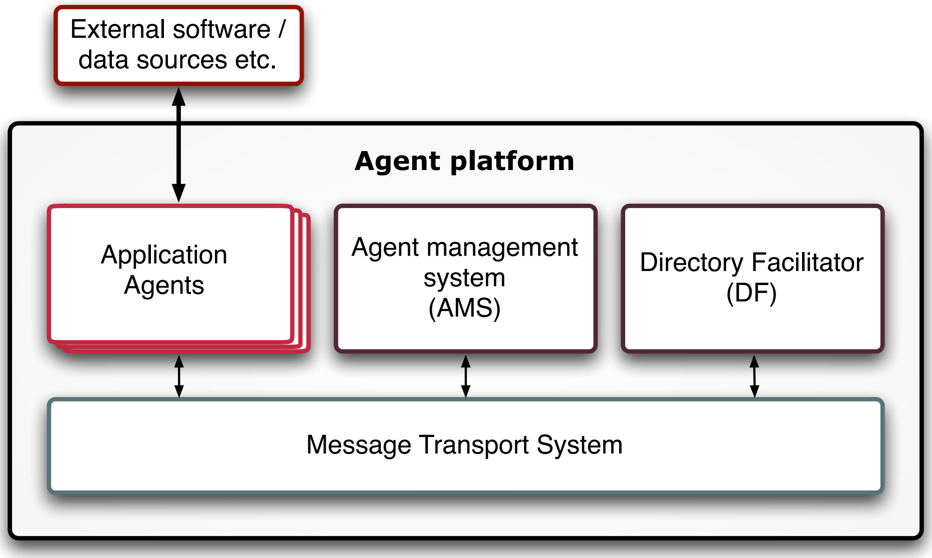


Figure 3.1: The FIPA agent management reference model

Message transport service: The communications channel between physically separated agents on different hosts.

This model specifies a logical structure for agent platforms, but leaves functional details to individual implementations. One such implementation, JADE- the Java Agent Development Environment, has become the most widely used platform for FIPA multi-agent systems development in recent years, because of its maturity [75], complete documentation [76], and intuitive approach to constructing agents.

3.4 An overview of agent technology

3.4.1 Communication

One of the key attributes of an intelligent agent is an agent's social ability, which is drawn from the capacity to communicate effectively with its peers. With roots in speech-act theory, agent communication differs from traditional object-oriented message-based communications as agents cannot only share knowledge as with traditional messaging techniques, but also signify *intent* in their messaging. FIPA-compliant agent messaging is split into two primary components: communication language and content language. Communication languages, such as FIPA-ACL (FIPA Agent Communication Language) [77], are used to define the type of message, or *intention* of the sender, whereas

Table 3.1: Subset of commonly used FIPA communicative acts [80]

Performative	Description of communicative act
Agree	The action of agreeing to perform some action
Failure	The action of telling another agent that an action was attempted but the attempt failed
Inform	The sender informs the receiver that a given proposition is true
Not Understood	The sender informs the receiver that it could not understand the receiver's previous action
Query Ref	The action of asking another agent for an object referred to by a referential expression
Refuse	The action of refusing to perform a given action
Request	The sender requests the receiver to perform some action

content languages, such as FIPA-SL (FIPA Semantic Language) [78] are used to define message syntax. In addition to these components, FIPA standards provide other mechanisms for agent communication which combine to create a rich framework for inter-agent messaging. To illustrate how this is achieved, the components will each be described in turn.

3.4.1 a) Communication Language

The communication language provides a standard message structure for agent communication, consisting of 13 fields for message addressing and conversational control as well as the message content [79]. The only mandatory field within an ACL message is the performative, representing one of 22 communicative acts [80]. An 'ideal agent' should be able to understand all 22 possible performatives, however, the engineering requirements to implement all 22 performatives typically leads to agent designers only implementing a small subset. To illustrate, commonly implemented performatives are described in Table 3.1.

3.4.1 a) Interaction Protocols

To provide structure to agent conversations, a number of interaction protocols have been defined by FIPA to act as patterns for agent message flow [81]. In Figure 3.2, the FIPA Request Interaction Protocol [82] is shown, which governs the message flow for a conversation based upon an initial `request` message.

Here, we can see the possible responses to the initiator of the conversation after a `request` message is sent. An agent supporting this message protocol can expect a response containing one of the performatives listed, and should be capable of acting appropriately. For instance, if an `agree` response is received, the initiator must wait for a further response (either `failure`, `inform-done` or `inform-result`). Alternatively, if a `refuse` response is received, the initiator may wish to start a new conversation with a different agent to achieve the desired outcome. Outside of the main protocol communication band, the receiver may send a `not-understood` communication if, for instance, the received message was syntactically or semantically incorrect and could not be understood.

3.4.1 a) Content Languages

In addition to the performative, each ACL message contains optional message `content` and `language` fields, which hold the message payload and describe the semantic language used. FIPA has standardised four content languages; FIPA-SL (Semantic Language) [78], KIF (Knowledge Interchange Format) [83], CCL (Constraint Choice Language), and RDF (Resource Definition Framework).

OWL, the Web Ontology Language [84], has also been proposed as an additional ACL content language [85]. This format, based upon RDF and endorsed by the World Wide Web Consortium (W3C) [86], has become the de-facto standard for web-based ontologies, and its support as a communication language is intended to ensure interoperability with web-based systems.

Due to it being the first standard to reach maturity, FIPA-SL emerged as the content language of choice within the power engineering community [71]. FIPA-SL is an implementation of first-order logic, a method of representing knowledge based around objects and relations, based upon the s-expression syntax commonly associated with Lisp [87]. At its core, FIPA-SL can be used to make a proposition, request that an action is performed, or identify an object within the domain of discourse. By combining these cases in various config-

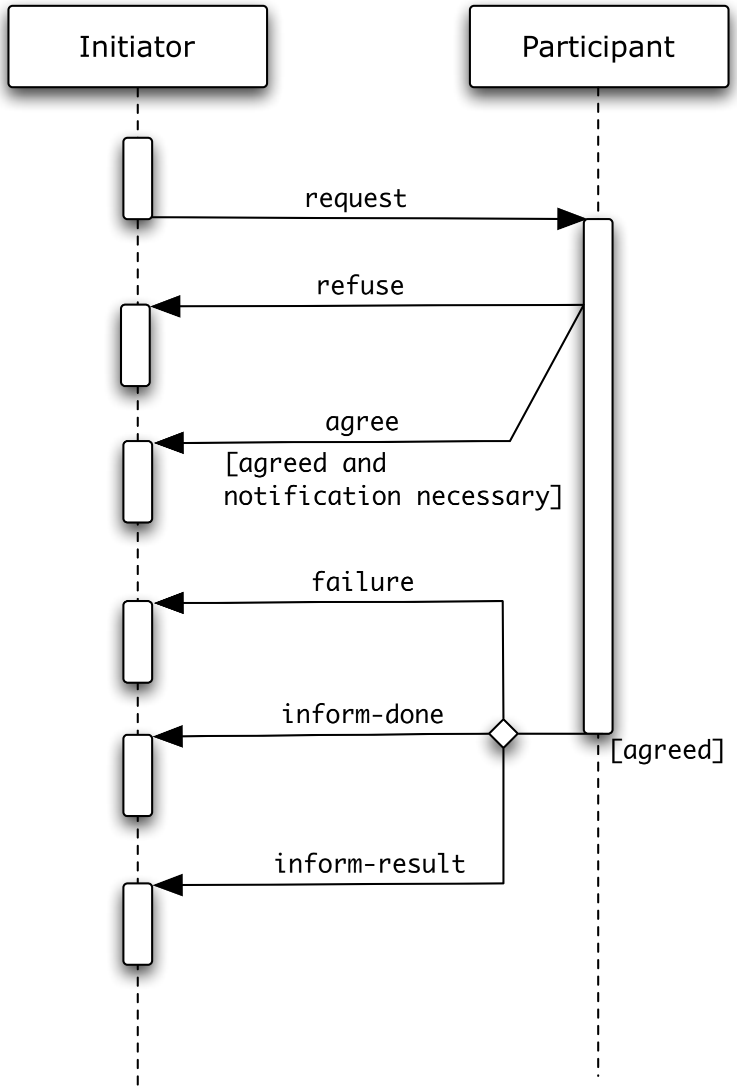


Figure 3.2: FIPA request interaction protocol representation [82]

```
(REQUEST
    :sender (agent-identifier
        :name gas-provider@agents.eee.strath.ac.uk)
    :receiver (set (agent-identifier
        :name duvals-agent@agents.eee.strath.ac.uk))
    :content "((action (agent-identifier
        :name duvals-agent@agents.eee.strath.ac.uk)
            (CalculateDuvalsTriangleFault
                :C2H4 (DissolvedEthylene :ppm 90)
                :C2H2 (DissolvedAcetylene :ppm 18)
                :CH4 (DissolvedMethane :ppm 43))))"
    :language fipa-sl
    :protocol fipa-request)
```

Figure 3.3: An example of a FIPA REQUEST message

urations, complex expressions can be constructed [78]. For instance, a `not-understood` message contains a tuple consisting of the original action and a proposition representing the reason for the failure [80].

An example of a FIPA-ACL `request` message using FIPA-SL as its content language is shown in Figure 3.3.

This message represents a request from a 'gas provider' agent to a Duval's triangle agent to identify a fault based upon the given rise in gas levels. The message opens with the performative, `REQUEST`, followed by a list of message fields. For the receiving agent to be able to form a valid response to the message, it must be able to:

- Understand the `fipa-request` protocol, defined in the `protocol` field;
- Understand the `fipa-sl` content language, defined in the `language` field; and,
- Parse the `content` field, supporting the `CalculateDuvalsTriangle-Fault` action.

From the message above, `CalculateDuvalsTriangleFault`, `Dissolved-Ethylene`, `DissolvedAcetylene` and `DissolvedMethane` are domain-specific terms, not defined within the FIPA protocol framework, but rather in an agent ontology. If the FIPA protocols described represent the message syntax, it is the agent ontology that represents the message semantics: the common domain model shared between agents.

Before moving on to agent ontologies, it is necessary to digress for a moment to discuss FIPA message representations. This section of the FIPA specifications is often overlooked, however it is an important factor relating to this thesis as the message encoding determines the resultant size of a message, which will be shown in the next Chapter as an important issue for wireless communications.

3.4.1 a) Message Representation

FIPA agent message representations govern the encoding used to send the message. For ACL message representations, FIPA has defined the following three representation specifications [79]:

- FIPA ACL Message Representation in Bit-Efficient Specification;
- FIPA ACL Message Representation in String Specification; and,
- FIPA ACL Message Representation in XML Specification.

The ACL message shown in Figure 3.3 is represented in string format, meaning that, when it is sent across a communication link, it is sent verbatim in the human-readable form shown here. Alternatively, it is possible to represent the message in eXtensible Markup Language (XML) form, which may be beneficial for web service integration [88], or bit-efficient form, where message symbols such as 'sender' and 'receiver' in the default string format are replaced by hexadecimal codes, significantly reducing the message overhead. Hex codes are predefined for all keywords within the FIPA specifications, although these do not cover any application-defined concept or slot names. For the bit-efficient protocols to extend to these symbols, it is necessary for agent hosts to keep a local code table.

The code table operates as follows: each time a message is sent which includes a previously untransmitted string, it is added to the code table at the sender and transmitted in its original string form. When it arrives at the receiver, it is recognised as previously unseen, and added to the receive code table. Subsequently, when the word is to be retransmitted, the code table key is sent in its place, to be re-substituted upon arrival at the receiver. A summary of an example code table is shown in Table 3.2. This table contains all of the application-specific words within the example ACL message in Figure 3.3, using a 16-bit code size represented in hexadecimal form.

Once a string has been entered into the code table and transmitted to the sender, it can subsequently be substituted with the escape character `0x15` fol-

Table 3.2: A FIPA bit-efficient code table

Code	Encoded Word
`0x0000`	gas-provider@agents.eee.strath.ac.uk
`0x0001`	duvals-agent@agents.eee.strath.ac.uk
`0x0002`	CalculateDuvalsTriangleFault
`0x0003`	C2H4
`0x0004`	DissolvedEthylene
`0x0005`	ppm
`0x0006`	90
`0x0007`	C2H2
`0x0008`	DissolvedAcetylene
`0x0009`	18
`0x000A`	CH4
`0x000B`	DissolvedMethane
`0x000C`	43

lowed by the code table index. Therefore:

```
CalculateDuvalsTriangleFault (28 bytes)
```

becomes:

```
0x0015 0x0002 (4 bytes)
```

In [89], Helin *et al.* carried out a study of ACL message encodings, comparing the size of messages using 5 different schemes, including the three FIPA schemes plus compressed String (using the ZIP algorithm) and Java object serialization. Their study was to establish the efficacy of the bit-efficient specification for use in wireless environments, finding that it outperformed all other encodings in terms of message size. However, their results did not take into account the burden of maintaining code tables; this is investigated further in Chapter 4.

3.4.2 Ontologies

An ontology is a formal hierarchical representation of the concepts and relationships within a particular domain [90]. These concepts follow an inheritance model, so for instance within a substation, both *Busbar* and *Cable* concepts may inherit attributes from the *Conductor* concept. Each concept consists of *slots*, which represent the properties of that concept; therefore, for instance, a *Busbar* concept may have a 'rating' slot, in which its operational rating may be defined. This rating, in turn, could be an instance of a *Measurement* concept, whose 'type' slot would be *Volts*. Once defined, an ontology can be used to model instances of the domain knowledge, which together form a knowledge base.

There are a number of benefits in defining an ontology for a particular domain, including providing the ability to share understanding of information between software agents, to make domain assumptions explicit and, probably most important when implementing industrial applications, to enable domain knowledge to be reused between systems [91].

To foster domain reuse and interoperability, a number of standard ontologies have been defined. The largest of these is, quite aptly, named SUMO; the Suggested Upper Merged Ontology [92]. This ontology, built using the KIF standard, has been ratified by the IEEE and brings together many elements from other upper ontologies, such as temporal definitions first defined in [93]. SUMO contains concepts for standard electrical measurements, although it does not contain concepts relating to the domain of power engineering.

Within the power domain, CIM, the Common Information Model is a semantic model describing the components of a power system used to exchange power system information between utilities [94]. CIM has been recommended as an upper ontology for power engineering applications by the IEEE Power & Energy Society Multi-Agent Systems Working Group in [71], and has been partially implemented in [62]. The use of CIM as an upper ontology, paired with the FIPA standards, will allow agents developed by different institutions to communicate and interact with each other, simplifying agent development for power applications.

Using CIM as an upper ontology would allow integration within the power domain, but integrating CIM as part of SUMO may have additional benefits, such as increased reasoning capabilities which may be derived from the explicit definition of standard concepts and processes within SUMO. This is outside the scope of this thesis, but may be of consideration to the agent/power system research community in the future.

One of the issues that would be faced when integrating two such ontologies is the identification of common concepts and the definition of relationships between them. Ontology mapping is the process of translating between multiple ontologies; this is covered within the FIPA standards through the 'FIPA Ontology Service Specification' [95]. An implementation of the Ontology Service has been attempted in [96], however the significant difficulties faced when mapping between ontologies, described in [97], have resulted in no functional implementation of the ontology service becoming available.

3.4.3 Mobility

One area of agent research which has received significant attention is agent *mobility*. Mobile agents are able to traverse a network, migrating from host to host as they see fit to best complete their goals. An introduction to mobile agents is given in [98], with the two key advantages of mobile agents described as the reduction in network traffic, and asynchronous operation. Arguably, these properties are inherent in any software agent, and in fact, a solid use case for mobile agents has yet to be proven.

By far the most referenced paper in the field of mobile agents is entitled "Seven good reasons for mobile agents" [99], which has become the key resource used to justify the use of mobile agents. These reasons are arguable, however, as with further analysis they do not, in fact provide solid proof of the benefits of mobile agents. The seven reasons state that mobile agents:

Reduce network load: The 'mobile agent mantra', 'Move the computation to the data rather than the data to the computation', reiterates that through agents processing data locally at the source, the burden placed upon communication networks can be reduced as only statistical data need be transmitted. While this requires an agent to be situated at the source of the data, it does not require the agent to travel with the processed data. In fact, transmitting the agent along with data will invariably increase the network load;

Overcome network latency: This is not an inherent property of mobile agents, but a property of distributed computing in general [100]. By processing locally, mobile agents can avoid the latency that may occur in the remote transfer of data. At some point, an agent is required to transmit data across a network link, and at this point, the data transfer will be subject to the inherent latency associated with the network;

Encapsulate protocols: Of all the reasons given, 'encapsulating protocols' provides the strongest case for mobile agents. By encapsulating protocols, behaviours or techniques, functionality can be deployed across a network, enhancing or replacing existing capabilities. However, as deployment is only carried out once, the agent itself need not necessarily be a 'mobile agent'; transmission of agent code could be included as a feature of the agent management system;

Execute asynchronously and autonomously; adapt dynamically; are naturally heterogeneous; and are robust and fault tolerant: While all four of these assertions are true, it is because that these attributes are inherited as properties of agents, and are therefore equally applicable to static agents as mobile agents.

In addition, Vigna [101] provides ten more reasons against the use of mobile agents, specifically targeting their bad performance and security. In the literature, mobile agents have not been shown to provide any marked improvement in performance over their static counterparts, and their attributed benefits do not fully justify their use. However, the one redeeming property of an agent platform which supports agent mobility is that it may simplify the deployment of new agent functionality to remote nodes as code management can be achieved within the context of the agent platform.

3.4.4 Agent anatomy

The anatomy of an agent defines the way that it interacts with its environment, though knowledge storage, behaviour execution and messaging. Several approaches to agent anatomy have been proposed, defining models with which to build intelligent agents.

Brooks [102], asserts that by using a purely reactive model for agents, typified in the subsumption architecture, intelligent behaviour will emerge through complex interactions as intelligence is intrinsically embedded within the environment and is subject to the observer. This model decomposes behaviour into a set of hierarchical finite state machines, where each layer represents a particular goal of the agent, combining to obtain a global goal which may or may not be explicitly specified. This model has been successfully applied to autonomous robots [102], however its use outside of this field has been limited due to the difficulty in understanding and building of large-scale subsumption systems [103].

Rao and Georgeff take a different approach in the BDI model [104] which aims to mimic the human reasoning process. BDI agents are created using a composition of *Belief, Desire* and *Intention* mental states, representing the knowledge an agent has about its environment, the agent's goals, and the plans that the agent has constructed to achieve those goals.

The 'layered model' uses an object-oriented approach to agent development where agent functions are split logically and modelled as separate layers. Interactions can either occur horizontally, where data is processed by different layers in parallel; vertically, where data passes up and down between layers, or a combination of both, depending on the application.

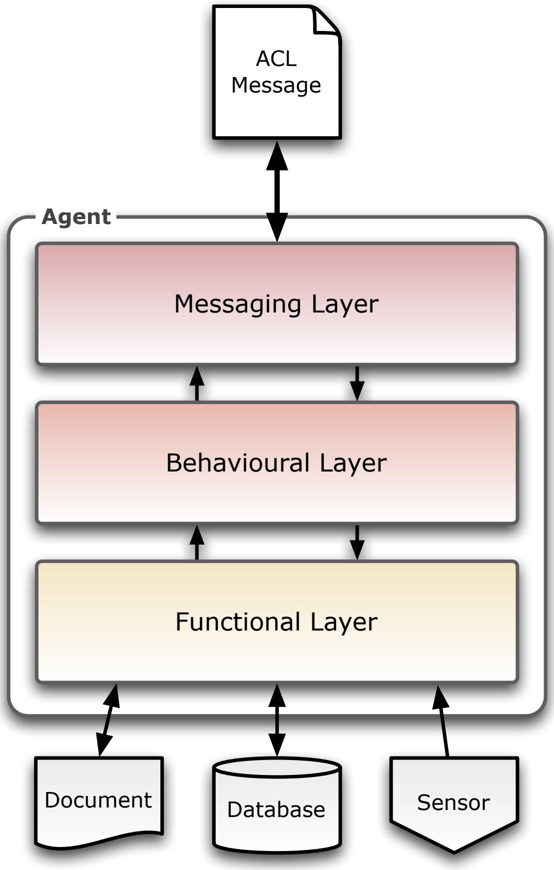


Figure 3.4: The JADE 3-layer model

JADE, discussed earlier, uses a hierarchical three-layered model. These layers consist of message handling, behavioural and functional components, illustrated in Figure 3.4. For condition monitoring applications using this model, functions such as diagnostic techniques are wrapped as agent behaviours which can be run according to a schedule, or based upon external factors such as the reception of a message from another agent, or a new sensor value crossing a predefined threshold. The messaging layer is required to parse and generate messages, converting between object-oriented data stored within the agent and the semantic content languages discussed earlier.

This 3-layered model has gained popularity through the JADE toolkit as it is straightforward to understand, and has become a popular agent anatomy within the power engineering field (illustrated in the following Section). There are, however some pitfalls associated with using this approach. Agents typically use static message parsing to interpret incoming messages, built as a finite state machine using `if-else` statements which step through a fixed process defined by the agent designer (see Figure 3.5).

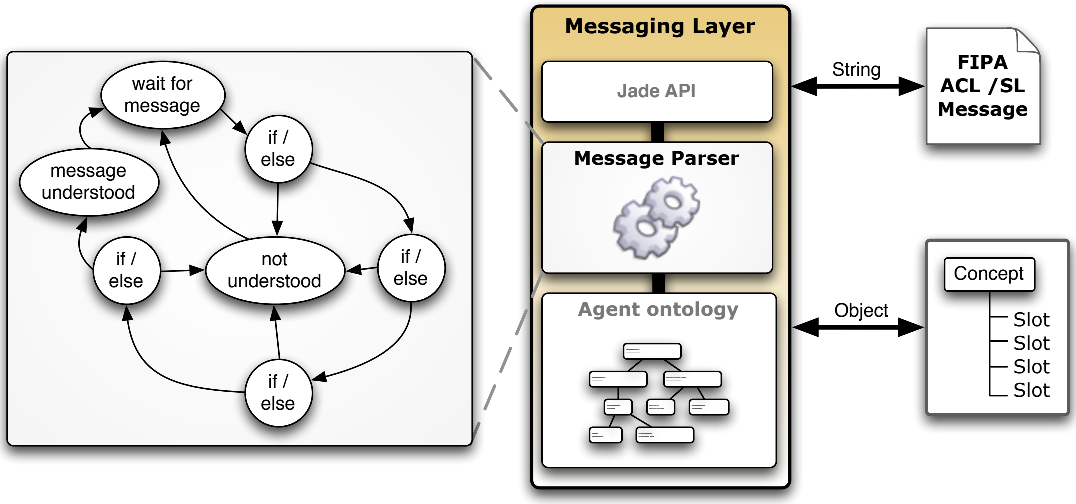


Figure 3.5: Message parsing within JADE using `if-else` structures

The drawback of this method is that if an agent receives a message which is not in the expected format, the static, `if-else` structure falls back to the `not-understood` state, and cannot generate a suitable response. Consider the two well-formed formulae show in Figure 3.6, both intending to resolve the unique plant item being monitored by a wireless sensor node with MAC address `0014:4F01:0000:538C`:

An ideal agent should be able to evaluate both message formats, but an agent using static message parsing may not support the evaluation of an additional atomic term within the message. Of course, it is possible to add addi-

```
(iota ?x
   (hasA (SensorNode
     :macAddress '0014:4501:0000:538C')) ?x)

(iota ?x (and
  (isA PowerSystemResource ?x)
    (hasA (SensorNode
      :macAddress '0014:4501:0000:538C'))) ?x)
```

Figure 3.6: Two differently formed well-formed formula in FIPA-SL syntax representing the same query

tional support to parsers, however as FIPA-SL is based upon first order calculus, these permutations are theoretically infinite. The complexity of SL statements and the disjoint between object-oriented and first-order logic representation typically leads to a minimal set of parsers being implemented by the developer, severely limiting an agent's social and reasoning abilities.

It is obvious that this represents a gap between the idealistic properties of an 'intelligent agent' and the realities of building real-world multi-agent systems using state-of-the-art tools. To bridge this gap, a flexible replacement to the static message parsing methodology is required that can evaluate each atomic term within the message individually. The JADE Semantic Addon (JSA) has been proposed as such a replacement, implementing a BDI-like anatomy and providing full support for the FIPA-ACL semantics [105]. JSA treats FIPA-SL messages as a hierarchy of nodes, evaluating each in-turn using a semantic interpreter. This approach has not yet been used within the power community as it is a relatively new approach to building agents, however there have been significant successes to-date in applying muli-agent technology within the power domain, which will be discussed in the following section.

3.5 Agent technology in the power engineering domain

Multi-agent systems (MAS) have been used for numerous power engineering applications, in areas including monitoring and diagnostics, protection, distributed control, modelling and simulation. In this section, the most noteworthy applications of multi-agent systems in the power domain are discussed. For further background information in this area, refer to the seminal reference papers on these topics which detail findings from the IEEE Power & Energy Society Multi-Agent Systems Working Group [70, 71].

3.5.1 COMMAS

COMMAS, the 'COndition Monitoring Multi-Agent System' is a multi-agent system designed to carry out automated diagnosis of UHF partial discharge measurements [106]. COMMAS has been applied to transformer [106] and HVDC reactor analysis [107], and can operate in both online and offline scenarios.

The system, shown in Figure 3.7, comprises a number of agents which work cooperatively to identify and classify partial discharges, incorporating data-driven classification [47], knowledge-based diagnostics [108] and result corroboration to increase diagnostic accuracy [109]. The system implements a layered architecture, allowing separate layers to have no constraints on their physical location. One advantage of this approach is that only relevant data passes between layers through well-defined interfaces, reducing the load on the underlying communications system.

The higher layers of the system contain the interpretation and corroboration components, which encapsulate the 'intelligent' aspects of the system. The data monitoring stage contains three agents, each of which performs simple reactive tasks on receiving raw partial discharge data. This layer carries out data management by formatting data and extracting statistical features to be processed by the higher levels. Figure 3.7 shows a single PD monitor attached to the system.

This layered approach is a useful tool for organising component functions, however it enforces a structure on the agent system which may impact the flexibility of the system as interactions patterns are well defined between separate layers of the system. A more flexible approach to multi-agent system design is to forgo a rigid structure, instead allowing agents to structure themselves organically based upon their individual goals.

3.5.2 Protection engineering diagnostic agents

Protection engineering diagnostic agents, or PEDA, is a multi-agent system designed to provide analytical and diagnostic support to protection engineers through the automated analysis of SCADA and DFR data [110]. The system uses a multi-agent approach to integrate a number of existing legacy systems, including:

- a rule-based system for digital fault recorder (DFR) data analysis;
- a rule-based system for automatic SCADA data analysis, which assesses the operation of protection systems and is used to focus the analysis of DFR data; and,
- a model-based reasoning (MBR) system for protection operation validation, which uses DFR data along with a protection model to validate the operation of protection equipment by comparing the predicted operation of protection equipment against the actual operation.

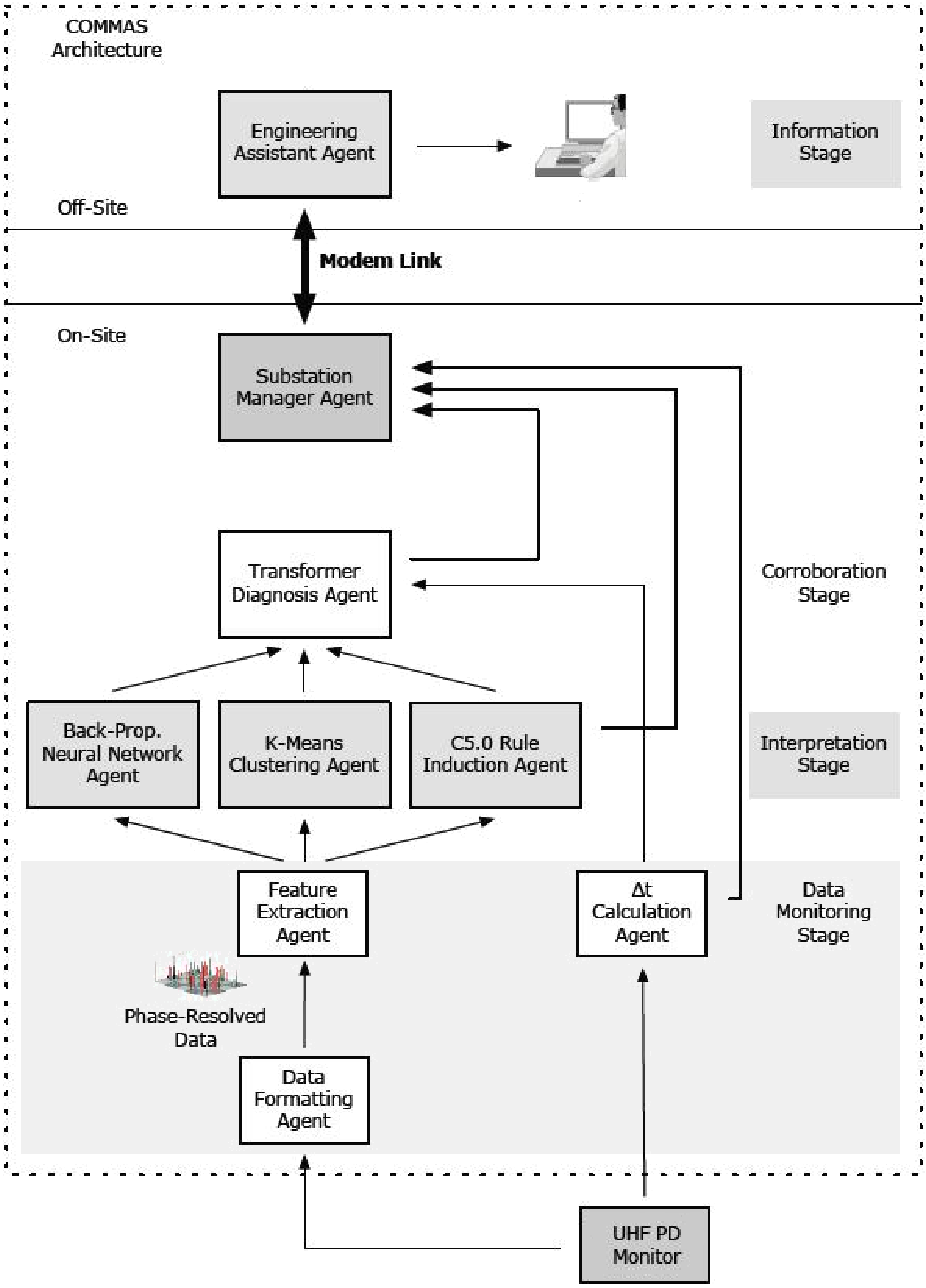


Figure 3.7: The COMMAS architecture [109]

The system is built upon JADE, employing 6 diagnostic agent types in addition to the system agents. By encapsulating each of the diagnostic techniques as an intelligent agent and using a multi-agent approach to integrate the techniques, enhanced diagnostic support is provided to protection engineers. PEDA represents the first full industrial application of a robust multi-agent system within the power industry operating continuously over a 6 month period [110]. This demonstrates that it should be used as a reference model for other power industry multi-agent systems.

3.5.3 Substation automation

Buse *et al.* demonstrate the use of multi-agent systems for substation automation [111]. Agents are used to wrap individual IEDs, databases, diagnostics and user agents, providing increased reliability as each agent can act autonomously within the system. Substation communication links are low bandwidth and unreliable, so this approach is more tolerant to communications failures than a traditional client-server approach.

Within this system, agent messaging is used to control switchgear operations. A mathematical analysis of mobile agent performance finds that, for the substation automation application, mobile agents perform better than static agents when sending more than 30-40 control messages between two nodes in a single session. If messaging sessions require less than 30 messages, then a static agent approach outperforms the mobile agent approach in terms of latency. This also assumes that this volume of interactions must be sent as separate messages; it may be possible to encode multiple commands as a single message.

The authors also suggest that under certain circumstances the use of mobile agents may introduce power network instability, and that additional static proxy agents may be used to mitigate this. Although the authors do find situations where the use of mobile agents may be advantageous, the benefits are tenuous and do not warrant the increase in system complexity.

3.5.4 Mobile agents for improved circuit breaker maintenance

In [112], Kezunovic *et al.* propose the use of mobile agents to improve circuit breaker maintenance strategies. When collating circuit breaker failure reports, data from multiple sources is required and to collate this information, mobile agents are proposed to visit each of the data sources in turn, collecting and collating information.

In this application, mobile agents are used to reduce the burden of data transfer between sources, however, as we have seen in the arguments against mobile agents earlier in this chapter, the added complexity of mobile agents is unnecessary. As each of the data sources must have an attached agent container in which the visiting mobile agent executes, there is no reason why one or more static agents could not be permanently situated at each source providing access to and processing of the data.

This would further reduce network bandwidth as the agent code would no longer have to be transmitted twice to receive the required data; instead a message could be sent to the data source wrapper agent, and only the required response need be transmitted back. This would achieve all the benefits sought by using agents without the extra overhead and complexity of introducing agent mobility.

This application is well suited to an agent-based approach to provide an extensible distributed platform for data aggregation, although in this case, there is no benefit of using agents with mobility except for dynamically updating code.

3.5.5 Power system restoration

Power system restoration is the process of reconfiguring a network to the optimal state after a fault has occurred, to ensure that any disconnected generation and load can be reconnected as quickly as possible. A number of multi-agent approaches have been applied to this problem which, like multi-agent systems, is highly distributed in its nature due to the structure of the electrical network.

Nagata *et al.* [113] introduced the multi-agent approach to this application area, albeit using a non FIPA-compliant approach. This concept is extended by Solanki *et al.* [114], who demonstrate a FIPA-compliant multi-agent approach to distribution system restoration is possible for a number of scenarios, including full restoration and partial restoration with load shedding.

In [115], Baxevanos and Labridis also apply a similar multi-agent approach for a distribution network restoration scheme. Simulation results demonstrated that, for the authors' target application, an agent-based approach can optimise the restoration process to isolate the minimum segment of the network containing the fault.

In both of these examples, application-specific ontologies are defined. It could be advantageous in the future to reuse the upper ontology discussed in Section 3.4.2 for both of these applications, which would theoretically allow their systems to interoperate, allowing a number of different restoration schemes to be tested in parallel.

3.5.6 Power system protection and control

Baxevanos and Labridis' approach discussed above is reused in [116] for distribution network control and protection management, where the authors simulate the use of a multi-agent system for fault detection in underground lines. The simulation environment, using JADE, was distributed across a number of network-attached PCs to represent multiple substations, with additional components connected by serial link used to simulate Intelligent Electronic Devices (IED), demonstrating that the multi-agent approach is naturally suited to the distributed nature of the power network.

AuRA-NMS, the Autonomous Regional Active Network Management System, is a technology demonstrator integrating voltage control, power flow management and automatic restoration for next-generation "smart-networks" [117]. This project, based in the UK, aims to apply active network management so that new generators may be added to the grid at distribution level, while mitigating or deferring the need for additional network reinforcement. A multi-agent approach was chosen with flexibility and extensibility in mind, so that the system could adapt to future changes to conditions of the network.

Using a multi-agent approach is well suited to this type of problem as it is inherently distributable, and well suited towards simplifying the integration of multiple parallel techniques. Within AuRA-NMS, an 'arbitration agent' was proposed to integrate a number of different control algorithms, determining the best course of action for network reconfiguration through reflection. This approach to integration of multiple techniques was previously applied to partial discharge monitoring in the COMMAS system [109].

3.5.7 Power plant control

In [118], the authors present a conceptual design for a multi-agent fault diagnosis system to improve the performance of power plant control using a distributed diagnostic agents based upon artificial neural network classifiers. The fault diagnosis system is shown to detect the presence of faults in real-time under simulation, using 12 diagnostic agents to represent the 12 subsystems within a 600MW oil-fired thermal power unit. This concept is expanded further in [119], where particle swarm optimisation is employed for a multi-objective optimisation problem aimed at identifying optimal plant operating conditions within the same generation unit. In both cases the authors' demonstrate the use of intelligent diagnostic techniques, however the multi-agent systems presented are purely conceptual without any actual implementation. This work represents the furthest advances in the use of multi-agent systems for power plant control, and despite no concrete implementations being produced, the applicability of the multi-agent methodology to the issues of power plant control and fault diagnostics is clearly shown.

3.5.8 COMMAS v2

In [62], an online agent-based condition monitoring system for power transformers is presented which integrates sensor data from a field trial of around 50 sensors attached to two National Grid transmission transformers as part of the SUPERGEN "Asset management and performance of energy systems" (AMPerES) project. This system integrated a number of conventional and novel sensor types, including temperature and vibration sensors mounted on external cooling components, and internal sensors measuring oil temperature as well as dissolved gas, water and hydrogen levels. This system, henceforth referred to as 'COMMAS v2', also incorporates the original COMMAS diagnostic agents discussed earlier in this chapter, representing the second iteration of the COndition Monitoring Multi-Agent System.

The system uses an ontology based upon CIM, as recommended in [71] to leverage the existing CIM domain model and simplify data integration with other CIM-compliant system in the future. In addition, 5 high-level agent roles are defined, under which the functionality of condition monitoring application agents can be effectively modelled. These roles are:

Data provider agents: which provide a standardised agent interface to external data sources such as databases, web services and sensors;

Service provider agents: providing data conversion, analysis and diagnostic functions;

Data director agents: which provide behavioural control within the multi-agent system, linking the other agent roles together;

Archive agents: which are capable of persisting agent data in a centralised repository, and providing that data to agents when queried; and,

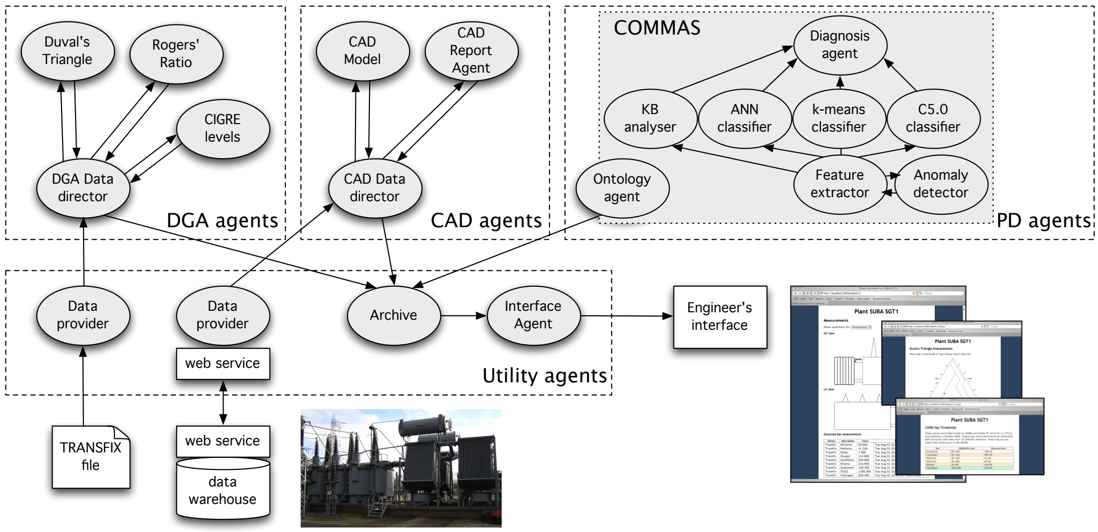


Figure 3.8: The COMMAS v2 online condition monitoring architecture, with agents split by high-level function [62]

Interface agents: which provide the opposite function to data provider agents, converting data in agent format to an external format, for instance, to communicate with an engineer's user interface.

An overview of the agents deployed within this system is shown in Figure 3.8. In this diagram, the agents are organised by high-level function, integrating:

Dissolved gas analysis (DGA) agents: which provide automated interpretation of fault gas levels;

Conditional anomaly detection (CAD) [120] agents: which provide contextual interpretation of anomalous measurements based upon environmental factors. In this system, statistical models of the transformer operation and the environment are used to identify anomalous readings which may be indicative of, for instance, a transient switching condition rather than a fault condition [62];

COMMAS agents: incorporating the original COMMAS partial discharge diagnostic agents agents described in Section 3.5.1; and,

Utility agents: linking external data sources, a system data archive, and the engineer's interface.

The scale and scope of this system's diagnostic capabilities demonstrate the power of multi-agent systems for condition monitoring applications, however some deficiencies still exist. This system does not currently connect directly to sensors, but instead executes remotely off-site receiving data through a web-service interface. Referring back to the 4 components of a condition monitoring system discussed in Section 2.2.1, this system demonstrates advanced fault detection and diagnostic capabilities, however the dislocation of diagnostics from sensors and data capture requires that raw data must be transferred from the sensor to the diagnostic system running on a corporate network, which could potentially introduce scalability issues in the future as the number of monitoring sensors increases.

In Chapters 6 and 7, this issue is addressed directly through the creation and application of a new condition monitoring architecture. This system complements the COMMAS v2 system, allowing diagnostic capabilities to run at the sensor level whereas they were previously were limited to running on a remote PC.

3.6 Summary

In this chapter we have seen an overview of multi-agent systems, including their structure, function and application to the power domain. The strengths of multi-agent systems over conventional distributed software methodologies lie in the autonomous nature of software agents, their ability to adapt to changes in their environment through external factors or reconfiguration of the agent society.

Multi-agent systems have been applied to a wide range of problems in the power domain, providing high-level abstractions required to build flexible, intelligent diagnostic and control systems. Mobile agents have been applied to certain problems, however their supposed benefits over static agent systems have never been successfully proven. The only area where agent mobility does provide clear benefits is to provide dynamic code updating, where new agents can be deployed into a remote agent container at runtime to amend or add new functionality. Applying code updates within an agent context allows agents to be updated without adversely affecting other agents operating on remote nodes.

Multi-agent systems have been shown to benefit power domain applications, however they do not address the issue of physically connecting sensors and sensor data to diagnostic capabilities, and ultimately monitoring engineers. Installations of agent-based diagnostic systems have, to-date, been connected to sensor systems either directly, or via bespoke communication channels in limited deployments. Utilities face a challenge in delivering new condition monitoring data to monitoring engineers, requiring a communications architecture which can operate under the constraints of a substation environment. In the next chapter, we will see that this can be realised through wireless sensor network technology.

Chapter 4

Wireless sensor networks

"Engineering is a great profession. There is the satisfaction of watching a figment of the imagination emerge through the aid of science to a plan on paper. Then it moves to realisation in stone or metal or energy. Then it brings homes to men or women. Then it elevates the standard of living and adds to the comforts of life. This is the engineer's high privilege."

- Herbert Hoover, 1874-1964

4.1 Overview

In the previous two chapters, we have seen that management and analysis of large amounts of monitoring data has been tackled through the application of multi-agent systems, although such installations have been limited to one or two pieces of plant within a single site. For condition monitoring to become more widespread within substations, a suitable communications and processing infrastructure is needed that cannot only connect all of the sensors to the corporate network, but also ensure that as more sensors are added, the flow of information to engineers is properly managed as to not overload engineers or communication networks.

Wireless sensor network technology has the potential to fill that role within the substation, as it can combine remote sensing with data storage and processing capabilities to manage condition monitoring data at the source. This chapter aims to give a brief overview of wireless sensor network (WSN) technology; incorporating the fundamentals of sensor network technology, current research directions, and previous applications of the technology to the power domain.

4.2 Introduction

Recent developments in miniaturisation of digital electronics devices have fuelled the development of wireless sensor networks. These networks are made up of a number of discrete battery-operated nodes which can 'self organise' to form an ad-hoc network with redundant links, and have already seen deployments in a number of application domains including military, the environment, home automation, health and transport. Sensor nodes have 5 principal components [121]:

Microprocessor: to perform on-board processing of data and execute sensing applications;

Data storage: modern sensor nodes typically have in the order of megabytes of storage space, which can be extended through the addition of on-board flash memory if required;

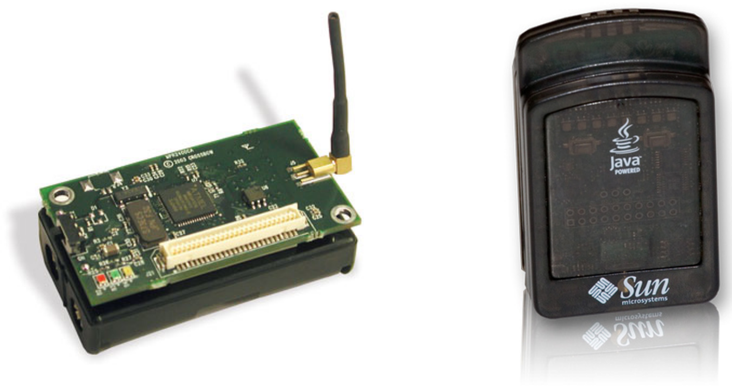
Sensing: through either on-board or externally connected sensors, connected to the processor via an input/output (I/O) interface through analog-to-digital converters;

Wireless transceiver: providing data communications between nodes, and inevitably connecting to a base station for data archival and dissemination; and,

Power: usually by means of a battery, although often including some form of renewable energy harvesting.

There are two streams of development for sensor networks; the first is ultra-low-power devices which are destined to become iteratively smaller over generations of the technology until they reach cubic millimetre size. These devices are designed to be integrated with the environment, to be 'painted on' to surfaces to measure ambient temperature, light, and movement [122]. The second stream of sensor network technology includes devices which are in the order of a cubic inch in diameter which, over time, will maintain their size but gain increased processing and memory capabilities in line with Moore's Law, increasing in capability year-on-year. For condition monitoring applications, the larger, more powerful devices are most attractive as they will allow more complicated local processing routines to be deployed and allow interfacing with custom sensors.

Two examples of common commercially available sensor nodes can be seen in Figure 4.1.



(a) Crossbow MicaZ 'Mote' (b) Sun Microsystems 'SunSPOT'

Figure 4.1: Examples of commercially available sensor nodes

4.3 Sensor network technology fundamentals

4.3.1 Communications

One of the key components of a wireless sensor node is its radio. Each node must have an RF transceiver with which to transmit sensor data, and receive updated configuration information or domain knowledge.

A number of communications standards have been published for sensor networking, with the most common being IEEE 802.15.4 [123] which defines a transport level protocol for low-power, low-bandwidth applications aimed at extended device battery life. The Zigbee standard [124] builds upon this to provide other features such as application security and mesh networking. The terms 'Zigbee' and '802.15.4' are often used interchangeably, however 802.15.4 is the network-layer communications protocol, and Zigbee is a standard published by the Zigbee consortium for application-level compatibility.

Wireless transmission of data within a sensor network is by far the most costly of all actions performed by a sensor node [121], therefore it is necessary to perform local processing of data where possible. In [125], the authors find that transmitting 1KB of data 100m takes the equivalent energy as performing 30 million processor instructions. Therefore, it follows that in monitoring applications, it is more efficient to calculate statistical features or fault diagnoses locally rather than sending raw data to a central point for processing. For condition monitoring applications, this means that diagnostic capabilities should ideally be embedded within the sensor network; not only reducing the energy consumption of sensing applications, but also reducing the communications burden on the underlying monitoring network while reducing the time taken to achieve a fault diagnosis.

4.3.2 Synchronisation

The ability to synchronise geographically dispersed nodes is critical within all distributed systems. Each node within a network must have an accurate time reference so that sensor data may be labelled appropriately. All digital hardware clocks experience frequency drift, where the internal time of the sensor node deviates from the actual time. This can be caused by a number of factors, including voltage fluctuations, age and temperature. In [126], the authors provide a survey on time synchronisation methods. There is no single method that is recommended, however a domain-specific approach is recommended depending on the application.

For any particular application, it is necessary to define the maximum sensor data timing error, and then synchronise nodes at the required interval. To give an idea of how much drift may be observed, the FC-255 crystal unit used within the SunSPOT sensor node [127] has a frequency tolerance of 2×10^{-5}, which translates to a maximum error of ±1.73 seconds over a 24 hour period. If sensor readings within a substation sensor network are to be synchronised—for instance, to the nearest second—then synchronisation of nodes in this case must be synchronised at least every 12 hours. This is a specific scenario, but illustrates the generic considerations that must be made when implementing a sensor network application.

In terms of actual implementations, it is possible to take a domain-specific approach. One example targeted at the power engineering domain is described in [128]. In this application, synchronisation is achieved by combining two methods; the first is a standard method where nodes sleep for a predetermined period, to wake at a preset time and carry out a duty. The second method uses an RF detector tuned to a specific frequency, which is triggered by a reference broadcast emitted by the base station. Sensor network synchronisation for substation applications is outwith the scope of this thesis, however a theoretical generic approach to power system sensor network synchronisation using a domain-specific solution is introduced in Section 8.2.3.

4.3.3 Sampling and processing

Current PC architectures benefit from clock speeds in the gigahertz range and memory capacities in the gigabyte range. Such ample resources provide a comfortable environment for software development, where memory usage and clock speeds limit only the most intensive of applications. However, the current generation of sensor network nodes run at megahertz speeds with memory limited to less than a few megabytes, therefore software development for such platforms must be carried out with those resource constraints in mind. Moore's law has proven that the computing power of sensor nodes will only increase over time, although this does little to help current endeavours in sensor network application development.

For simple data collection applications such as temperature and vibration monitoring, the hardware constraints of sensor nodes do not pose an issue, although for more complex applications it is necessary to ensure that diagnostic algorithms can operate within the processor, memory and energy constraints of the device.

4.3.4 Sensor lifespan and operation

The energy capacity of a sensor node is the key factor in determining its sensing, sampling, processing and communications capabilities. For a sensor node to operate across an extended timespan, it must spend most of its life in a deep sleep state consuming as little power as possible. In this state the processor, sensors and radio are deactivated, with only the internal master clock and working memory kept active. This places a restriction on sensing applications; sensor network application features must be balanced against the amount of available energy. If energy is abundant, then high-power sensing methods, continuous sampling and diagnosis may be feasible. However, if sensor energy must be conserved, then sensing must be periodic, and diagnosis and communication may need to be deferred.

Sensing typically falls under two categories: periodic and episodic sampling. In each of these scenarios, the available energy, expected operational lifespan and energy required to sample, process and transmit data will determine the system active state, or duty cycle; shown in Figure 4.2. Here, we can see that the sensor node spends the majority of its lifetime asleep, only waking to sample, process and communicate.

For periodic sampling where the sample is scheduled, such as a temperature sensing application, the wake period, τ, can be very low. Most sensor network applications fall under this regime. In this case, the sensor will wake, sample, store, and optionally process and transmit a sensor reading before going back to sleep. For episodic sampling, where the time of the event under observation cannot be determined such as a partial discharge event, the sen-

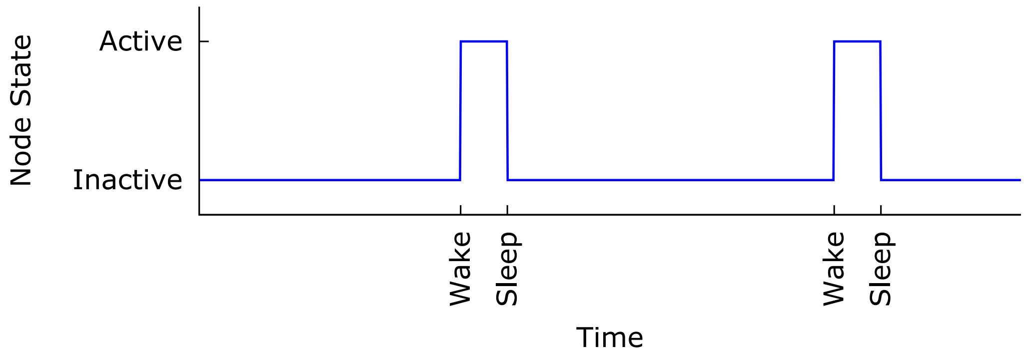


Figure 4.2: Active state of a sensor node

sor must either wake and wait for an event to occur or must be woken and triggered by external stimulus in anticipation of an expected event.

According to [129], with any wireless sensing application, it is necessary to analyse the energy profile of all components—both hardware and software—so that optimisations can be made. For substation applications, the lifetime of a sensor node must be several years as the assets they are observing have lifetimes in the order of decades. No utility would feasibly wireless sensors if it is necessary to replace them or their batteries yearly. Therefore, sensor nodes must operate continuously for several years, requiring sensing hardware and applications to be designed accordingly. Figure 4.3 illustrates issues relating to energy usage for a wireless sensing application, which should be considered when building applications.

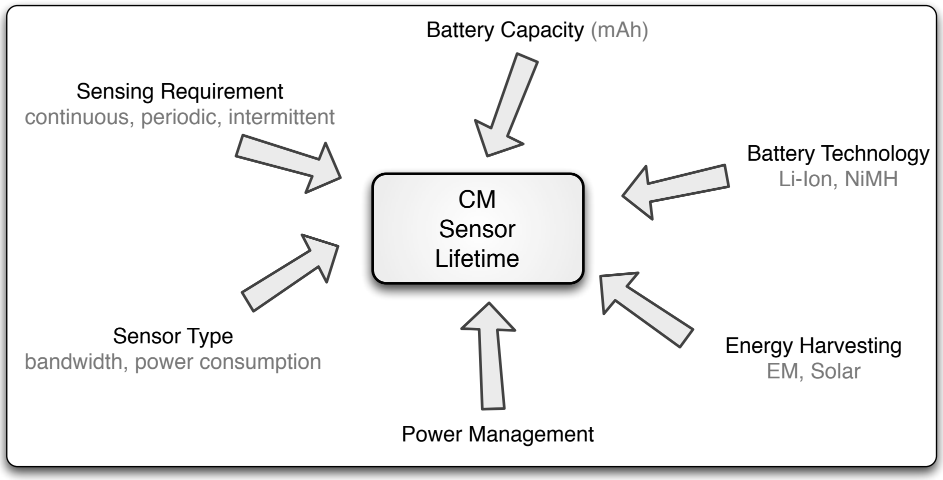


Figure 4.3: Sensor network lifetime issues

To loosen the constraints of battery operated sensing application and to allow more complex sensors and diagnostic techniques to be used, alternative sources of energy must ideally be considered to power sensor nodes.

4.3.5 Energy harvesting

With large-scale sensor network deployment, it is not feasible for batteries to be replaced regularly, so some other method of powering devices needs to be used. The limited battery capacity of sensor nodes requires that for extended lifetime and operational ability, it is necessary to harvest latent energy from the surrounding environment. As well as conventional renewable sources such as wind and solar, the environment surrounding electrical power systems provides a unique opportunity for wireless sensing applications as the constant oscillating electromagnetic field generated by energised components can be used as a source of energy. With these sources in mind, the practical methods of energy harvesting for wireless sensor nodes within a substation are:

Capacitively coupled harvesting: where a device placed within the electric field of a busbar consisting of a harvesting plate and switching circuit transfers energy from the harvesting plate to an energy buffer such as a capacitor. This can then be used to pulse-charge a battery, or alternatively power a sensor node directly [130];

Inductively coupled harvesting: where a toroidal core is placed around a conductor, whose magnetic field induces a current in a coil. This has previously been implemented as a commercial device for measuring voltage, current, phase angle and temperature of overhead lines [131]. For substation applications, a prototype device has been created that can potentially harvest 14W from a busbar [7]. Theoretically, this is enough energy to power a single board computer (SBC)[1] rather than a sensor node, although connecting this type of harvesting device to a busbar would require an outage, and sensing applications would be limited to monitoring the conductor, ambient measurements, or other components at the same potential. An alternative method is to place a coil away from the conductor, while still in the magnetic field. This will significantly reduce the amount of energy that can be extracted from the field, although would allow a much wider range of sensors to be used;

Vibration: Within power transformers, the electromagnetic field exerts mechanical stresses which cause vibrations. According to [7], by mounting an off-the-shelf vibrational energy scavenging device to the dampener at the base the unit, it is theoretically possible to extract a constant 4.5mW that may be used to power sensors; and,

Photovoltaic (PV): significant research has gone into PV harvesting for sensor network applications, although the suitability of this approach depends on the local climate. Solar power may be applicable in certain locations, although for general substation sensing applications a domain-specific approach such as those listed above would be more reliable.

[1] For instance, the Technologic TS-7800: a single board computer which uses 4W under normal operating conditions. See: `http://www.embeddedarm.com/products/arm-sbc.php#ts-7800-series`.

Energy harvesting for wireless sensor network applications is a relatively new area for research, although promises to extend the lifespan and expand the capabilities of wireless sensing applications in the future. Each of the techniques described above has certain pros and cons; for any sensing application, the choice of energy harvesting technique must be made based on environmental factors on a case-by-case basis. The nature of substation condition monitoring presents specific opportunities through EM harvesting; in the future, it is likely that once the technology matures both inductively and capacitively coupled techniques will make self-powered substation sensors a reality.

4.4 Sensor network platforms

Historically, applications running on embedded microcontrollers were typically written in the C programming language, requiring in-depth knowledge of the underlying hardware platform. In recent years, a number of platforms have been developed which abstract hardware functions, exposing them to the application developer through a standard high-level API. Two such platforms which have gained popularity for research applications are TinyOS and SunSPOTs, whose relative merits are described below.

4.4.1 TinyOS

TinyOS is a modular, threaded sensor network operating system which has become the de-facto standard for sensor network research applications due to its maturity and open architecture [132]. TinyOS nodes are colloquially described as 'motes': an example of one such mote can be seen in Figure 4.1(a).

The programming language developed for TinyOS, nesC [133], is a C-variant which provides an event-driven model with all hardware timers and interrupts delivered as asynchronous events. This 'flexible concurrency' model allows multiple application threads to run in parallel. Messaging within TinyOS is carried out using Active Messaging (AM) [134], which is an event-driven distributed software messaging system. This is discussed further in Section 4.5.3.

As both execution and messaging within TinyOS is event-driven, it is highly suited to low-power, periodic, long-lifetime applications. As soon as execution of a particular task has finished, the sensor node enters a power-saving 'sleep mode', only to be awakened when a scheduled task is to be executed. However, there are certain constraints faced when developing applications in nesC. TinyOS does not support dynamic memory allocation, where applications can vary their memory usage at runtime. This allows significant compiler optimisations to be made which are beneficial on resource constrained devices, however it does place fundamental limits on applications. For instance, in an agent-based system, an agent's knowledge can typically be expanded dynamically to accommodate new information. With static memory allocation, an agent's memory size would be fixed, so to accrue any new information an agent must first have to 'unlearn' any existing knowledge.

For simple applications such as temperature monitoring, this type of system may be sufficient. However, to be able to apply intelligent diagnostic techniques for condition monitoring applications, dynamic memory allocation is required to ensure flexibility. Therefore, a system such as TinyOS could feature within a substation monitoring application to perform data acquisition and processing, but is not suitable for higher-level diagnostic tasks.

4.4.2 SunSPOT

The Sun Small Programmable Object Technology, or SunSPOT system, is a wireless sensor platform based around the Java programming language. SunSPOT devices, seen in Figure 4.1(b), run the Squawk Virtual Machine (VM) [135], which is a Java 2 Mobile Edition (J2ME) runtime for embedded devices. This VM allows Java byte code to be compiled into an instruction format that can be executed directly by an embedded processor. It follows that building both sensing and diagnostic platforms in a single language has its benefits and, depending on the application, Java code that has been developed for server-based applications may be recompiled and deployed directly onto a SunSPOT.

As a diagnostics platform, the use of Java has its benefits. It has all the features of a modern object-oriented language, plus interoperability and code reuse are both critical issues to consider when writing software, so it is clear that using a single development language in both server-based and sensor-based applications is advantageous. We have seen in Section 3.3.1 that JADE is the most commonly used multi-agent system platform, and has been used for a number of condition monitoring systems. As it is written in Java, it follows that code reuse between JADE and a SunSPOT-based sensor network would be beneficial for engineers building a diagnostic system using both platforms.

One drawback of the SunSPOT system is that unlike the TinyOS system, there is only a rudimentary messaging system; in the SunSPOT API it is up to the application developer to include this, and to-date, no standard messaging system has been developed.

Comparing the two sensor network systems described here, TinyOS represents the lower end of WSN systems, offering reactive, event-based processing suitable for simple sensing applications. The SunSPOT platform represents the next generation of sensor nodes, utilising a high-level language targeted at building rich applications that can integrate with existing enterprise systems. Therefore, for intelligent diagnostic systems, this type of platform is better suited.

4.5 Middleware

So far in this chapter, we have seen the physical components of a sensor node, and examples of software abstractions that allow applications to be built upon them. In addition to these, middleware platforms provide software abstractions on which to build distributed applications, exposing low-level hardware components through a standard software API. Within a wireless monitoring application, middleware provides data and communication management while hiding the complexity of the underlying sensor network platform. In [136], 4 key components of a sensor network middleware architecture are defined:

Programming abstractions: exposing low-level functions through standard software interfaces;

Runtime support: such as task scheduling and thread support;

Quality of service: improving message delivery performance (This is an advanced feature, not present in most middleware systems).

System services: to perform routine tasks, and ease deployment of applications; and,

Within the 'system services' component, 5 key services are defined:

Code management: to allow remote updating of software code transparently and unobtrusively while the system is in operation;

Data management: to ease data storage within the distributed system, and simplify communication between parties;

Resource discovery: to allow distributed resources to be able to find local and remote services, and to detect and integrate additional sensor nodes and sensors;

Resource management: responsible for managing sensor node power and communications links; and,

Integration: providing integration to other networked systems, for instance; the Internet and FIPA multi-agent systems.

There have been a number of different approaches to sensor network middleware platforms, including a database-centric approach in tinyDB [137], and a proto-agent approach through the virtual machine environment of Maté [138]. The most prevalent approach to sensor network middleware, however, is the agent-based approach.

4.5.1 Sensor network multi-agent systems

Out of the many approaches to sensor network middleware, the one that endures above the rest is multi-agent systems. There is a natural analogy between multi-agent systems and wireless sensor networks; both are distributed in nature and consist of autonomous process centres which interact through messaging, with the key difference being that sensor nodes are physically discrete, and agents are logically discrete.

A number of agent systems have previously been developed for sensor networks. TinyLIME [139] is an implementation of the Linda [140] blackboard communications system for the TinyOS platform, where each node is modelled as an agent. Each node employs a shared memory model where 'tuples', or associative arrays, are inserted and removed from a central space accessible to all agents. Agents subscribe to particular tuples, through 'reactions', which are analogous to event listeners. Using a tuple space allows these platforms to effectively decouple their agents from the underlying sensors.

Fok *et al.* introduce 'Agilla' [141], a mobile-agent platform for TinyOS based upon Maté [138], using a virtual machine approach to allow simple reactive mobile agents to execute and migrate between nodes. Agilla also employs tuple spaces; within this system, on arrival to a sensor node, a mobile agent can discover which compatible tuples are available by inserting a 'reaction' into the tuple space, effectively subscribing to new and archived local sensor data. Both local and remote tuple-space operations are supported, where remote tuple space operations replace the traditional messaging layer found in other multi-agent systems. This type of data management system is well suited to sensor network applications, as it has much lower overheads than service-discovery and messaging system.

There have been several other applications of mobile agent systems to sensor networks, including the use of mobile agents to carry out data integration [142], where agents travel between sensor nodes aggregating data to obtain increasingly accurate sensory data, and event tracking and data collection applications [143]. Georgoulas *et al.* present 'In-Motes' [144], an agent platform heavily based upon Agilla. This platform implements a federated system [145], sacrificing agent autonomy to a 'facilitator agent' which is in charge of a number of agents, aiming to reduce network communication through individual agent federations using a subset of the global agent communication language.

Within these systems, agents must be written in byte code to allow mobility at the expense of functionality and flexibility. An example of an agent written for the Agilla platform can be seen in in Figure 4.4. This agent reads the local temperature every 10 minutes, asserting a 'fire' condition if the temperature is in excess of 200 degrees[2]. While this code may represent an agent, it certainly does not represent a flexible, autonomous intelligent agent, and although developing 'agents' in this format generates optimised sensor network programs, the development effort to create a complex intelligent agent is prohibitively large.

A number of high-level agent platforms have been developed for mobile ad-hoc phone networks, including AgentLight [146] and JADE -LEAP [147]; a J2ME version of the JADE platform. In these cases, it has been possible to translate the FIPA agent model to mobile devices, however this is because in the mobile ad-hoc environment energy consumption is not paramount as devices are typically charged every few days. Compared to sensor nodes, mobile phone and PDA devices have a relatively high energy consumption due to their more powerful radio (GPRS / WiFi), and a graphical interface. The relaxed energy consumption constraints allow the FIPA model to be directly translated to mobile devices, however this model is not suitable for sensor networks. Fundamentally, this problem is due to the complexity of FIPA communications protocols, discussed in the next section.

[2]It is not stated by the authors whether this is 200 degrees Celsius, Kelvin or Fahrenheit

```
BEGIN     pushc TEMPERATURE
          sense               // measure the temperature
          pushcl 200          // push 200 onto stack
          clt                 // set condition=1 if temperature > 200
          rjumpc FIRE         // jump to FIRE if condition=1
          pushcl 4800
          sleep               // sleep for 10 minutes
          rjump BEGIN
FIRE      pushn fir           // push string ''fir''
          loc                 // push current location
          pushc 2             // stack has fire alert tuple
          pushloc 0 0
          rout                // rout fire alert tuple on node at (0,0)
          halt
```

Figure 4.4: An example of a fire detection agent written for the Agilla platform [141]

4.5.2 The problem with applying FIPA protocols to sensor networks

In Section 3.4.1, the expressiveness of the FIPA communications protocols has been shown to allow multi-agent systems to communicate with both message *content* and message *intent*. The benefits of these protocols, however, are not without an associated cost that is critical for sensor network applications, in the form of significant protocol overhead to the original message payload. In fact, FIPA protocols are unsuitable for sensor network applications: this can be demonstrated by analysing the string encoded ACL message shown in the previous chapter in Figure 3.3. Table 4.1 shows a breakdown of the message contents by the number of bytes required for each message component.

Table 4.1: Breakdown of a string-encoded FIPA request message

	Message	**Payload**	**Data**
Size (Bytes)	412	202	12
% of total	100	49.0	1.45

Here, we see that the content constitutes less than half of the overall message, and of that message content, only 6 bytes, or 1.45% constitutes data. In this case, the FIPA protocol overheads double the message size, and increase the message size by two orders of magnitude over the basic constituent data content. Within broadband networks, these overheads typically have an insignificant impact on the operation of the overall system, as both bandwidth and energy are plentiful. In these cases, the use of FIPA protocols has little impact as there is minimal cost of sending such a message. However, within a wireless sensor network this is generally not the case as devices are energy-constrained therefore communications must be highly optimised to minimise energy usage.

Considering the memory footprint of such a message, storing it in the memory of a server with hundreds of megabytes of RAM has little impact, but storing the 412 byte message on a device such as the MicaZ, with 4KB of RAM, will have a much larger impact. This message alone would consume around 10% of the total working memory; a significant proportion of memory for a simple message. Using string-encoded FIPA messages in this case would quickly fill a device's memory, even before taking into account the resources required to generate, parse and process such messages. To tackle this issue for resource-constrained devices, FIPA has defined bit-efficient message encodings for resource-constrained devices, previously discussed in Section 3.4.1).

A study of the efficiency of the FIPA bit-efficient codec over other encoding schemes was carried out by Helin *et al.* [89], which found that compared to String encoding, the bit-efficient codec was found to provide a 50% reduction in message size, which increased to 75% when using a code table. The use of bit-efficient codecs also decrease message parsing times, as the hexadecimal symbols used can be matched faster than their string counterparts, however this does not necessarily make it suitable for use within sensor networks. Even if the 412 byte example message above was reduced by 75%, 103 bytes would still be required to transmit only 6 bytes of actual data.

The practicalities of using code tables must also be examined. Within the FIPA specification, the bit-efficient code table is defined as both *unidirectional* and *per-agent pair* [148], so the same codes are not reused for sending and receiving, and cannot be shared between multiple recipients. This means that for a sending and receiving pair, it is necessary to store a unique sending and receiving code table. Assuming an ideal situation where the sending and receiving code tables are both full, at which point communications will be fully optimised, then there will be a duplicated sending and receiving code table for each agent pair at both the sender and receiver. This is illustrated in Figure 4.5, which shows sending and receiving code tables duplicated for each agent.

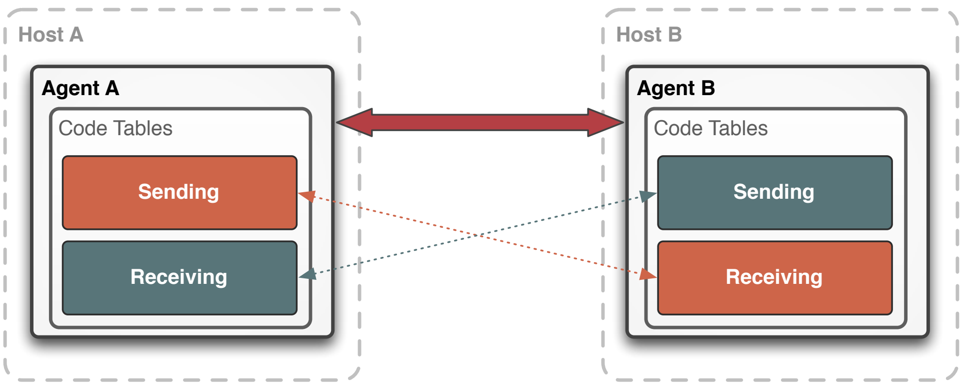


Figure 4.5: Code tables of a sending and receiving pair of agents using FIPA bit-efficient communications

Using the FIPA bit-efficient specification as a guide, it is possible to establish the maximum code table size under this condition. Here, we will see that the use of code tables and the bit-efficient codec becomes impractical for use within sensor network applications. Let:

- b_c equal the code table index length in bytes;

- b_w equal the average code word length in bytes;
- S_c equal the size of the code table; and,
- M_{ct} equal the total code table memory usage for an agent pair.

The FIPA bit-efficient specification defines the maximum value of b_c of 2 bytes, giving a maximum code table length, S_c, of 2^{16} or 65536. Assuming an average code word size of 13 bytes, such as in the example shown in Table 3.2, if 2^{16} 13-byte words were stored in both sending and receiving code tables for an agent pair, then M_{ct} for a sending and receiving agent pair would be:

$$\begin{aligned}
M_{ct} &= 2 \times (sending\ code\ table\ size + receiving\ code\ table\ size) \\
&= 2 \times (S_c \times ((b_c + b_w) + (b_c + b_w)) \\
&= 2 \times (2^{16} \times 2(b_c + b_w)) \\
&= 2 \times (65536 \times 30) \\
&= 3932160\ bytes \\
&= 3.75MB
\end{aligned}$$

In this case, for each agent pair the code size memory usage would be 3.75MB. Within an agent system with n agents employing bit-efficient messaging, assuming each of the agents communicated with each other and maintained full code tables, the code table memory usage for the entire agent system would be:

$$M_{ctSystem} = n! \times M_{ct}$$

The global code table memory usage for a multi-agent system with a varying number of agents can be seen in Figure 4.6. For a 5 agent system operating within this scenario, the code table memory footprint would be 450 megabytes, whereas a 10 agent system would require approximately *13 terabytes*. It is worth bearing in mind that this is a worst-case scenario, and this value could be drastically reduced via code table optimisation, although we can see from this example that even with a relatively small agent system, the code table memory usage can become too large for sensor network operation where memory usage is constrained to megabytes per node.

This outcome shows that, while bit-efficient encoding can reduce the message size by up to 75%, the remaining overheads coupled with the large mem-

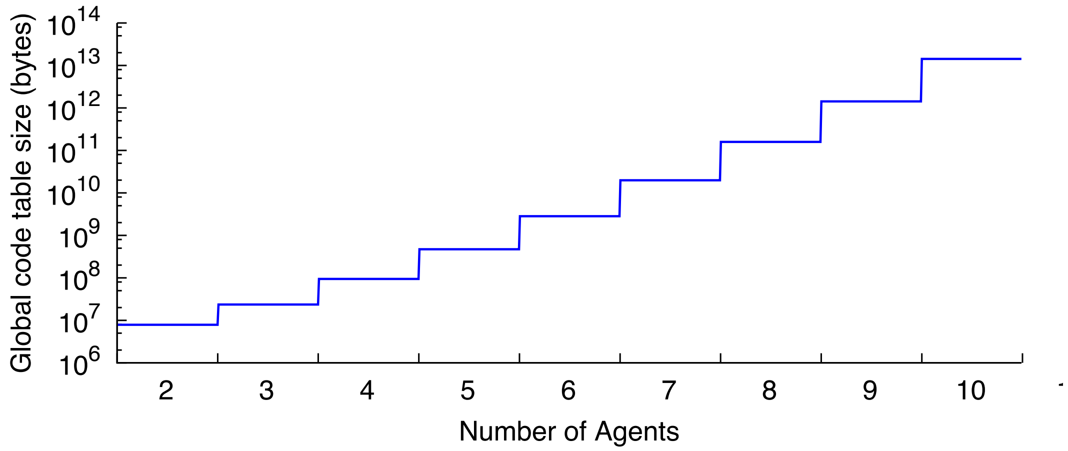


Figure 4.6: Global code table size, in bytes, for a multi-agent system using bit-efficient encoding scheme for FIPA-ACL messages

ory costs and lack of code table scalability makes FIPA messaging unsuitable for use within sensor networks.

4.5.3 Alternative application protocols

To date in the field of sensor network research, the vast majority of research on communication protocols has been focused on low-level communications, generally concerning the optimisation of network links between sensor nodes and in fact it is recognised that there is a lack of application level protocols in general [121].

The one protocol which stands out within this area is Active Messaging [134]. This is as much to do with its simplicity and suitability as well as the fact that it is part of the TinyOS platform.

In the active message model, on each node, event handlers are registered to receive a particular type of message. These event handlers consume the incoming messages which match a particular message type ID, integrating the payload data into the receiver node's execution and optionally sending a response. This type of communications is well suited to sensor network applications where the transmission of wireless data must be kept to a minimum.

The syntax used for a TinyOS Active Message, shown in Figure 4.7, only defines the message type and payload. In TinyOS, the active message protocol overhead is a single byte to denote the message type over and above the addressing information. As the message payload must be as compact as possible to minimise message size and therefore message transmission costs, the structure of the payload must be defined on a per-application basis.

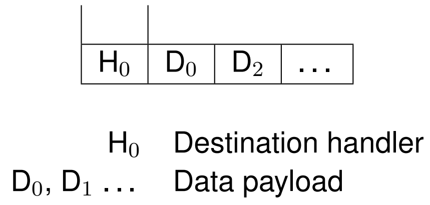


Figure 4.7: Simplified Active Message structure, with network routing fields removed

For heterogeneous data, such as a single value or array of values, message structure is straightforward as as long as the field size and order is known. Data is effectively serialised into a byte array by the sender, and deserialised by the receiver using the message type ID to establish the message structure.

For multi-faceted data, message structuring is a more complex problem, for instance when data is incomplete, or may contain a variably sized payload. A presentation protocol such as JSON [149] could be employed to give structure to complex messages without incurring a large overhead; this approach has recently been applied in [150] and [151]. To-date, this is still an open problem which must be addressed by the wider multi-agent system and sensor network research communities; standardisation in this area would lead to increased systems interoperability.

The use of sensor network communications protocols, however, presents a new problem when using FIPA-compliant agent systems; namely, how to handle translation between bit-efficient messages and FIPA-based protocols.

4.5.3 c) The need for a WSN-to-FIPA proxy

Multi-agent systems are well suited to integration problems, where a 'wrapper agent' can be written to provide a proxy to an external system. For instance, with a data source such as a database, the standard approach is to write an application-specific adaptor to wrap the data source so that it may be used within the multi-agent system. This same approach can also be applied to a sensor network—in fact, in TinyDB a similar approach has been taken to present a sensor network as a database—however, despite this providing a simplified abstraction for a sensor network, it removes the ability to deploy and manage intelligent diagnostic capabilities within the network, reducing the entire system to a logical data source. For an effective link between the two architectures to be realised, an agent-based approach should be taken to sensor network application design, where the sensor network itself hosts a

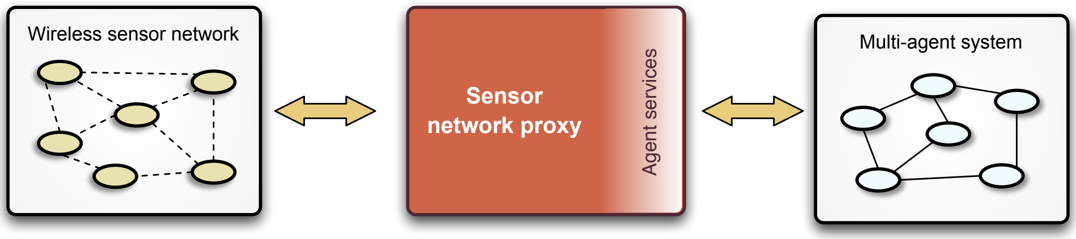


Figure 4.8: Sensor network proxy providing a standard interface between agents operating within the sensor network and FIPA multi-agent systems

multi-agent system.

This has obvious benefits for agent-based condition monitoring systems, as the same programming abstraction can be used from the sensor level up to the engineering interface. However, this introduces new challenges that must be addressed. For a FIPA-based multi-agent system to be able to interact with the agent-enabled multi-agent system, a simple adaptor agent such as those used for databases or file access is not suitable. As well as translating lightweight sensor network messages to a messaging format understood by FIPA-based agents, a sensor network gateway must also transparently expose services within the sensor network as standardised agent services. This is illustrated in Figure 4.8.

The proxy-based approach to sensor network integration has been applied previously. In [152], a FIPA-compliant JADE gateway agent provides an interface to an Agilla sensor network application which provides incoming messages from the sensor network to any agents that have subscribed to a particular service, allowing multiple agents to receive the incoming sensory data as soon as it becomes available. In [153], an 'edge server middleware' is proposed to act as a proxy between sensor networks and application servers, running application-specific code to process and optimise sensor network data before it is transmitted across a network. Both of these systems are application specific, where the 'wrapper' approach has been chosen to provide explicit rather than a general-purpose translation capabilities. A flexible, generic proxy server could provide the link between new sensor network and existing agent-based condition monitoring systems. This issue is tackled later in this thesis in Chapter 6.

4.5.4 Sensor network ontologies and integration

There have been a number of different approaches to to describe sensors, sensor networks and sensor data for sensor calibration, configuration and integration. Standards include IEEE 1451 [154], which is a 'smart transducer' interface standard, which allows the self-description of sensors through an electronic data sheet, and plug-and-play interoperability between sensor nodes and network-connected processors. This standard has not yet received widespread adoption due to its complexity, although IEEE 1451-compliant sensors have been integrated with existing middleware platforms in [155] and [156], with the aim of using IEEE 1451 hardware descriptors for zero-programming deployment.

SensorML is an open XML standard for sensor metadata, which allows sensor network hardware, data and processes to be described in a machine-readable document format. The standard, published by the Open Geospacial Consortium, is designed not just for wireless sensor networks but for any type of sensor system. SensorML allows the modelling of both sensor components and processes for measurement and post measurement data analysis [157].

There have been a number of other ontologies built for sensor networks, including OntoSensor [158], an ontology built using the Web Ontology Language (OWL) which extends SUMO, the Suggested Upper Merged Ontology. This system, based upon SensorML, allows sensor datasheets to be created in OWL format (discussed in Section 3), which can be used for calibration and data integration.

In [159], the authors propose an ontology based upon IEEE 1451 for sensor network data, with the aim of optimising data retrieval. Avancha *et al.* propose an ontology driven adaptive sensor network in [160], where an ontology is used to define sensor metadata which can be used to calibrate new and existing sensor data.

A key aim of the sensor network ontology research area is to integrate sensor network data with existing systems. Within power utilities, standards for integration have already been published for substation automation through IEC61850 [161], and inter-utility communication through CIM [94]. We have seen in Section 3.4.2 that an upper ontology for agent-based applications has been proposed to ease integration with existing utility systems, and if multi-agent systems are to be used to their full potential to enhance sensor networks with intelligent capabilities, it would be necessary for a power system ontology such as CIM to be integrated with a standard sensor network ontology such as IEEE 1451 or SensorML.

Ontology integration has been addressed for condition monitoring applications in [162], where it has been identified as a non-trivial task. It is outside the scope of this thesis, but is discussed further in Section 8.2.

4.6 Sensor networks in the power engineering domain

Prior to the advent of wireless sensor network technology, the use of radio links for power system sensors has been widely explored [163, 164], with the most well documented case being that of the 'Power Donut' [131] which is designed for transmission line monitoring integrating wireless line current, conductor temperature, ambient temperature monitoring with inductive energy harvesting in a single package. However, a proprietary radio protocol is used, and no on-board processing is carried out. All data is transmitted in its raw form back to a base station for analysis.

With the advent of wireless sensor network standards, there have been limited studies into the use of this technology in the power engineering domain. The practical aspects of using modern wireless communication standards in substations have been reported upon; impulsive noise from switching and partial discharge is known to pollute the radio spectrum used by standard wireless communication protocols [165]. However, a study on the performance of the Zigbee protocol within substations found negligible effects on communications from impulsive noise [166], suggesting that noise from partial discharge and switching operations would not have an adverse effect on a wireless condition monitoring system.

In terms of applications, in [167] a survey of networking options for substation automation included a commentary on the applicability of wireless communications to this area, finding that the lower quality of service (QoS) compared to wired communications make wireless communications infeasible for automation applications that include protection and switching. However, for substation condition monitoring applications where the change in state of plant health is typically over weeks, months or years, the drawbacks of wireless communications are minimal, vastly outweighed by their inherent benefits. Applications in this area include that of Leon *et al.* [167], where a wireless sensor network was designed for mechanical health monitoring of transmission lines, to allow operators to schedule preventative maintenance and aid in the analysis of post-fault conditions.

At the distribution level, wireless sensor networks have been applied to both operational monitoring, and plant condition monitoring. In [168], a wireless sensor concept for electrical distribution networks is presented, which uses phase current characteristics to estimate fault locations. In [169], this concept is explored further, including a novel domain-specific time synchronisation scheme for sensor nodes.

A wireless temperature sensor network for substation monitoring was deployed by Nasipuri *et al.* [170]. This installation measured transformer tank surface and circuit breaker temperature, utilising photovoltaic energy harvesting to extend the operational length of the sensor network system to 8 months before all nodes drained their batteries. In [171], a fully-customised wireless sensor network was designed for busbar joint temperature monitoring. Both of these systems give little information as to the performance of their systems, representing proof-of-concept systems rather than generic approaches to wireless substation condition monitoring.

Each of these applications uses in-situ sensors that take measurements of components directly adjacent or attached to the wireless sensor; for instance in [170], temperature sensors were attached to the transformer tank surface to report the external temperature of the transformer. This type of monitoring application is typical of wireless sensors, however this approach is far removed from diagnostic methods such as partial discharge detection which, discussed in Section 2.4.2, can inform monitoring engineers about the internal state of plant. For wireless sensor networks to fully benefit substation condition monitoring, new sensing methods are required which are suited for wireless applications. These should be able to inform engineers on the internal state of plant as with existing methods, albeit using low-power techniques.

4.7 Summary

This chapter has provided an overview of wireless sensor networks, describing the core components that make up the technology. Applications of the technology are gaining ground across industry, and although there have been limited deployments to date, deployments to electricity distribution systems will be no exception in the future. Wireless sensor networks have the potential to simplify the process of instrumenting existing plant items while circumventing restrictions on cabling within a high-voltage environment. Sensor nodes can also host diagnostic capabilities, processing raw sensor data at the source to manage the flow of information at the earliest possible stage.

There are a number of opportunities for energy harvesting within substations, and while this can potentially ensure long lifespans for wireless substation sensors, it is still necessary to design and build useful sensors that can enhance existing condition monitoring systems and provide monitoring engineers with new diagnostic information with which to make maintenance decisions. This thesis investigates the practicalities of implementing such sensors; in the following chapters a low-power approach to condition monitoring is described, using partial discharge detection as an example application. This method is then integrated with an existing condition monitoring system using a multi-agent based approach, to demonstrate that wireless sensors can feed into and enhance existing condition monitoring systems.

4.8 Chapter Bibliography

The following sources were used as bibliographic sources for this chapter, in addition to the inline references:

- D. E. Culler, D. Estrin, and M. Srivastava, "Guest editors introduction: Overview of sensor networks," *IEEE Computer, Vol. 37, No. 8*, Aug. 2004
- I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks, Volume 38, Issue 4*, pp. 393–422, 2002

Chapter 5

Low-power wireless condition monitoring sensors

"Results! Why, man, I have gotten a lot of results. I know several thousand things that won't work."

- **Thomas A. Edison, 1847-1931**

5.1 Chapter overview

The previous three chapters presented a background on substation condition monitoring, multi-agent systems, and wireless sensor networks. While wireless sensor networks are suitable as an architecture on which to build distributed condition monitoring applications, the use of this technology comes with several constraints. WSN applications must operate under strict energy restrictions, with power limited to what can be stored in on-board batteries or harvested from the surrounding environment.

To illustrate the opportunities for utilities to exploit WSNs, this chapter investigates existing monitoring techniques that may be suited to low-power operation. We will see that there are a number of shortcomings in these approaches, and therefore there is a need for low-power, non-invasive sensors that can go further than detecting ambient or surface measurements by giving an indication of the internal state of plant.

In this chapter, partial discharge monitoring is chosen as an application testbed upon which low-power monitoring can be investigated. Partial discharge detection was chosen for this particular study as PD activity is a widespread concern for utilities, and can give an indication of the insulation health of plant—an important factor in determining maintenance requirements. From this study, a novel partial discharge detection technique is presented to demonstrate that insulation health monitoring is feasible using a low-power approach.

The aim of this study was to establish whether partial discharges could be detected and classified using a battery-operated device which could be deployed into a substation as part of a wireless condition monitoring system. This chapter will demonstrate that partial discharges may be detected, analysed and classified through using a low-power approach which can be executed on a wireless sensor node.

5.2 Low-power options for substation sensing

Section 4.6 described a number of applications of low-power sensor networks to power engineering applications, each of which targeted simple measurands such as current and temperature. Wireless sensor networks are well suited to these types of applications as they function most effectively with relatively simple sensors which generate low volumes of data. However, as discussed in the previous chapter, there are limitations these sensing methods for substation condition monitoring applications as the nature of the sensor means that they can only sense the periphery of the asset—for example, the external temperature of a transformer tank. While this can potentially benefit substation monitoring, it stops short of providing information of greatest interest to monitoring engineers: the internal insulation integrity of plant. Therefore, for wireless sensing applications to be of greatest benefit, alternative low-power sensing techniques are required that are not limited to monitoring peripheral components.

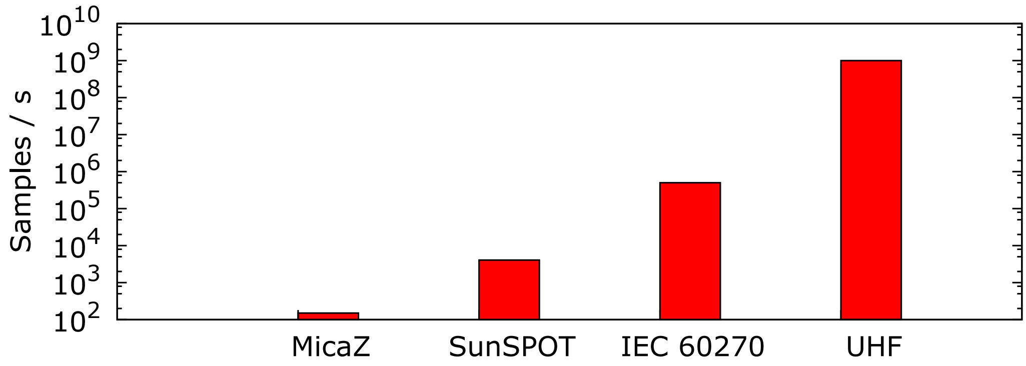
Based upon the types of monitoring techniques applied within substations today, this poses somewhat of an issue. Diagnostics have evolved with the increased availability of computing power, with wideband sampling and computationally complex diagnostic techniques becoming the norm within the research field in recent years. The computational costs required to process volumes of data generated by wideband sensors is far greater than can be supported by the current breed of sensor network hardware, therefore, if sensor network technology is to be applied in this area, a low-power approach must be taken.

5.3 A different approach to PD monitoring

Existing PD pulse monitoring techniques, such as the IEC and RF methods discussed in Chapter 2.4.2, typically use high-speed sampling to capture wideband signals. Based on the specifications of the current state-of-the-art in sensor network technology, it was clear that wideband signal capture is not yet feasible on today's low-power sensor node hardware. Figure 5.1 illustrates this by showing the sampling and memory capabilities of a number of off-the-shelf sensor nodes compared to the basic requirements for IEC 60270 and UHF PD monitoring schemes. To the left of the figure the sampling rates of the MicaZ and SunSPOT sensor nodes are shown, against, on the right, the minimum number of samples per second required for IEC 60270[1] and the implementation of the UHF method described in [29].

It is clear that the sampling rates offered by the current breed of sensor nodes are orders of magnitude less than is required for wideband partial discharge monitoring. Therefore, for utilities to exploit WSN technology in this area, a different approach to PD monitoring is required. As wideband digital sampling is impractical to embed within the substation, PD monitoring should

[1] From IEC 60270 Standard, Section 4.3 "Measuring systems for apparent charge" pp27-31



- **MicaZ:** 300Hz
- **SunSPOT:** 8.1kHz
- **IEC 60270:** 500KHz [30]
- **UHF method:** 1GHz [29]

Figure 5.1: A comparison of sensor node capabilities against PD monitoring standard requirements

be carried out using an alternative method. To address this, it is necessary to look at the fundamental process involved in detecting partial discharges.

5.3.1 Partial discharge detection

As discussed in Chapter 2, partial discharges are the result of localised ionisation in the insulation structure. This emits wideband electromagnetic radiation which can be captured using acoustic, optical or UHF detection, and used to diagnose the root cause of the PD emission. Figure 5.2 shows a typical partial discharge RF signal that has been captured using a broadband oscilloscope.

The rise-time of an RF partial discharge signal can range from a few nanoseconds to several microseconds depending on the insulation medium [21]. To accurately sample a signal, Nyquist's theorem states that the sampling frequency must be at least double the signal frequency. Therefore, for a partial discharge pulse with a 1 nanosecond rise time, this places the sampling rate requirement in the range of 2GHz. Modern oscilloscopes can comfortably sample at these rates, with recent UHF diagnostics using sampling rates of up to 5GSs^{-1} [173], however these sampling rates are currently infeasible for low-power devices. For low-power PD identification, a trade-off against function must be made; instead of the entire signal being captured, specific features of the PD wave-

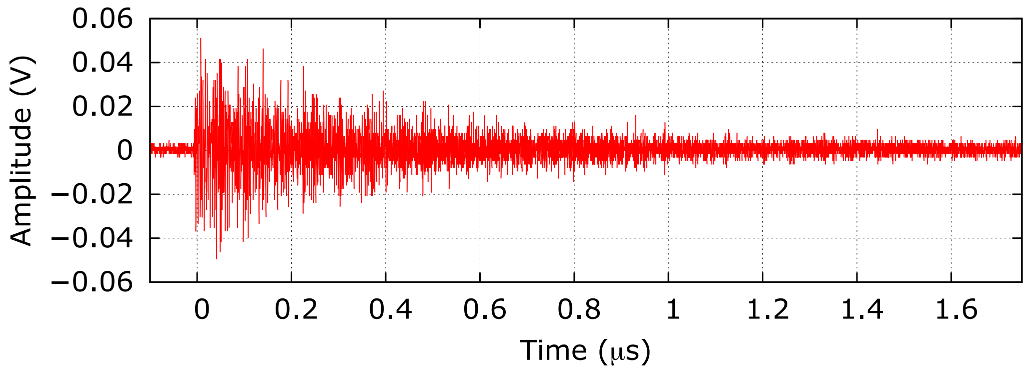


Figure 5.2: An RF partial discharge signal

form should be identified which contain defect-specific information that that can be extracted using a low-power method and may be used for diagnosis. By using an RF pulse detection circuit, we can convert the PD waveform into a signal representing the PD pulse envelope, shown in Figure 5.3.

Partial discharge envelopes have been shown to contain defect-specific information [51]. However, as the pulse width of a PD envelope is still in the region of less than 1μs, it requires a sampling rate of at least several MHz to capture an entire PD pulse envelope signal. As discussed previously, a low-power application can feasibly support sampling in the kilohertz range, therefore to be able to identify partial discharge using a low-bandwidth approach, only fundamental properties of the partial discharge pulse can be measured: for instance, its magnitude.

To achieve this, it is necessary to 'sample-and-hold' the PD pulse, using an analog circuit to follow the PD pulse magnitude, holding the maximum magnitude until it can be digitised. This method is simple to implement in hardware as the PD can be sampled when the magnitude exceeds a predefined thresh-

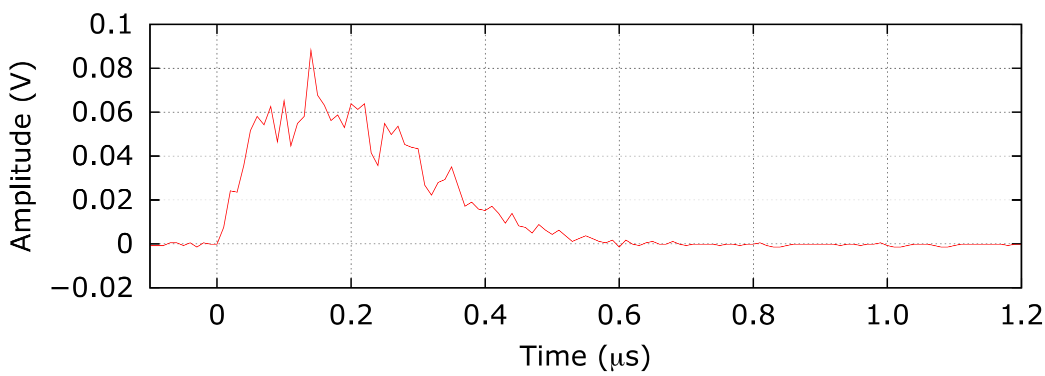


Figure 5.3: Partial discharge signal envelope

old. However, the magnitude alone cannot give an indication of the partial discharge type, location or severity, and simple RF detection is not resistant to interference, as any RF impulses would generate the same resultant value. For this simple technique to be effective at defect diagnosis, additional features must be obtained from the PD pulse.

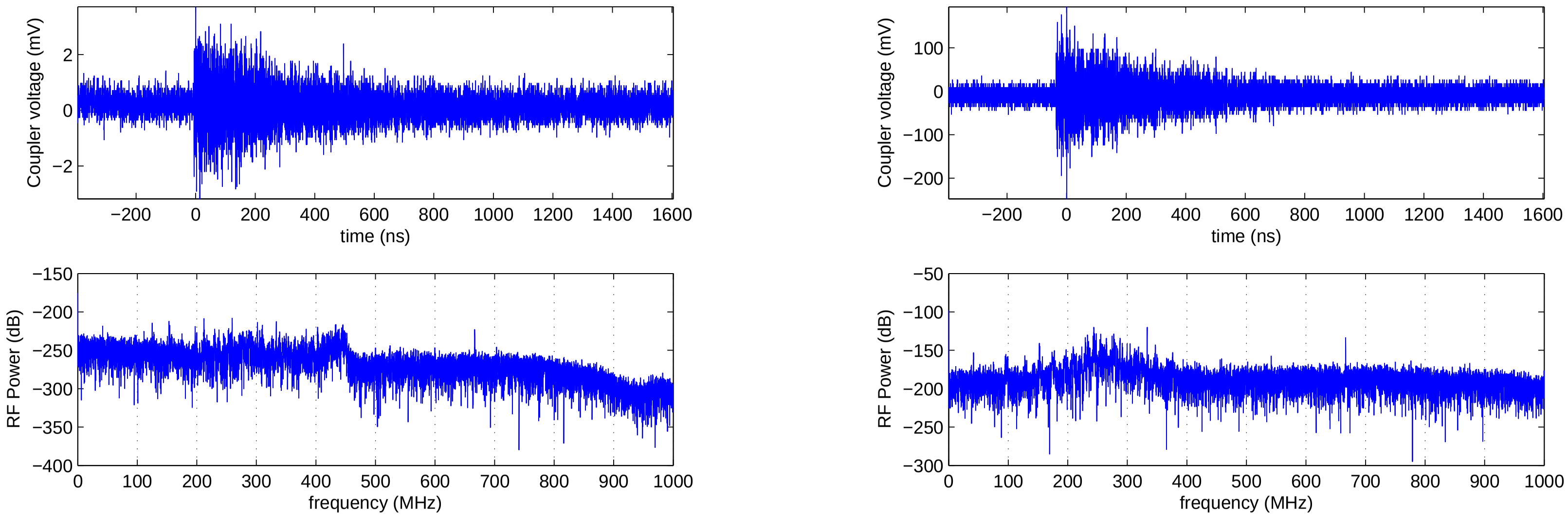
5.3.2 Frequency-based diagnostics

The RF signals emitted by different PD sources have previously been shown to contain varying proportions of energy across different frequency bands [174, 53]. Contin *et al.* [32] state that measuring the frequency content may be advantageous in the separation and classification of PD sources, however it is discarded using conventional peak detection equipment. Previous research into PD monitoring has highlighted the advantages of separating PD pulses into homogeneous classes, each relevant to a specific defect, which makes subsequent analysis and identification easier [175]. Based upon these assertions, by calculating the spectral energy components of a partial discharge emission, it is theoretically possible to classify defects based upon their frequency components.

Figure 5.4 shows two PD pulses and their corresponding frequency-domain representations generated using a fast-fourier transform (FFT). The protrusion defect shows a higher proportion of energy below 400MHz than above, whereas the rolling particle defect shows a peak in spectral energy at around 250MHz. The frequency domain representation of PD pulses do not appear to be too distinct in these figures, however the specific differences in energy spectra will be drawn out and made clear later in this chapter.

5.4 Requirements for a low-power PD detector

To establish whether partial discharge diagnostics could be effectively carried out using a low-power technique, a prototype PD detector was built implementing the frequency-based technique outlined above, so that test data may be gathered to validate the technique. Fundamentally, this detector must be capable of detecting partial discharges and identifying the proportion of RF energy in a number of frequency bands. To successfully implement the detector, it was first necessary to fully specify its functional requirements.



(a) Protrusion defect

(b) Rolling particle defect

Figure 5.4: Time-domain and frequency-domain plots of PD pulses for protrusion and rolling particle defects [176]

5.4.1 Functional requirements

The key functional requirements of the detector are that it can:

- sense and record the intensity of PD signals and their relative spectral magnitudes for a set of predefined frequency bands;
- differentiate between PD and background RF noise such as mobile phone signals and other impulsive events such as corona discharge; and,
- have an operational power consumption that makes prolonged battery operation feasible.

As the detector is ultimately to be directly attached to plant, it should be relatively small, although size constraints are not paramount as the plant under observation will be orders of magnitude larger than the monitoring device.

Another requirement is the ability for the detector to interface with RF sensors attached to plant enclosures. New GIS installations commonly include integrated RF sensors, and sensor mounting facilities are undergoing standardisation efforts for oil-filled transformers [66], where they may be mounted to inspection hatches and through oil valves [44]. Examples of such sensors can be seen in Figure 2.7 in Section 2.4.2 b).

5.4.2 Frequency bands

To implement a frequency-based diagnostic technique, the number of bands used for analysis must first be established. Operating from DC through the UHF band, from 0 to 3 GHz, the frequency spectrum must be split in such a way that the difference in frequency content between defects may be isolated. A larger number of bands and, therefore, a larger number of resultant features may yield a greater separation of results, but as the number of bands increases linearly, so does the complexity and energy consumption of a physical detector implementation. Therefore, to optimise energy consumption, a lower number of frequency bands is desirable.

To determine frequency bands for an initial prototype, a study of the literature highlighted existing research within this area. Meijer [53] previously used the relative energies of a low-frequency span of 100MHz to 350MHz compared to a high-frequency span of 750MHz to 1100MHz with which to calculate a classification feature. Building upon this research, these bands were used as

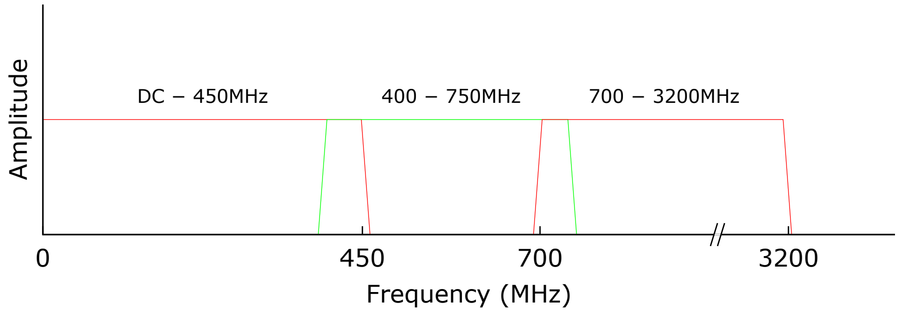


Figure 5.5: Detector channel frequency bands

a basis for the detector, with an additional centre band added to cover the entire spectrum. The bands used for the case study are illustrated in Figure 5.5. The exact choice of frequency bands was governed by the availability of filters from the filter supplier, which led to a 50MHz overlap between bands and the high-frequency band being limited to a maximum frequency of 3200MHz based upon the filter specification.

5.5 The design of a low-power partial discharge detector

Based upon the requirements discussed above, a prototype multi-channel RF detector was built for laboratory testing. This prototype is capable of generating three envelope pulses, which can be sampled by an oscilloscope and stored for subsequent analysis. This approach allowed the technique to be validated before proceeding to an integrated design and, in addition, initial test data could be analysed for any features that may be useful for automated diagnosis.

5.5.1 Functional overview

An overview of the detector circuit is shown in Figure 5.6, and a functional block diagram is shown in Figure 5.7 with an enlarged view of a single channel's constituent parts. Supplementary circuit diagrams can be found in Appendix A. Each detector channel is capable of converting wide-band RF signals to an output pulse which approximates the RF signal envelope, and is broken down by function as follows:

Band pass UHF filter: a filter, or combination of filters, is attached to the in-

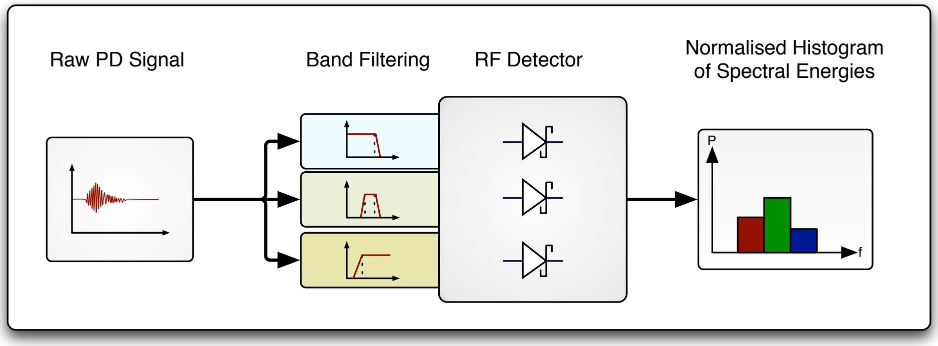


Figure 5.6: Functional overview of the PD detector

put of each channel to filter the input signal frequency range to match one of the bands defined in Figure 5.5.

Detector: At the core of the detector circuit is a Schottky diode, which transforms the PD input signal into a pulse representing the envelope of the input signal. Schottky diodes are commonly used in RF applications as their low forward-voltage drop equates to a low bias voltage required for the diode to operate in its non-linear region, and their fast switching time compared to conventional diodes is suitable for high-frequency applications such as RF detection [177]. The detector time constant was set at $1\mu s$; this is the appropriate duration of a typical PD pulse, matching the specification for narrow band IEC 60270 pulse detection defined in [30].

Low-pass filter: to filter the output of the detector, removing stray UHF noise that may be present.

By combining three detector circuits in parallel, each with different filter bands, the resultant output is three envelope pulses, whose magnitudes are related to the energy within each frequency band. These pulses can then be sampled and stored for analysis.

An example of the output pulses generated by the partial discharge detector can be seen in Figure 5.8. In this figure, the raw PD pulse has a regular logarithmic PD pulse shape decay. The pulse shape is a function of the rate of change in the electric field at the defect site, which in turn is governed by the underlying physical ionisation process. In addition, the geometry between the defect and the sensor, as well as propagation effects, can effect the observed PD pulse shape. The regular shape observed here is due to the controlled experimental conditions in which the measurment was made, which is reflected in the shape of the three pulse envelope signals. However, observed PD pulses

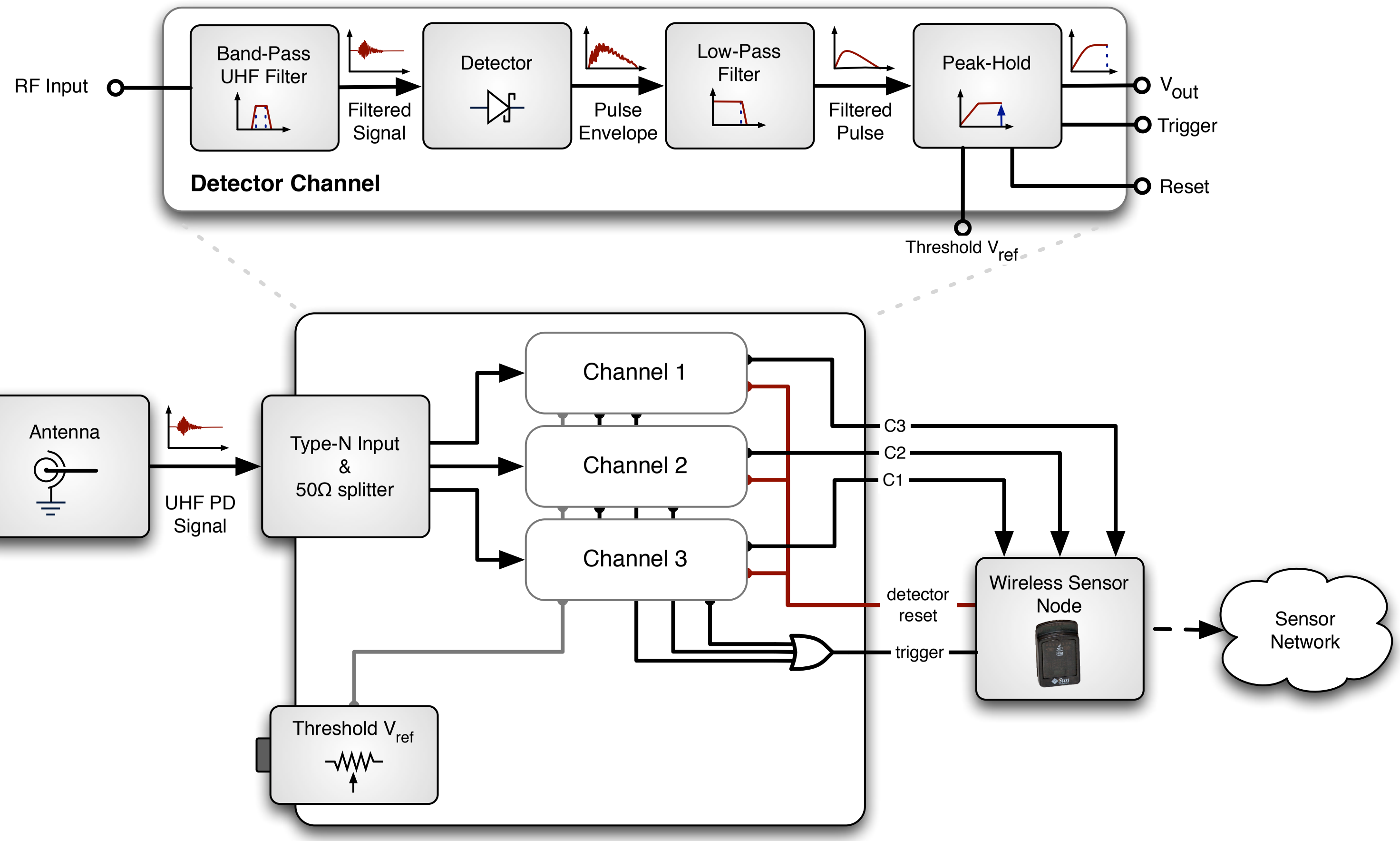


Figure 5.7: PD detector block diagram with enlarged view of a single detector channel

in the field will contain more complex and irregular PD pulse shapes. This is discussed further in Section 2.4.2 b).

In Figure 5.8, each pulse can be seen to have a different peak magnitude. This represents the different spectral density across the three frequency bands, and its use as a diagnostic aid is explained further later in this chapter. Also, it should be noted that the voltage drops below 0V after the PD pulse because the detector is capacitively coupled to the measurement oscilloscope, therefore the positive charge delivered during the RF pulse is replenished by a small current in the opposite direction over a longer period of time.

5.6 Detector experimental method

To establish the efficacy of the frequency-based method, a series of laboratory experiments were carried out to determine the effectiveness of the PD detector, and to establish whether it could differentiate between both different defect types and multiple defects at different positions. Testing of the device was carried out within a laboratory environment using an air-filled 1.3m x 1.3m x 2.3m metal enclosure, shown in Figure 5.9(a) with a number of test cells designed to simulate different defect types.

The test tank was fitted with three monopole RF sensors, as well as featuring a removable sensor panel which can accept a number of different RF PD sensor types, including dipole, spiral and disc coupler types. The capacitive disc coupler sensor, shown in Figure 5.9(b), was attached to the tank for use in these tests, as it was previously found to have the best sensitivity across the entire frequency band [178].

5.6.1 Experimental method

Three partial discharge test cells were used for the experiments, modelled upon the following three defect types found in GIS: protrusion in SF_6, rolling particle in SF_6, and floating electrode in SF_6.

Each of the SF_6 cells consists of a perspex cylinder between two aluminium plates—top and bottom—which form a pressure vessel in which the simulated defect is situated. The cells were each filled with SF_6, pressurised at 2 bar for each experiment. The cells were previously developed to provide reliable constant sources of partial discharge for laboratory and on-site testing of the UHF technique [40]. A schematic of a test cell is shown in Figure 5.10, and two

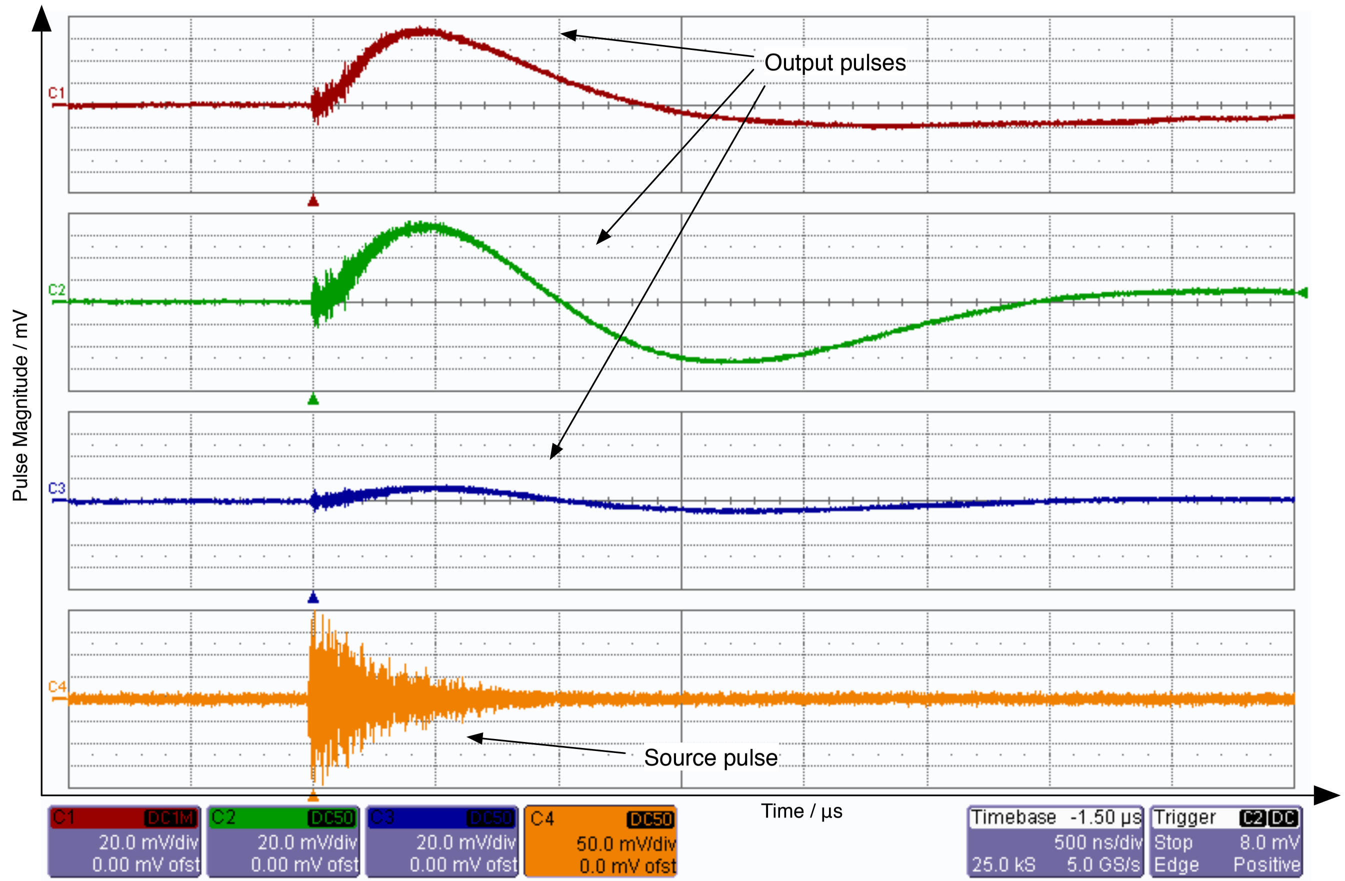
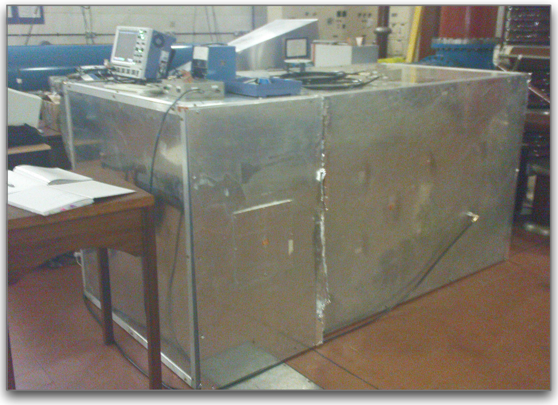
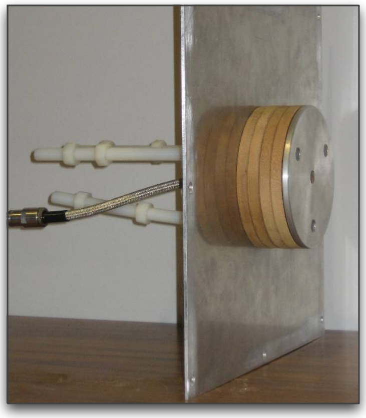


Figure 5.8: Annotated oscilloscope trace showing output pulses generated by the partial discharge detector. The timebase is set to 1μs per division, and the pulse magnitude per channel is 50mV/div for the input pulse and 20mV/div for the output pulses.



(a) External view of the aluminium PD test enclosure



(b) Disc coupler RF sensor

Figure 5.9: PD test enclosure and disc coupler sensor

of the actual test cells used for experimentation can be seen in Figure 5.11.

Each of the test cells were connected, in turn, to a 50kV Foster transformer as shown in Figure 5.12. These were then energised at up to 15kV rms (depending on the PD inception voltage) to generate PDs within each of the cells. The test cells were placed at two positions within the test tank, illustrated in Figure 5.13(a). The fixed position of the transformer and the length of the high-voltage cables were the limiting factors in positioning the test cells. To safely operate the transformer, it was necessary to position it directly beneath the sensor hatch so its controls could be manipulated from outside the test tank. With the transformer position fixed, the available locations for the test cells were immediately in front of the sensor hatch and in the centre of the test tank.

As well as being tested in two positions, the floating electrode test cell was oriented in three planes to simulate the RF emission of an individual defect propagating in different directions. This was carried out to see how the defect orientation might affect the recorded frequency spectrum. The defect orientations are shown in Figure 5.13(b).

The 3-channel detector was fed from a single sensor using a pair of N-type coaxial T-pieces. It was necessary to vary the level of external amplification or attenuation for each of the test cells as there is no internal gain compression within the RF detector. In future iterations of the detector, gain compression may be added to match the wide dynamic range of the PD signals with that of the envelope detectors. Signals were captured on a 4-channel 1GSs^{-1} LeCroy oscilloscope, and archived to disk for further analysis.

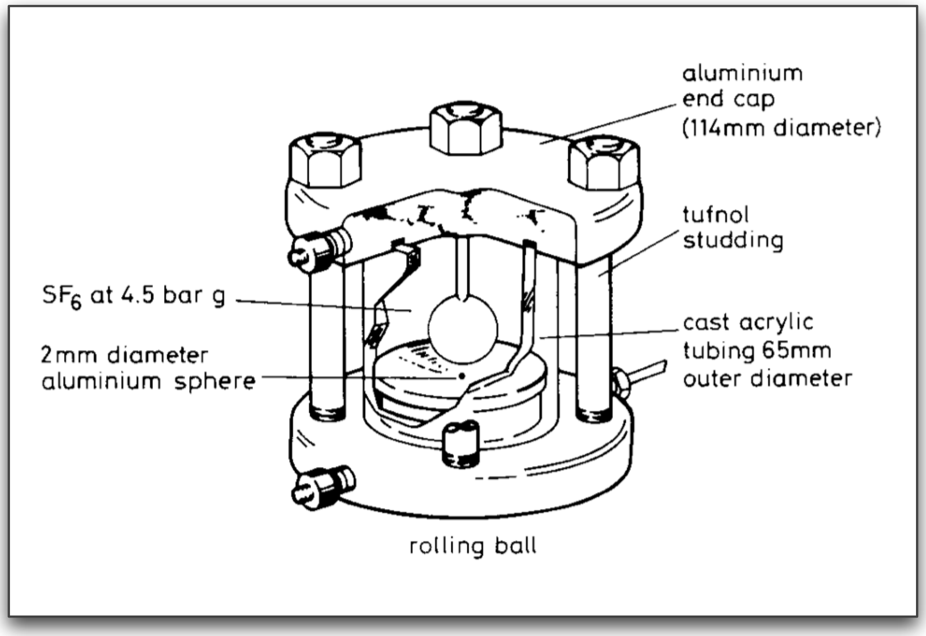
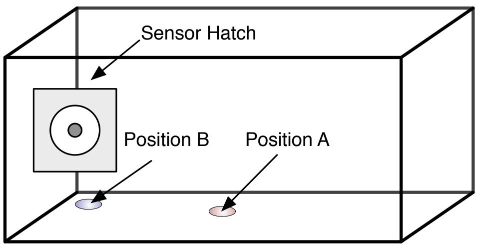


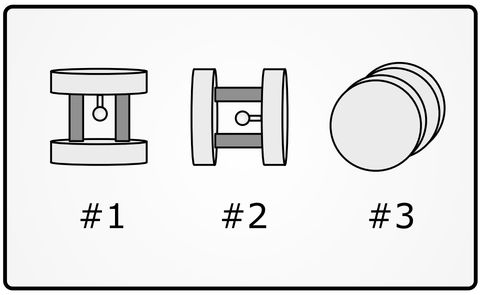
Figure 5.10: Rolling ball (rolling particle) cell schematic [40]

Figure 5.11: Examples of the test cells used for experimentation. To the left, the rolling particle test cell, and to the right, the floating electrode test cell

Figure 5.12: Internal view of RF test tank, with 50kV transformer and one of the test cells on an insulation platform behind



(a) Defect positions within the test tank



(b) Defect orientations used with the floating electrode test cell

Figure 5.13: Defect positions within the test tank, and defect orientations used with the floating electrode test cell

5.7 Data analysis

The conventional method of measuring the RF signal energy of an entire PD pulse is to capture an entire pulse digitally, then integrate the square of the voltage divided by the RF transducer impedance with respect to time; from the pulse inception at time $t = 0$, through the duration of the pulse. In most cases, the RF transducer impedance is 50Ω:

$$E_{RF} = \int_0^t \frac{f(V^2)}{50} dt \tag{5.1}$$

However, as the PD detector described here is designed for simplicity, instead of measuring the absolute RF energy in each band, a representative value approximating the RF energy is calculated.

Referring back to Figure 5.8 which shows the input and outputs of the detector, three rectified pulse envelopes are generated at the output of the detector for each PD pulse, each representing the PD pulse split into one of three frequency bands. It is the peak voltage magnitude of each of these pulses that is used for diagnosis.

These peak voltages are not directly mathematically related to the spectral energy in each band (the pulse shape is ignored), however by normalising the voltages against the total signal magnitude, the resulting proportional values are representative of the RF energy. This is achieved by summing the individual channel magnitudes, then dividing each channel voltage by the sum. The results detailed in this chapter will demonstrate that these representative spectral energy proportions are sufficient to carry out PD diagnosis.

This process can be further explained by example. Consider the following three voltages captured by the detector, generated by the rolling particle in SF_6 test cell:

- 0-450MHz band: 0.129V
- 400-750MHz band: 0.068V
- 700-3200MHz band: 0.141V

Summing these values results in the absolute peak signal magnitude:

$$\begin{aligned}\sum_{i=1}^{3} V_i &= V_1 + V_2 + V_3 \\ &= 0.098 + 0.204 + 0.129 \\ \sum_{i=1}^{3} V_i &= 0.431V\end{aligned} \tag{5.2}$$

The absolute magnitude, 0.431V, can now be used to normalise each channel voltage, generating an approximation of the spectral energy in each band:

$$\frac{V_1}{\sum_{i=1}^{3} V_i} = 0.098/0.431 = 0.227\% \tag{5.3}$$

$$\frac{V_1}{\sum_{i=1}^{3} V_i} = 0.204/0.431 = 0.473\% \tag{5.4}$$

$$\frac{V_1}{\sum_{i=1}^{3} V_i} = 0.129/0.431 = 0.299\% \tag{5.5}$$

The proportional values can be combined with the absolute magnitude into a 4-tuple to create a feature vector representing the data captured in a single measurement:

$$Features_{1..4} = \{\frac{V_1}{\sum_{i=1}^{3} V_i}, \frac{V_2}{\sum_{i=1}^{3} V_i}, \frac{V_3}{\sum_{i=1}^{3} V_i}, \sum_{i=1}^{3} V_i\} \tag{5.6}$$

This results in the following feature vector representing the original 3 detector outputs:

$$Features_{1..4} = \{0.227, 0.473, 0.299, 0.431\} \tag{5.7}$$

Converted into standard form shown in Equation 5.6, it is possible to visualise and analyse this pulse frequency data using the first three of these features.

5.7.1 Visualisation

Each point within a normalised proportional data set of dimension *n* falls within an *n - 1* dimension simplex. In this case, where 3 dimensional data is being used, the proportional sample values fall within a triangular 2D simplex, illustrated by example in Figure 5.14. This 2D simplex, or ternary plot, allows 3-dimensional data to be plotted on a flat surface to simplify interpretation, and is analogous to the technique used to visualise gas-in-oil ratios, which underpins the Duval's Triangle method [24]. Here, the ternary plot can be used to visualise the proportions of three PD pulse measurement frequency bands.

As the dimensionality of the data is critical to the simplex plot, it is only possible to use this particular technique for three dimensional data. For a larger number of dimensions, it would be necessary to apply a dimension reduction technique such as Sammon mapping [179] or Principal Component Analysis. [180]. However, for three-dimensional proportional frequency data, the simplex method was found to be effective.

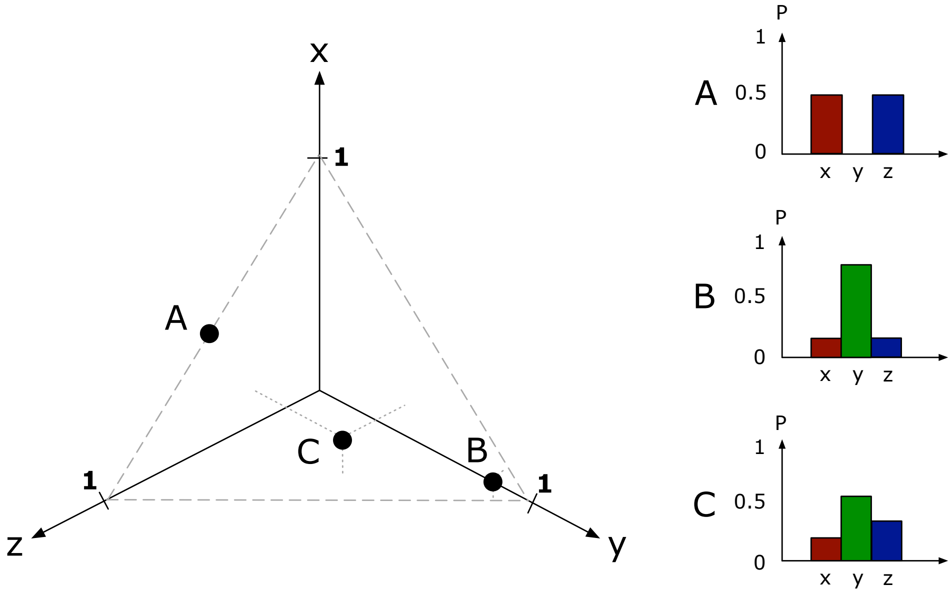


Figure 5.14: Ternary plot with equivalent histograms. Equivalent histograms are shown for three 3-dimensional points plotted within the 2-dimensional simplex

5.8 Experimental results

As discussed previously, the proportional PD pulse magnitudes are used to represent the approximate spectral energy in each of the three frequency bands. The results from tests of each PD defect test cell will now be discussed in turn.

5.8.1 Protrusion in SF_6

The protrusion in SF_6 test cell contains a vertically protruding needle from one of the conductors. When energised to inception voltage, corona discharge is visible at the tip of the needle. The results captured using the detector can be seen in Figure 5.15, which shows that the recorded spectra form a distinctive 'wedge' shape, where the spectrum varies in intensity between the lower and middle bands, but has less than a 10% proportion of high-frequency content.

We can see that the results fall into two distinct clusters, corresponding to two positions. The presence of two distinct clusters, purely based upon position, suggests that multiple protrusion defects in SF_6 may be distinguished based upon frequency spectra. It should be noted, however, that the geometry of the test tank is completely different to GIS, so further tests would be required to validate this across a range of geometries.

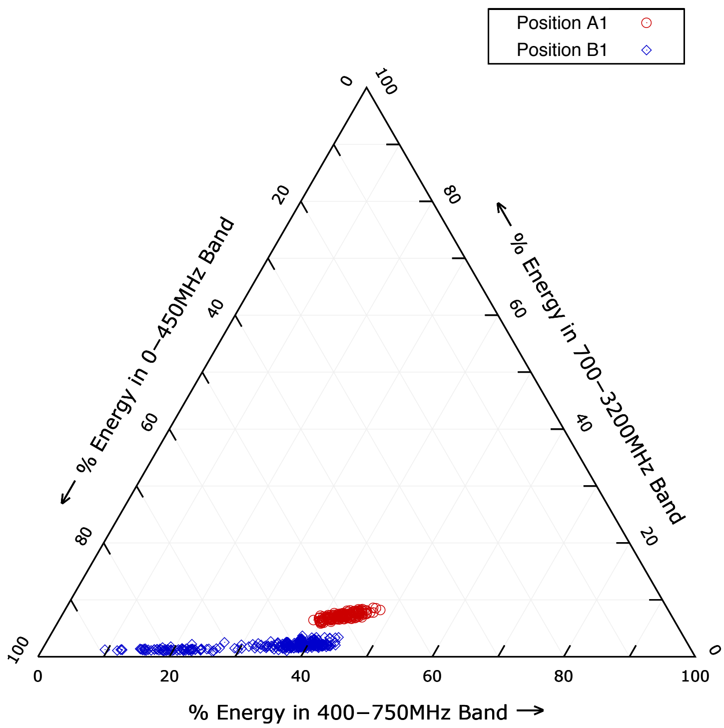


Figure 5.15: Ternary plot showing UHF signal frequency components from the protrusion in SF_6 test results. 152 measurements were taken at position A1 and 313 measurements at position B1

5.8.2 Rolling particle in SF_6

The rolling particle test cell contains an aluminium ball rolling on a concave plate at the bottom of the cell, an example of which is shown in Figure 5.10. When activated, PDs occur between the base-plate and the ball. Recorded PD emissions for this test cell can be seen in Figure 5.16. The results from both positions fall within the same region of the chart, with half the spectral energy falling within the centre 400MHz-750MHz band. The remaining spectral energy varies between the upper and lower bands based upon position. Results from each position are consistent with minor variance, falling in tight clusters.

There is a clear separation between results from each position, with no overlap between clusters. This infers that the frequency spectrum of the rolling particle defect test cell is consistent over time, and the measured spectrum varies only with location. As the frequency spectra of fixed defects does not change over time, and free particles in SF_6 may move within the GIS enclosure, an observed change in frequency spectrum in a single observed defect would therefore infer that the defect is a free particle and is mobile.

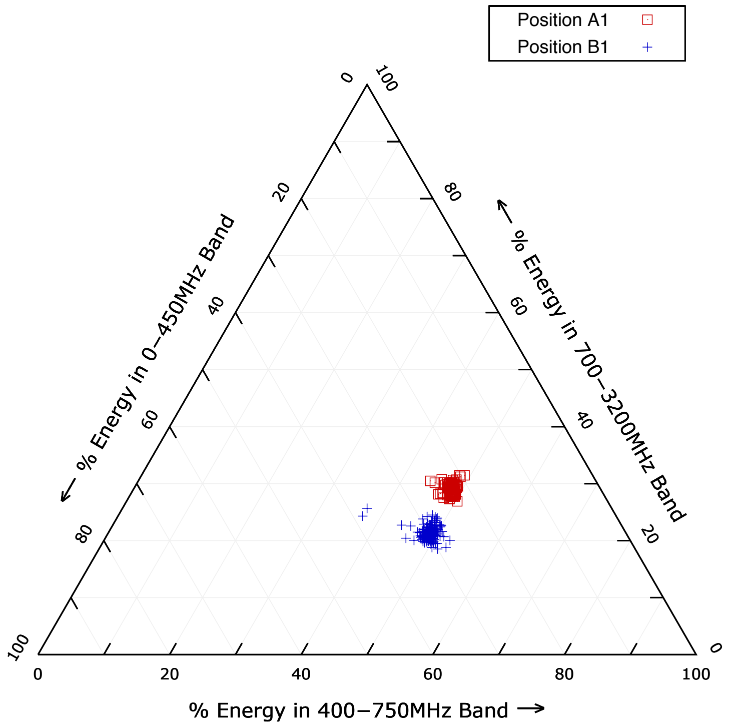


Figure 5.16: Ternary plot showing UHF signal frequency components from the rolling particle in SF_6 test results. 100 measurements were taken at position A1 and 212 measurements at position B1

5.8.3 Floating electrode in SF_6

The floating electrode test cell, shown in Figure 5.11, contains a polythene ring which holds a metal electrode approximately 1mm from one of the energised plates. When activated, a PD (visible spark) occurs between the plate and this electrode. The RF emissions from each PD pulse were also by far the largest of all the test cells, requiring a 10dB attenuator to be inserted betwen the coupler and the detector circuit. As the discharges from this test cell were consistent and of a large magnitude, testing with this cell was carried out using multiple orientations as well as positions A and B. The orientations used are defined in Figure 5.13.

PDs generated by this test cell were found to form a tight cluster, with their spectra largely consistent even across multiple positions and orientations, as shown in Figure 5.17. The defect orientation is shown to have little effect on the recorded RF spectrum, although the test tank is mostly free space, so propagation direction may have less of an effect than in a more complex structure such as a power transformer where solid materials would be present.

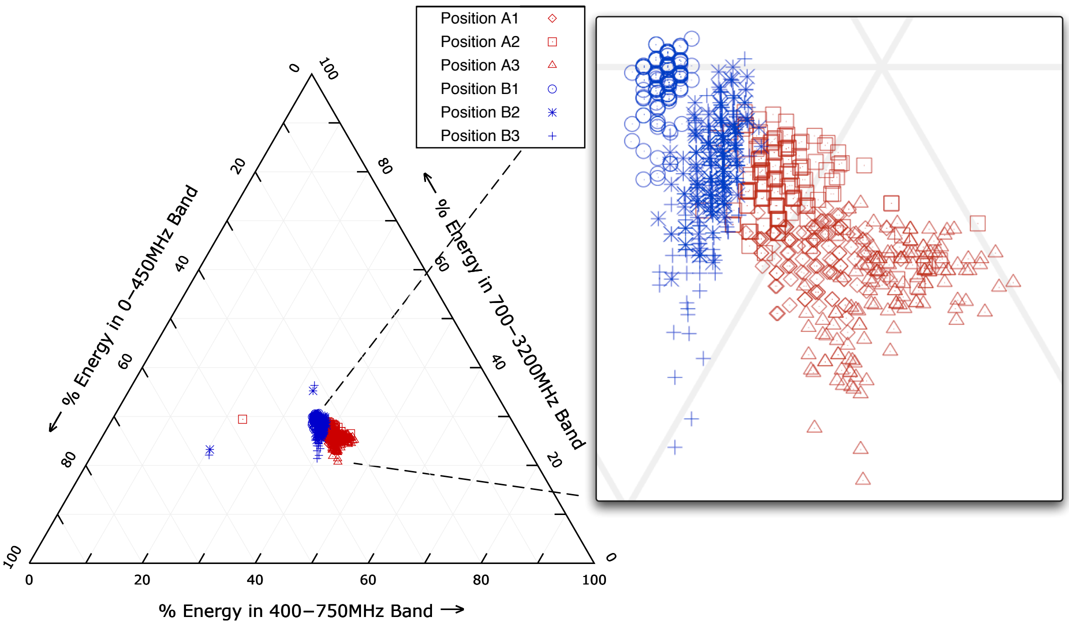


Figure 5.17: Ternary plot showing UHF signal frequency components from floating electrode in SF_6 test results, with an enlarged view of the main data cluster. The following number of measurements were taken at each position: 157 at A1, 168 at A2, 173 at A3, 154 at B1, 156 at B2 and 156 at B3

5.8.4 Discussion

The results presented above demonstrate that a ternary plot can be used to assist the interpretation of frequency spectra of different PD defects, where 3 frequency bands are used. Figure 5.18 shows all of the experimental results on a single chart, which are summarised in Table 5.1.

Both the protrusion in SF_6 and rolling particle in SF_6 results form two distinct clusters for the different positions. As the defect position was the only variable altered between these results, it is clear that the tank geometry can have an effect on PD pulse frequency spectrum. Several assertions can be made based from the results, as follows.

For a piece of plant, if multiple defects are present in separate locations, their observed frequency spectra should fall into separate clusters based upon the geometric relationship between the defect source and sensor, even if the defects are of the same type. This can be used to the advantage of monitoring engineers, as observing multiple clusters using a frequency-based diagnostic method would therefore signify the presence of multiple defects.

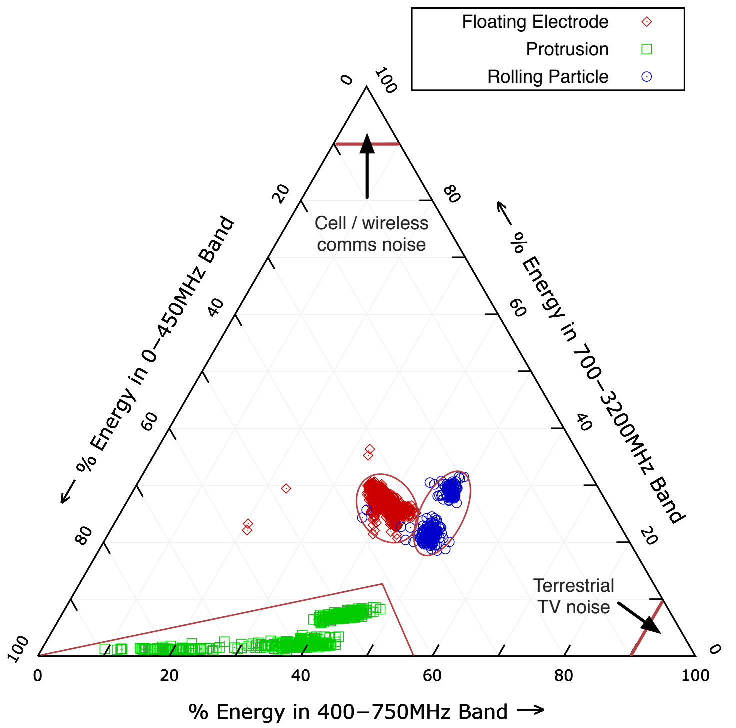


Figure 5.18: Ternary plot showing the UHF signal frequency components from all experimental tests, with specific marked regions indicating defect types and potential noise sources

Table 5.1: Summary of PD detector experimental results

	Frequency band			
Defect	**Low**	**Mid**	**High**	**Characteristics**
Protrusion in SF_6	40-90%	0-50%	0-10%	Irregular
Rolling particle in SF_6	20-40%	45-55%	18-32%	Regular
Floating electrode in SF_6	30-40%	30-50%	20-30%	Regular

However, this was not the case for the floating electrode defect, where the recorded frequency spectra fell into 6 clusters. Floating electrodes typically emit relatively high signals compared to other defects, so combining the signal magnitude as an additional feature may allow for more accurate analysis in this case.

The floating electrode and rolling particle defects have a similar frequency fingerprint, although the results given in this Chapter are still linearly separable. In practice, it may not be possible to distinguish these defect types based upon frequency alone, although it is clear from the results that the protrusion defect has a unique frequency fingerprint which can be easily classified using this technique.

These results demonstrate that frequency spectra of partial discharge RF emissions do contain defect-specific information. For this fact to be exploited and the technique to be applicable to a wireless sensing application, it is necessary to integrate this detection technique with a wireless sensor node hosting an online diagnostic algorithm.

5.9 Sensor node integration

For the frequency-based PD method to be effective, the detector must function as part of an integrated wireless sensor. To achieve this, it is necessary to consider issues relating to sampling, to ensure that the data generated by the detector can be digitised.

5.9.1 Sensor node sampling

In Section 5.3.1, we have seen that it is necessary to use an ADC with a sampling rate of at least 2MHz to accurately capture a digital representation of a UHF PD signal envelope. As the ADC speed of current state-of-the-art sensor nodes is in the kilohertz range, it is not possible to capture the entire pulse with

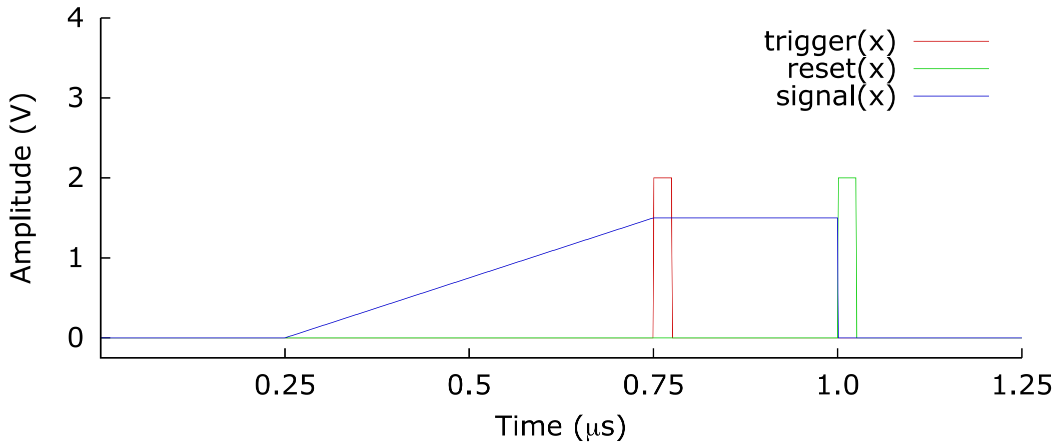


Figure 5.19: Peak hold process overview

this type of device. However, a peak hold circuit can be used to sample and hold the maximum peak value of the PD pulse, so that the magnitude may be sampled at a lower sampling frequency. As well as signal input and output, peak hold circuits also include trigger and reset functions; to capture the peak value at the output, and subsequently reset the detector output to zero once it has been sampled. This process is illustrated in Figure 5.19. Further background information on peak detector circuits can be found in [181].

Peak detection is advantageous in low-power applications as pulse magnitudes can be captured using a sampling rate below that of the frequency of the pulse envelope. When a value above a certain threshold is detected, the sensor node can be triggered to sample the values being held by the peak detector. Sampling of features using this method is not only limited to magnitude; by employing a different conditioning mechanism, such as a frequency-to-voltage converter, it is, for instance, possible to measure the maximum frequency of a signal.

As well as allowing a lower sampling rate to be used, this approach also removes the requirement for a sensor node to sample continuously, further reducing the energy consumption of the sampling process. This technique can be applied generically to other sampling applications, where the input signal is passed through an analog conditioning circuit to create an output pulse which can then be captured by a peak detector.

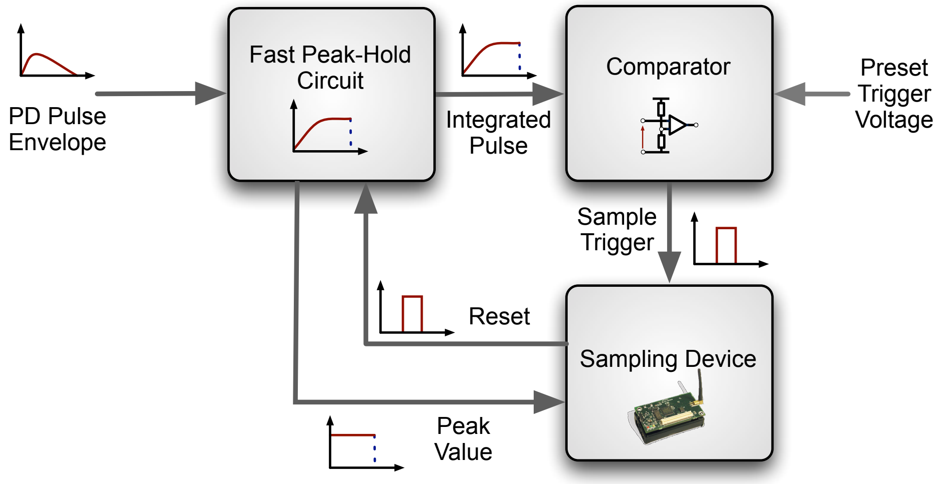


Figure 5.20: Overview of the PD detector integration circuit

5.9.2 Sample triggering and detector resetting

Peak detectors hold the maximum voltage for an appreciable period, while the buffered peak value is sampled and until the detector buffer voltage is pulled to ground. To trigger a sensor node to sample this value, it is necessary to use a comparator circuit to generate a pulse when the detected voltage reaches a preset level. After sampling has occurred, the detector buffer must be returned to 0V. An overview of the triggering and reset circuits can be seen in Figure 5.20.

Implementing a sampling regime using a peak detector to intermittently sample continuous, high-frequency events introduces certain constraints. When employing a peak detector and reset circuit, the effective sampling rate is limited by the sampling rate of the sensor node ADC, plus the trigger and reset latencies. In Figure 5.21, we can see an illustration of this, with the key time points being:

The pulse trigger time, t_t: which is the point at which the circuit responds to the PD pulse magnitude exceeding the threshold crossing. ;

The sample time, t_s: which is the point at which the peak value is sampled after an interrupt has been generated. This delay, t_s-t_t, is dependant on the latency between the sampling hardware measuring a change in voltage to generating a software interrupt; and,

The reset time, t_r: which is the point at which the peak detector is reset, and its buffer value returns to zero. This delay, t_r-t_s, is the latency between

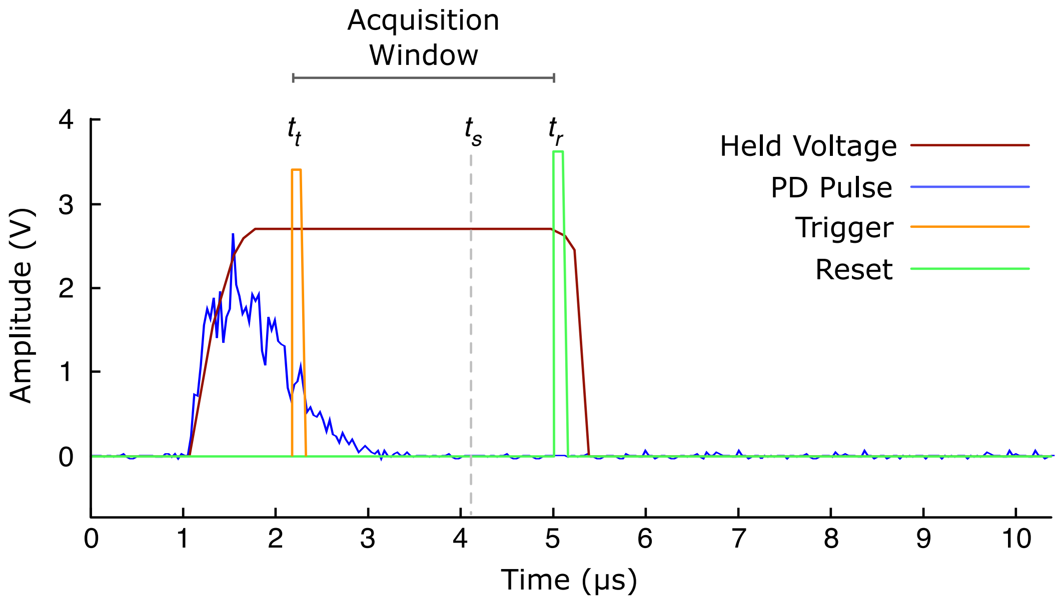


Figure 5.21: Detector acquisition window, showing the partial discharge pulse and detector signal voltages

the sampling software dispatching a reset pulse in software, to the pulse being generated in hardware.

The detector acquisition window length, which is the time between the threshold point, t_t, and the detector reset, t_r, is the limiting factor in RF detection using this method. During this period, any subsequent RF pulses that are measured by the detector circuit will be superimposed on previous values. This was experienced when attempting to integrate the PD detector design with the SunSPOT during laboratory testing, and is a constraint that will always be present in sampling applications such as this that use a peak detector.

The specification laid out above demonstrates that the PD detector can be integrated with a sensor node with appropriate circuitry, however, unfortunately it was not possible to integrate the PD detector fully with the SunSPOT device. It should be noted, however, that despite the detector and sensor node not being fully integrated, the fundamentals behind the PD detection and diagnostic technique still stand, as proven by the results shown in Section 5.8. Outstanding design issues and suggested improvements to the detector design are detailed in the following section.

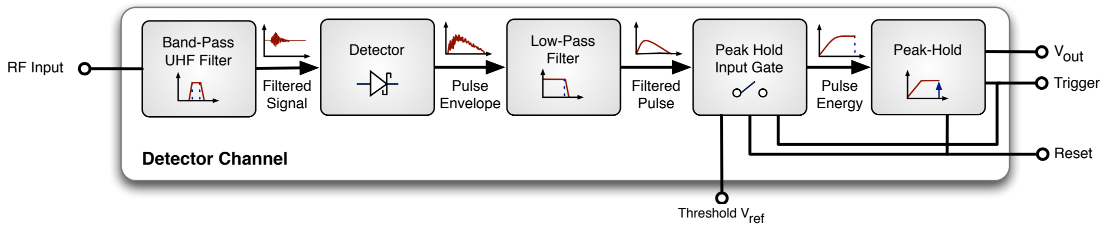


Figure 5.22: Proposed addition of an peak hold input gate component to the PD detector design to ensure accurate sampling by a sensor node

5.9.3 Outstanding issues with the PD hardware integration

The existing detector design requires that an additional control mechanism is added to the peak detector to ensure that consecutive PD pulses can be differentiated. By disabling the input to the peak detector as soon as a PD has been detected, subsequent PD pulses can be ignored. This can be achieved with the addition of a digital switch at the input of each detector channel, as illustrated in Figure 5.22. This will affect the operation of the PD detector by disabling its inputs at the rising edge of the sample trigger, to then be re-enabled at the falling edge of the reset pulse. While limiting the rate at which PD pulses can be measured, it will, however, ensure that all pulses that are measured are accurate.

In addition to this, a higher speed of sampling must be achieved. In the case of the SunSPOT device, the ADC has a maximum physical sampling rate of 22KHz, translating to a sampling period of approximately 45μs. To sample PD pulses using the detector, this would require the acquisition window, shown in Figure 5.21, to be at least this length. Unfortunately, due to the latency between the SunSPOT device main processor and ADC, this is further reduced to a practical sampling rate of 8.1KHz [182], translating to a maximum acquisition window length of around 123μs.

Therefore, a faster ADC should be used to optimise the acquisition window length. If the full 22KHz of the SunSPOT device could be exploited, the sample time (t_s) component of the acquisition window can be minimised to around 45μs. Alternatively, a separate ADC may be used to achieve a higher sampling rate, however this would add an additional power requirement to the integrated design.

5.10 Summary

To fully exploit wireless sensor network technology for substation condition monitoring applications, it is necessary to choose the right mix of sensing techniques appropriate to each plant item. While previous research and industrial deployments have shown that it is possible to use wireless sensor networks to monitor primitive measurands such as temperature and vibration, this chapter has demonstrated that the RF partial discharge detection technique can be adapted for low-power usage so that it may be feasibly applied in a wireless environment.

Previous approaches to RF monitoring have focused on using wide-band sampling methods to capture large amounts of high-frequency data, which is subsequently analysed using data mining or intelligent diagnostic techniques. The approach presented in this chapter differs from traditional methods as it identifies and classifies impulsive noise events using a very small amount of data by comparing approximate spectral energies within specific frequency bands for a single pulse, rather than accumulating a large number of pulses upon which pattern recognition or other techniques are applied.

This method is not as accurate as wideband methods, simply because necessarily captures only a subset of data features. However, it does not require wide-band signal capturing and conditioning hardware which is both resource intensive and expensive and cannot economically be deployed across numerous pieces of plant. Comparing this method with conventional methods, on the one hand, broadband UHF diagnostic systems have been shown to provide accurate defect diagnosis and even defect location, but on the other hand, they are not currently widely used in the field due to cost, additional data volume produced and practical constraints. The PD detection method described in this chapter cannot only detect instances of multiple concurrent PD sources, but is also readily deployable into a low-power package which could be installed onto any high-voltage plant which already features a compatible RF sensor.

The issue of 'information overload' is significant when increasing condition monitoring sensor coverage; the partial discharge detection technique presented here introduces yet another data format which must be understood and analysed by monitoring engineers. To simplify this process, a novel visualisation method based upon a 3-axis ternary plot was developed. This allows defects to be analysed and compared on a 2-dimensional chart, which maps the data effectively for subsequent interpretation and data analysis.

This chapter has demonstrated that it is possible to implement an effective low-power PD detection system, and that low-power detectors are feasible that can give an indication of the internal insulation health of electrical plant. This type of technique is not limited to partial discharge diagnosis, and could potentially be extended to other CM applications. The key generic outcome of this is that if suitable features are identified for a particular monitoring application that can be effectively measured using a low-power detector, then the technique can be used within a wireless sensing context. Using this approach could potentially pave the way for other diagnostic methods in the future that could enhance the field of substation condition monitoring.

Partial discharge detection and the detector demonstrated in this chapter, however, would only constitute a single component of a larger wireless condition monitoring system. For engineers to be able to access monitoring information generated by such sensors, a suitable software architecture must be in place to support diagnostic algorithms, sensor network management and integration with existing condition monitoring systems. Chapter 6 reports upon these issues, demonstrating that diagnostic algorithms and data management can be deployed directly on sensor nodes, and through a multi-agent approach, flexible wireless condition monitoring sensors can be integrated with existing agent-based monitoring systems.

Chapter 6

An integrated wireless condition monitoring architecture

"If you want to make an apple pie from scratch, you must first create the universe."

- **Carl Sagan, 1934-1996**

6.1 Chapter overview

The previous chapter demonstrated that it is possible to implement a low-power condition monitoring sensor that could potentially be deployed as part of a wireless network. To meet the needs of utilities, any new substation condition monitoring system such as this must ideally be flexible enough to integrate with existing systems. This chapter presents a wireless condition monitoring architecture which allows diagnostic capabilities to be deployed at the sensor level where sensor data is processed into meaningful information, and only pertinent information is passed on to monitoring engineers as required. This is then transmitted to an existing condition monitoring architecture where it can then be disseminated to monitoring engineers.

This system is realised through the combination of two complementary technologies: wireless sensor networks and multi-agent systems. In this chapter, we will see how these two technologies can be practically combined, and how an agent-based approach to wireless sensor networks can bridge the gap between the type of low-power diagnostic method described in the previous chapter with existing condition monitoring architectures, effectively delivering diagnostic information from the sensor direct to monitoring engineers.

The practical application of these concepts are then illustrated in the following chapter through a number of case studies.

6.2 The need for a wireless condition monitoring architecture

When deploying any new condition monitoring system, its data will inherently introduce some level of burden to monitoring engineers. If too great a volume of monitoring data is collected without appropriate filtering, it becomes increasingly difficult to extract meaningful information; this problem of 'information overload' has already been approached for SCADA alarm systems, where automated alarm processing has been achieved through the application of AI techniques such as rule-based and model-based systems [183]. For wireless sensor technology to be applied to substation condition monitoring, data management must also be applied. In this case, however, it can be embedded at the source to manage sensor data as soon as it is captured. This approach is especially applicable to wireless sensing applications where data transmission must be optimised to conserve energy.

For utilities to be able to implement data interpretation and diagnostics at the sensor level, suitable tools are required. Chapter 3 has detailed the advantages of using a multi-agent approach for diagnostic applications. In Section 4.5 the drawbacks of the current breed of agent platforms for sensor networks were discussed, where agents are typically written in assembly-style code making both algorithm development and code maintenance a non-trivial task. For this type of approach to be adopted by the power community at large to practically develop and deploy wireless sensing systems, high-level languages should be used so that the full gamut of modern programming techniques may be exploited.

In the past 10 years, the increase in capabilities of wireless sensors has led to the latest breed of devices supporting high-level languages, for instance Java on the SunSPOT platform [135]. This approach to embedded systems presents all low-level hardware and network functions through a unified software interface, greatly simplifying application development compared to alternatives such as C-based micro-controllers. In turn, this simplifies the deployment of high-level diagnostic capabilities on embedded hardware.

To demonstrate that this is possible, the remainder of this chapter outlines a specification for a wireless agent toolkit upon which intelligent monitoring applications can be built. In the following chapter, the functionality of this toolkit is demonstrated through a number of case studies.

6.3 SubSense: a wireless multi-agent system for condition monitoring

To address the problems detailed above, a novel diagnostic platform for sensor networks, called 'SubSense', was designed to meet the following requirements:

- The need to effectively manage monitoring data at the sensor level;
- The need to deploy intelligent diagnostic capabilities at the sensor level; and,
- To provide the tools necessary to build wireless condition monitoring applications.

To implement this, the SunSPOT wireless sensor network platform was chosen as a target architecture as it is capable of executing Java code natively, allowing the same code to be written for a desktop or server which can be deployed with little modification to a sensor node.

In the following sections, a brief overview of the toolkit is given, and the key benefits towards building enhanced condition monitoring applications are demonstrated. A full user manual and technical guide of the system can be found in Appendix B.

6.3.1 Architectural specification

An architectural overview of the SubSense agent toolkit is shown in Figure 6.1. The system design is loosely based upon the FIPA standards and the JADE platform, with the aim of providing a familiar model with which to develop agents. The architecture contains the following key components:

Agent Platform: A wrapper for all application agents, including two system-level agents; the `AgentManagementService` agent and `Directory Facilitator` agent which control agent lifecycle, registry and service listings;

Agent prototypes: An API containing agent supporting services, upon which application agents can be easily built;

Agent behaviour prototypes: A set of behaviours on which routine actions can be built, such as one-shot behaviours, periodic behaviours, and behaviours that present a service to other agents; and,

Functional components: Common functions used by agents for communication, storage, and physical sensor interfacing.

6.3.2 Building an agent-based wireless diagnostic system

Using the SubSense toolkit, the low-power partial discharge detection technique described previously in Chapter 5 can be wrapped and exposed through a standard software interface. Within the four components of a condition monitoring system, the partial discharge detector and RF transducer represents the 'sensor component': using the SubSense toolkit, the remaining components—data capture, identification and diagnosis—can be implemented to wrap the entire diagnostic functionality as an autonomous software agent.

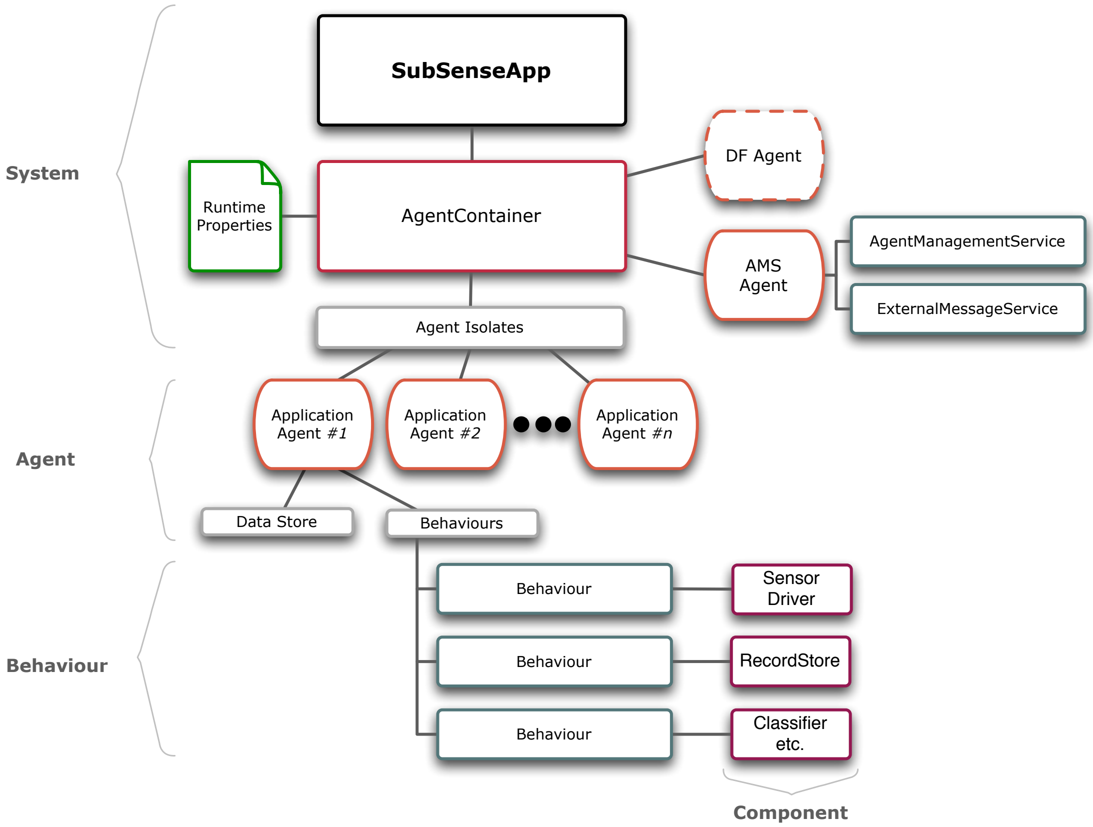


Figure 6.1: Overview of the SubSense architecture

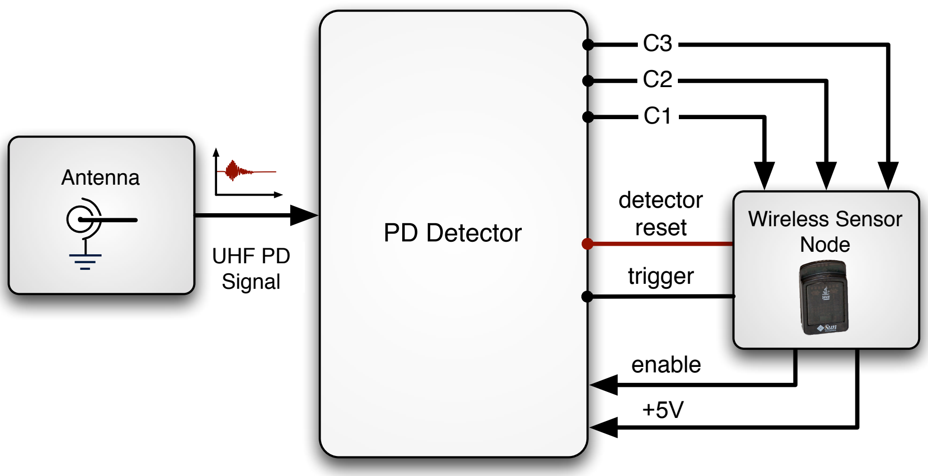


Figure 6.2: PD detector and sensor node signal diagram, showing inputs and outputs from the sensor node to the PD detector

Consider the PD detector and sensor node signal diagram shown in Figure 6.2 (shown previously in full in Section 5.5.1). For detectors such as this to be integrated with diagnostic applications, it is necessary to encapsulate these signals and provide software access to the physical PD detector. The SubSense platform achieves this through a standard driver and behavioural model that allows new sensor data to be acquired either periodically or based upon a threshold (the technical details of which are discussed in-depth in the SubSense manual in Section B.3.4).

Using the SubSense platform and the PD detector as an application example, it is now possible to easily integrate the four components of a condition monitoring system on a single sensor node running one or more data capture and diagnostic agents. In this case a single agent, `PDSenseAgent`, can be used to capture PD data and generate on-sensor diagnoses from sensor data. The four condition monitoring components are represented as follows (illustrated in Figure 6.3):

Sensing: As a PD pulse is sensed, the analog values representing the frequency spectrum of a PD pulse are generated by the detector, also triggering a threshold event;

Data Capture: The wireless sensor node receives the threshold trigger signal, which is configured as an input to the `PDSenseAgent`. This spawns a new one-shot data capture behaviour to capture the analog PD pulse values and generate a new digitised measurement; and,

Identification and Diagnosis: `PDSenseAgent` diagnostic behaviours are executed, generating a diagnosis which is persisted to the sensor node flash memory.

Table 6.1 shows the volume of data generated for a single PD measurement. In total, 48 bytes would be generated for each partial discharge event. This compares to 21,500 bytes generated for each PD event within the COMMAS system [184]. It should be noted that, as discussed in Section 3.5.1, the COMMAS system uses a different PD method along with multiple diagnostic classifiers, however it is clear that the SubSense system can lead to agent-based monitoring applications which generate PD diagnoses with 4 orders of magnitude less data than existing systems. This has the knock-on effect of being inherently more scalable than conventional systems, as the communication burden associated with scaling from one or two plant items to an entire substation becomes minimal.

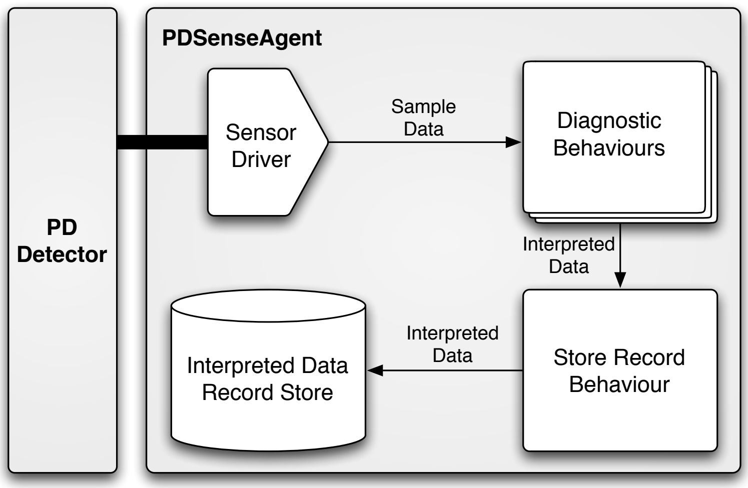


Figure 6.3: Example SubSense system architecture for a PD monitoring application

Table 6.1: Data generated at each stage within SubSense for PD pulse diagnosis

Stage	Fields	Data size (bytes)
Sensing	Analog values	—
Data Capture	4 frequency pulse values, timestamp	36
Diagnosis	defect code, timestamp	12

6.3.3 Data dissemination and integration with existing monitoring systems

In addition to on-sensor diagnosis, it is necessary to integrate sensor nodes with existing condition monitoring systems so that diagnostic information may be disseminated to monitoring engineers. Established in Section 4.5.3, FIPA communication protocols are unsuitable for sensor network operation, therefore the SubSense platform uses a messaging system based upon the Active Message model (described in Section 4.5.2). This is a bit-efficient protocol which prefixes the message payload with a simple message identifier, used to indicate the size and structure of the message. As this protocol is of a completely different nature to the FIPA protocol suite, SubSense employs a sensor network gateway to allow agents operating within the sensor network to interact with FIPA-compliant multi-agent systems.

6.4 Sensor network gateway

The sensor network gateway consists of a FIPA-compliant gateway agent and a physical sensor node configured to act as a base station, providing an interface between the FIPA-based multi-agent system and the physical wireless sensor network. This new agent fills the requirement laid out in Section 4.5.2 for a proxy between bit-efficient sensor networks and FIPA compliant agents. This is illustrated in Figure 6.4.

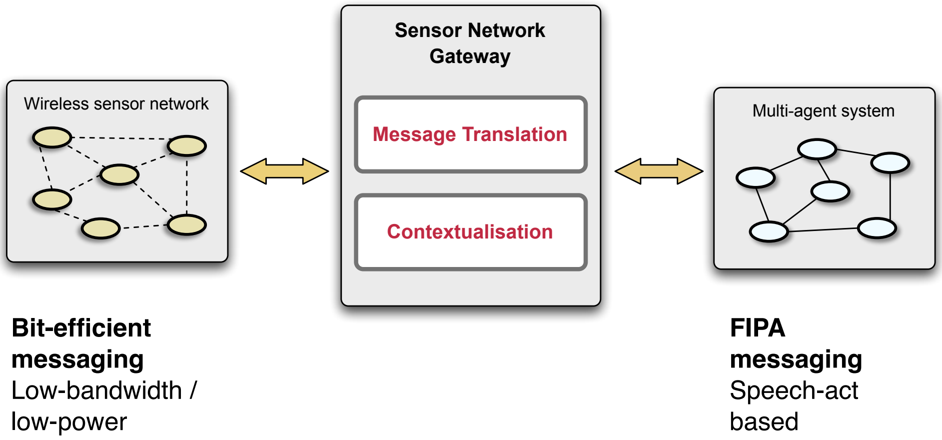


Figure 6.4: The SubSense sensor network gateway that translates between bit-efficient and FIPA message formats

Sensor network traffic must be highly optimised, therefore static contextual information relating to sensor data, such as details of the plant under observation, should ideally not be transmitted with each measurement. Through optimising message payloads, the energy cost of data transmission can be lowered, but through removing data this leads to received messages losing some of their associated meaning. This presents an issue when data is received by the sensor network gateway, as key elements pertaining to the data must be added to provide context to the measurement value.

For instance, if an oil temperature measurement is received by the gateway, it may only include the measured value, timestamp, and source address of the sensor node. An agent that calculates top-oil temperature is not necessarily concerned with the details of the physical sensor node which took the measurement, but it will be interested in exactly which transformer the measurement originated from. It is, therefore, necessary to add relevant information to incoming data packets upon reception at the sensor network gateway so that analysis and interpretation may be carried out in context.

6.4.1 Message translation

The sensor network gateway is capable of translating incoming messages from the sensor network. This is best illustrated by example. Consider a measurement generated by a temperature sensor to be transmitted to a base station for archival, whose payload is broken down by field in Table 6.2.

Table 6.2: Breakdown of a temperature measurement packet

Field	Size (bytes)	Description
Sensor ID	4	A unique ID indicating the sensor
Type ID	4	The active message ID, indicating 'temperature'
Value	8	The temperature value
Timestamp	8	The time at which the measurement was taken
Total:	24	

For this message to be understood by an agent-based system upon reception by the sensor network gateway, it is necessary to convert it to an agent 'fact' pertaining to a concept which has been defined in the agent ontology. The sensor network gateway does this by using a unique message type ID to establish which concept the sensor data relates to, so that it can add context to the values stored within. In this case, the concept type and units must be associated with the temperature scale (e.g. Celsius or Kelvin). With this information, the gateway can generate a `Measurement` fact which can be properly interpreted by an agent. A temperature measurement (from the COMMAS v2 ontology) is represented as follows:

```
(Measurement
   :measurementType "temperature"
   :measuredValue (Value
      :value "32.5" :unitSymbol "C")
   :timestamp 1274719774)
```

Figure 6.5: Example Temperature fact in FIPA-SL notation

Message translation between external data sources and a FIPA multi-agent system has previously been achieved within the COMMAS v2 condition monitoring system (discussed in Section 3.5.8) to allow web services to be integrated with the multi-agent system. This is carried out using an 'ontology map', which statically maps each message type to a particular ontology concept using a lookup table.

Figure 6.6 shows an example of a static ontology map from the COMMAS v2 system, used to translate data for a Kelman Transfix online DGA monitor that includes a list of fault gases, plus water, along with the associated class, measurement units, source and chemical composition. This table can be used to programmatically populate the fields of the `DissolvedGas` concept, shown in Table 6.3.

Table 6.3: `DissolvedGas` concept fields

Field Name	**Class**	**Inherited from**
value	String	`Value`
unitSymbol	String	`Value`
unitMultiplier	String	`Value`
chemicalComposition	String	-
percentage	PerCent	-
gasName	String	-
ppm	float	-

This method is re-used and extended within the sensor network gateway to add contextual information to data as it is received by the sensor. The sensor network gateway is capable of dealing with any message type, as long as a suitable message handling behaviour is included. To integrate with the COMMAS v2 system, two message handlers are required, to handle both `Measurement` and `InterpretedData` concept instances, which represent raw sensor readings and on-sensor diagnoses respectively.

For each message type, a unique identifier must be defined. The message type IDs and concept types are defined in Table 6.4.

Table 6.4: Active message types and ontology concepts for COMMAS v2 integration

Active Message ID	**Ontology concept**
10	`Measurement`
30	`InterpretedData`

6.4.2 Application ontology

To support wireless sensing applications while reusing the upper ontology discussed in Section 3.4.2, Figure 6.7 shows an agent ontology for wireless sensing applications which is built upon the upper ontology for power engineering

```
map = {{"Hydrogen", "DissolvedHydrogen", "ppm", "Transfix", "H2"},
       {"Carbon Monoxide", "DissolvedCO", "ppm", "Transfix", "CO"},
       {"Ethylene", "DissolvedEthylene", "ppm", "Transfix", "C2H4"},
       {"Methane", "DissolvedMethane", "ppm", "Transfix", "CH4"},
       {"Ethane", "DissolvedEthane", "ppm", "Transfix", "C2H6"},
       {"Acetylene", "DissolvedAcetylene", "ppm", "Transfix", "C2H2"},
       {"Oxygen", "DissolvedOxygen", "ppm", "Transfix", "O2"},
       {"Water", "DissolvedWater", "ppm", "Transfix", "H2O"}}
```

Figure 6.6: An example of a static ontology map for a Kelman Transfix

applications, with key changes to the existing ontology to make it suitable for wireless operation. In this ontology, additional concepts introduced are:

MeasurementValueSource: a CIM class type representing an entity which generates measurement values;

SingleSample: representing a single measurement, for example temperature or humidity;

MultiChannelSample: a sample containing multiple values, for example the multi-channel frequency-based PD detector introduced in the previous chapter;

SensorNode: representing a node in the sensor network; and,

Sensor: representing a particular sensor attached to a sensor node. Multiple sensors can belong to a single sensor node.

Specific changes have been made to upper ontology concepts used within this ontology, including the removal of all string-based fields to optimise data volumes. For instance, in the case of the `FaultClassification` concept, the `technique` field is represented by an integer; the sensor network gateway interprets this, converting the integer value to a string representation using the mapping technique described above.

This ontology supports monitoring an on-sensor diagnostics, however it falls short of features supported by sensor networking standards such as SensorML. Recommendations for the integration of CIM and SensorML standards are presented in Section 8.2.

6.4.3 Gateway architecture

To achieve the translation tasks described above, the agent uses a set of reactive behaviours to configure the gateway and read messages from the sensor network and generate ontology facts. The behaviours are as follows:

LoadObjectIdentifierBehaviour: A one-shot behaviour which loads a list of `ObjectIdentifier` instances from a database. `ObjectIdentifier` is a CIM concept used as a unique identifier for items connected to the electrical network. In this case, the `ObjectIdentifier` objects contain references to the unique address of each sensor node, enabling them to be matched up to each sensor network packet upon reception;

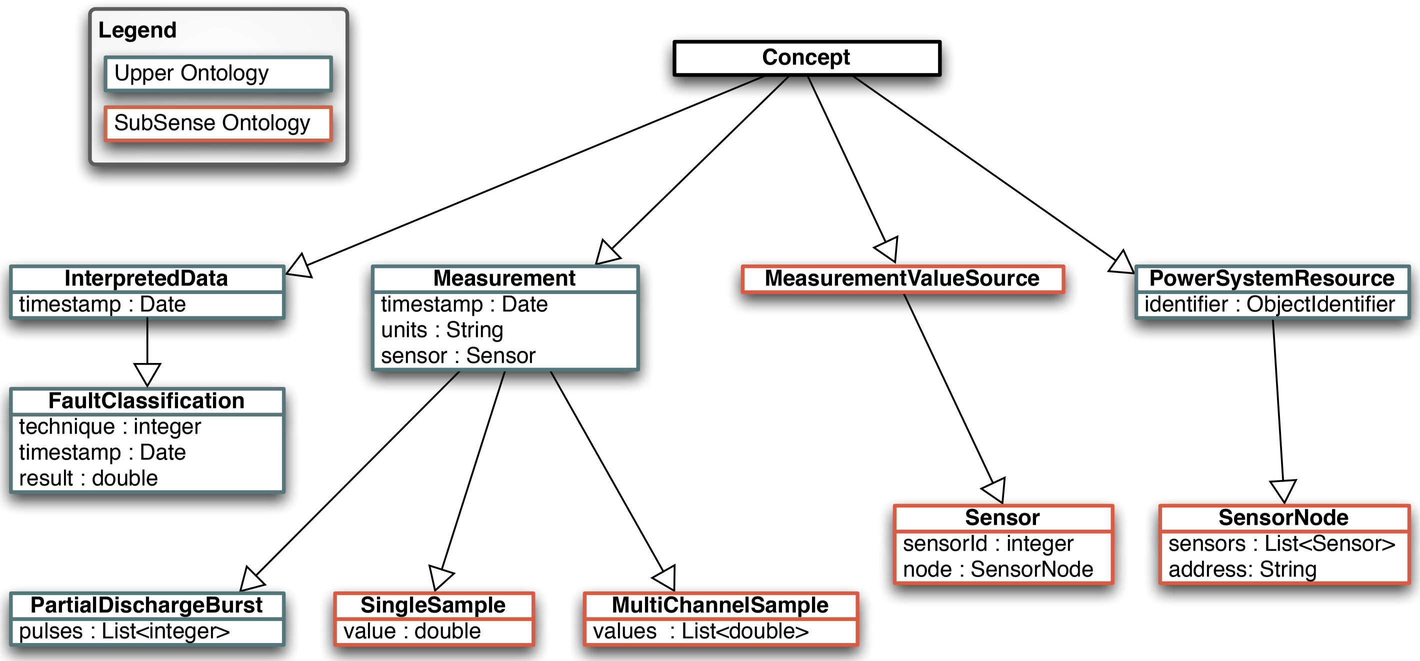


Figure 6.7: Extensions to the upper ontology for power engineering applications for SubSense applications

FindBasestationAddressBehaviour: A one-shot behaviour which locates the attached physical sensor gateway node device; and,

ReceiveSensoryDataBehaviour: A cyclic behaviour which initially instantiates the physical sensor network gateway, then listens for incoming sensor network messages. Upon receiving a new message, a message handler behaviour is dispatched to convert the received message to an ontology fact.

To add application-specific functionality for the COMMAS v2 multi-agent system, the following behaviours are added to carry out message translation of specific message types and pass on the converted data to other application agents:

SingleSampleMessageHandlerBehaviour: Converts raw sensor samples to `Measurement` facts, and asserts the new fact within the sensor network gateway knowledge base. Single samples are raw sensor measurements which are sent directly from a sensor node;

MeasurementQueryResponder: Responds to FIPA `REQUEST` messages for `Measurement` facts from other agents, providing measurement data upon request. This behaviour was taken verbatim from the COMMAS v2 system and added to the Sensor Network Gateway without modification;

InterpretedDataMessageHandlerBehaviour: Converts interpreted data messages to `InterpretedData` facts, asserting them in the sensor network gateway knowledge base. Interpreted data messages contain defect diagnoses generated by agents executing on sensor nodes; and,

ArchiveConceptBehaviour: A one-shot behaviour which requests that the `InterpretedData` facts generated by the gateway are stored by the `ArchiveAgent`, a core component of the COMMAS v2 system.

Together, these behaviours are capable of translating bit-efficient messages into FIPA-SL format, providing services to pass on the data to other agents as necessary. Figure 6.8 illustrates how the components of the sensor network gateway interact to translate messages and disseminate monitoring and diagnostic information. This process is further illustrated in the next Chapter, which contains three different monitoring applications using the SubSense toolkit.

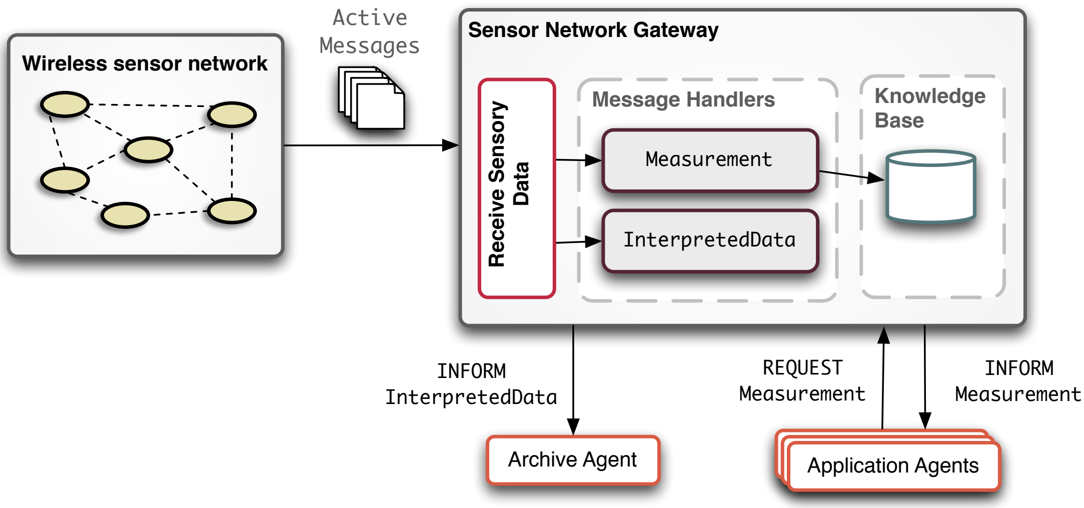


Figure 6.8: Sensor network gateway operation, showing message flows between sensor nodes and application agents.

6.5 Summary

This chapter has described the steps required to integrate a wireless sensor network with a FIPA-based multi-agent condition monitoring system. Firstly, a novel toolkit for building agent-based sensor network applications has been presented, which allows the multi-agent methodology to be applied down to the sensor level. The problem of integrating agent-based sensor networks with FIPA multi-agent systems has been approached through the implementation of a sensor network gateway, which translates between bit-efficient sensor network data to FIPA-compliant messaging. Together, these components provide a solid foundation upon which substation condition monitoring applications using wireless sensor networks can be built.

The following chapter describes how these components are put into practice through three case studies, demonstrating their versatility and applicability to a number of monitoring and diagnostic applications.

Chapter 7

Case studies

"Practice, the master of all things."

- **Augustus Octavius, 63 BC-AD 14**

7.1 Introduction

In this chapter, three case studies are presented which build upon the concepts described in the previous chapters. These demonstrate an enhanced approach to substation diagnostics using low-powered sensing techniques coupled with agent-based wireless sensor networks. Each of the case studies includes a different monitoring application using the SubSense platform and sensor network gateway to enhance the COMMAS v2 condition monitoring system, used for data dissemination through its web-based user interface.

The first case study describes an implementation of a laboratory demonstrator designed for temperature and vibration monitoring. The second case study expands upon this, by showing how the frequency-based partial discharge detector, described in Chapter 5, can be integrated with on-sensor diagnostics to identify partial discharge defects on-sensor. The final case study demonstrates the versatility of this approach, by deploying components of the original COMMAS multi-agent system (discussed in Section 3.5.1) on the SubSense platform. This case study pushes the envelope of the agent-based approach to substation diagnostics through the deployment of an integrated diagnostic system at the sensor level.

To support data translation within the sensor network gateway agent, the data shown in Table 7.1 was asserted into the agent knowledgebase for the purposes of the case studies.

7.2 An integrated temperature and vibration monitoring system

Temperature monitoring has been identified as a prerequisite for substation condition monitoring [14], with applications including ambient temperature and transformer hot-spot temperature. Vibration monitoring has been recommended to measure on-load tap changer (OLTC) health as well as transformer cooling components. The use of wireless sensors in this setting has the potential to add monitoring facilities to plant components which, to-date, have not undergone significant monitoring, reinforcing existing condition monitoring systems. Therefore, a temperature and vibration monitoring demonstrator using the SubSense platform was built within a laboratory environment and integrated with the COMMAS v2 condition monitoring architecture described previously in Section 3.5.8.

Table 7.1: Case study data used for translation within the Sensor Network Gateway

(a) Measurement translation table

Concept ID	Concept name	Units	Sensor name
2	Temperature	C	Sunspot Temperature
5	Vibration	G	Sunspot Vibration
9	PerCent	%	SunSPOT Battery Level

(b) Interpreted data translation table

Concept ID	Concept name	Technique	Indexed defect types
50	PDDefectDiagnosis	Frequency-based Partial Discharge	Floating Electrode in SF6, Particle in SF6, Protrusion in Air-Protrusion in SF6
51	PDDefectDiagnosis	Phase-resolved Partial Discharge	Bad Contact, Floating Electrode, Protrusion,Rolling Particle, Surface Discharge, Suspended Particle

(c) Sensor node data

Node ID	Alias	Node address	Identified object name
6	SunSPOT Temperature and Vibration	0014.4F01.0000.538C	SS1

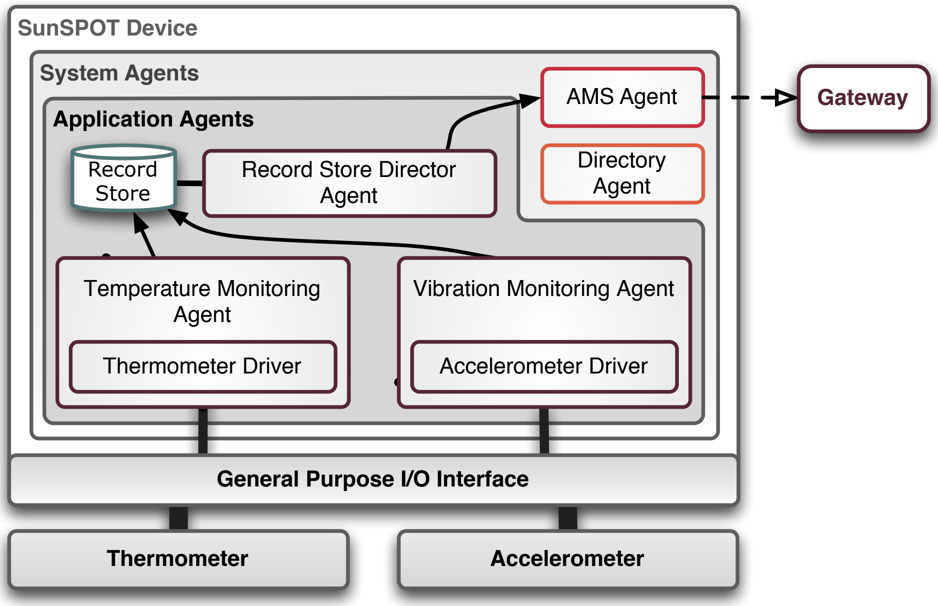


Figure 7.1: SubSense configuration for temperature and vibration monitoring application

The on-board sensors provided by the SunSPOT sensor node were used for this case study, however for an industrial deployment external sensors would be connected to the device I/O bus for more accurate measurement readings. The application architecture is shown in Figure 7.1.

7.2.1 Application agents

The application agents perform individual tasks of periodically sampling temperature and vibration sensors, before transmitting this data to a base station to be archived and displayed to monitoring engineers. The application agent functions are as follows:

7.2.1 a) Temperature and vibration monitoring agents

The temperature and vibration monitoring agents individually combine temperature or vibration sensor drivers with periodic sampling and storage capabilities. Each agent contains a single `SensorDataLoggerBehaviour`, which periodically samples a sensor which is specified at runtime, logging the sensor value in either the `TemperatureStore` or `VibrationStore` record stores [1].

In this application, temperature and vibration monitoring functions are modelled as separate agents to allow their functionality to be easily deployed

[1]A full description of record stores can be found in Appendix B.3.4

depending on the application. However, it would be equally valid to combine their functionality into a single agent which sampled both sensors in parallel.

7.2.1 a) RecordStoreDirectorAgent

The Record Store Director Agent is a general purpose agent tasked with transmitting new data available within agent record stores to remote sensor nodes. In this case, the remote node in question is the base station. The agent contains multiple instances of a single behaviour:

ArchiveBehaviour: listens for new records added to a particular record store, and forwards the data to the AMS agent requesting that it is sent to the base station.

The number of instances of this behaviour is configurable at runtime, dependent on the number of record stores specified to be monitored. In this case, the `ArchiveBehaviour` is reused to monitor both the `TemperatureStore` and `VibrationStore` for new data.

7.2.2 SubSense configuration

The temperature and vibration monitoring application is executed using the configuration file shown in Figure 7.2.

7.2.3 Sensor agent integration

To allow sensor data to be used within a larger condition monitoring system, the sensor network gateway, described in Section 6.4, was added to the COMMAS v2 condition monitoring system described in Section 3.5.8. Simulated sensor and plant information was provided to the sensor network gateway at runtime, allowing data contextualisation to be carried out. For this case study, COMMAS v2 is used purely for its user interface capabilities, however, the data could feasibly be used directly by other diagnostic agents within this system. The resultant agent architecture can be seen in Figure 7.3.

7.2.4 System operation

Running through the complete system operation, the resultant data flow can be observed. The following measurement is a human-readable temperature

```
subsense.basestation.address=0014.4F01.0000.53D3:50

#temperature and vibration agents
subsense.agent.org.poweragents.sunspot.agent.TemperatureMonitoringAgent = Temperature Agent
subsense.agent.org.poweragents.sunspot.agent.VibrationMonitoringAgent = Vibration Agent
subsense.agent.org.poweragents.sunspot.agent.RecordStoreDirectorAgent
        = Archiver,TemperatureStore,10,VibrationStore,10
```

Figure 7.2: SubSense configuration file for a temperature and vibration monitoring application

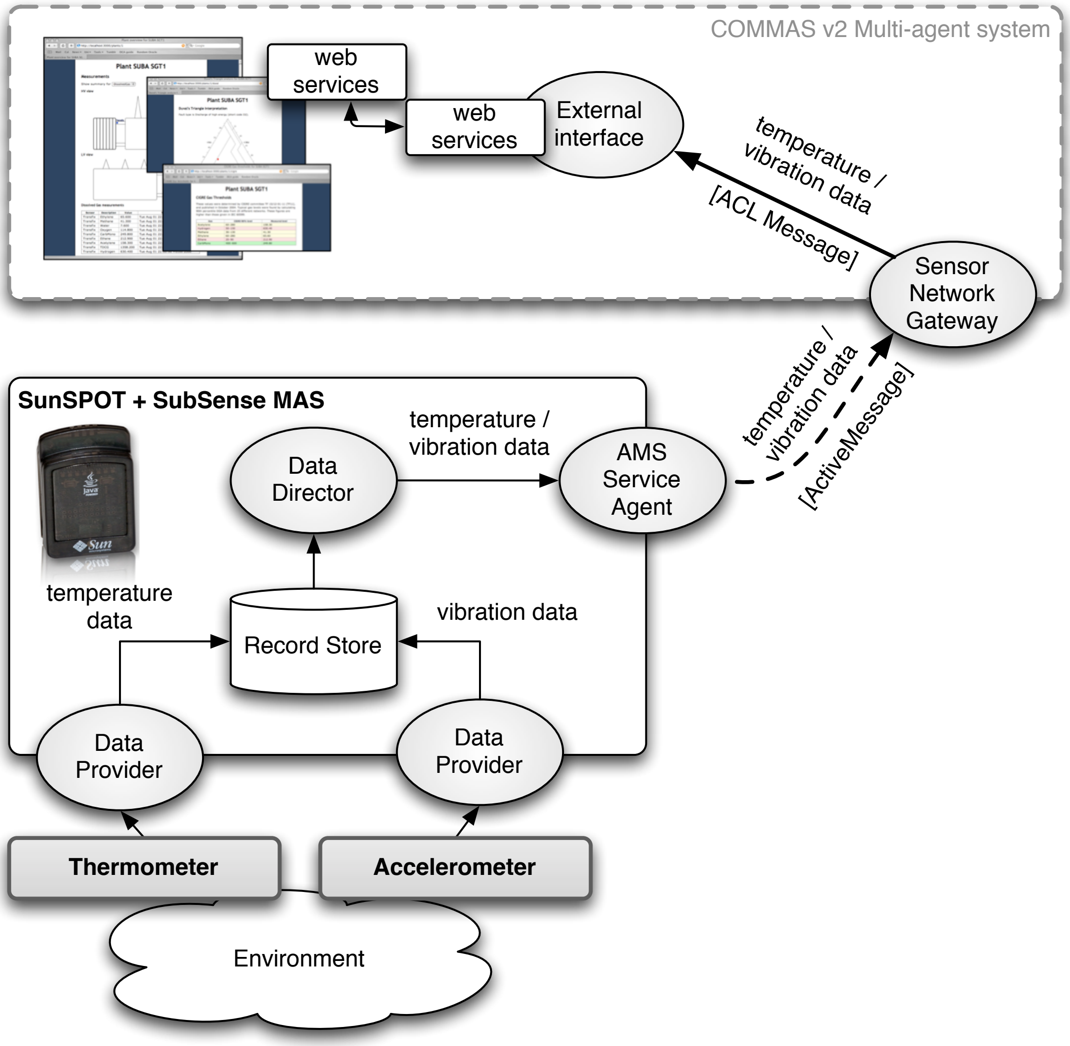


Figure 7.3: SubSense agent, communication path, and system architecture for a temperature and vibration monitoring application

```
(Measurement
   :measurementType Temperature
      :measuredValue (Temperature
         :value "32.25" :unitSymbol C)
   :plant (PowerSystemResource
      :resourceIdentifier (ObjectIdentifier
         :aliasName "SunSPOT Temperature and Vibration"
         :localName "0014.4F01.0000.538C"
         :identifiedObjectName SS1))
   :location "Sunspot Temperature"
   :timestamp (Timestamp :millisSinceEpoch "1276175971925"))
```

Figure 7.4: Temperature fact in FIPA-SL syntax translated by the sensor network gateway

sample generated by the system:

```
Thu Jun 10 15:19:21 BST 2010:  null:25[2] :  32.25 C
```

This is encoded into a 25-byte bit-efficient payload to be transmitted to the sensor network gateway:

$$\overbrace{[0A]}^{packet\ \ type} \quad \overbrace{[00][00][00][19]}^{sensor\ \ ID}\overbrace{[00][00][00][02]}^{sensor\ \ type}$$

$$\overbrace{[00][00][01][29][22][03][F6][55]}^{timestamp}\overbrace{[40][40][20][00][00][00][00][00]}^{temperature}$$

Upon reception by the gateway agent, each measurement is converted to an ontology fact according to the packet type. In the case of the message above, the fact shown in Figure 7.4 is generated, in FIPA-SL syntax.

Table 7.2 shows a comparison of the transmitted sensor node message against the full agent fact generated by the sensor network gateway. From this, we can see that there is approximately a 93% reduction in message size using the bit-efficient message compared to the FIPA-SL message. This in turn leads to an equivalent reduction in transmitted bytes using the active messaging system, while maintaining all of the relevant message information through data integration carried out by the sensor network gateway.

7.2.5 Discussion

This case study demonstrates how a sensor network may be integrated with an agent-based monitoring system to display temperature and vibration data

Table 7.2: Comparison of bit-efficient to FIPA-SL encoded `Measurement` facts, received and generated by the sensor network gateway

Encoding medium	Message size (bytes)	Size of SL message (%)
Active message	25	6.92
FIPA-SL	361	100

to monitoring engineers.

Together, the combined goal of these agents is to disseminate raw temperature and vibration data. This case study does not feature any embedded diagnostics but instead demonstrates that the multi-agent methodology can be used from the sensor level right up to the engineering interface, integrating two agent platforms so that sensor data can be displayed in near-real time to monitoring engineers. The benefit of taking a multi-agent approach in this case is that it is trivial to reconfigure the embedded agents for a different goal; for instance, an additional agent may be added to the sensor node to calculate statistical features of sensor data over time. If a data interpretation agent was added to this system, the Record Store Director Agent may be reconfigured to forward from an alternative store containing interpreted data. With this simple change, the volume of transmitted data can be reduced whilst still providing up-to-date plant data.

To demonstrate that on-sensor diagnostics can be achieved, the second case study details the use of an embedded online classifier for a low-powered partial discharge monitoring application.

7.3 An integrated wireless partial discharge detector

Combining the concepts described in the previous chapters, an integrated wireless partial discharge detector was constructed to combine sensing, data capture, defect identification and diagnosis into a single package. This was designed to make use of the low-power detector discussed in Chapter 5. However, there are outstanding issues for integrating the partial discharge detector with a wireless sensor node, which are discussed further in Chapter 8. Nevertheless, in lieu of a fully integrated detector, a microcontroller-based PD detector simulator was used in its place to simulate the generation of partial discharge events, replacing the sensing and data capture functions while leaving

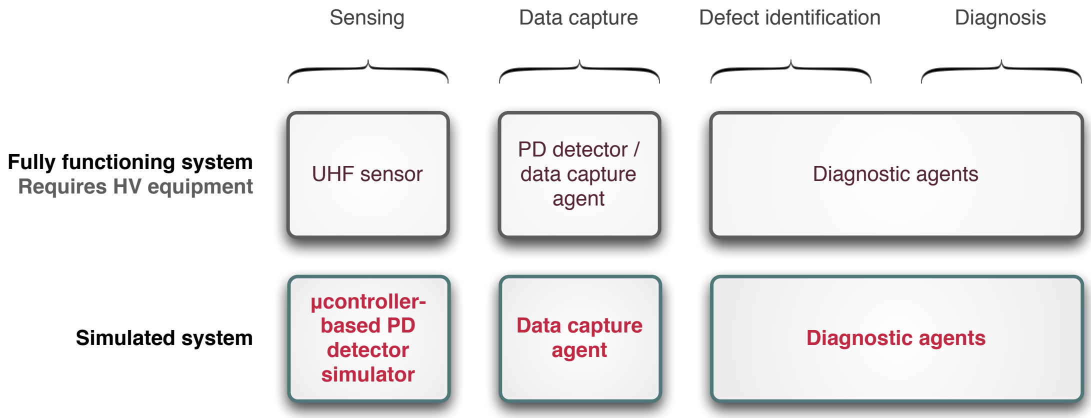


Figure 7.5: Microcontroller-based PD detector simulator block diagram, compared to a fully operational PD system. The microcontroller is used to simulate the physical PD detector functionality in hardware enabling the diagnostic system to be tested without the need for high-voltage apparatus

the rest of the system intact, as illustrated in Figure 7.5.

More information on the microcontroller-based PD detector simulator can be found in Appendix C. In the future, it will be possible to replace it directly with an integrated physical PD detector.

7.3.1 Establishing appropriate diagnostic capabilities

To implement the system, one of the key components required was an on-line diagnostic method suitable for both the sensing application and sensor network operation, which matched the diagnostic technique detailed in Chapter 5. Chapter 4 showed that there are limitations to implementing complex diagnostic systems on embedded platforms, therefore a suitable diagnostic method was identified based upon the experimental outcomes found in Section 5.6.

Within the field of machine learning, there are a number of techniques available for supervised and unsupervised learning. When choosing a technique, it is necessary to choose a technique based upon the specific application. In this case, it was necessary to choose a technique which is computationally light in both memory and processor usage, due to the resource constraints of the underlying sensor network platform.

Figure 7.6, shows all experimental results on a single ternary plot. Within these results we can see that the positions of various defects are linearly separable, within the result space. This simple distinction between result classes suggested that a statistical classifier may be trained from the experimental data.

There are several statistical classification techniques that may be applied to

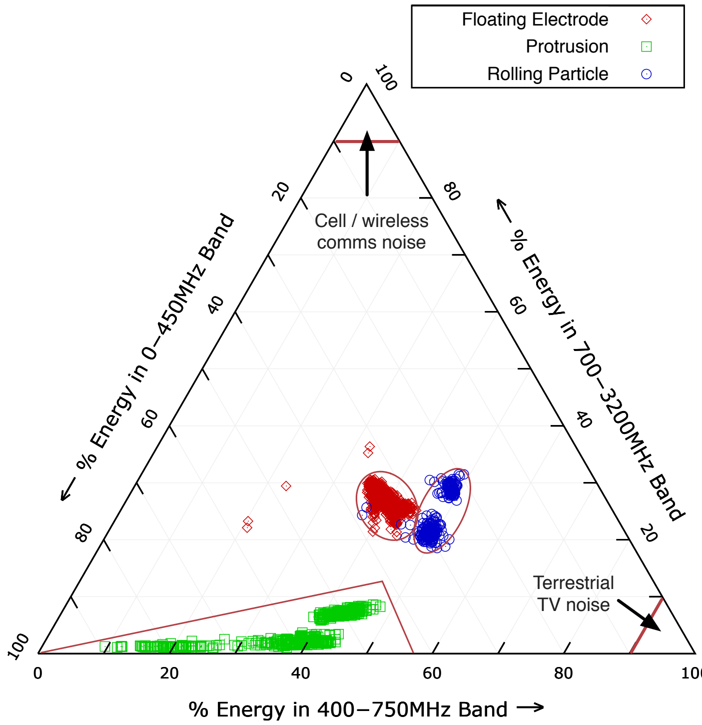


Figure 7.6: Partial discharge detector experimental results

this data, although one long-standing method that was deemed particularly suitable to this application is the C4.5 algorithm.

7.3.2 The C4.5 technique

The C4.5 algorithm, developed by Quinlan [185], is one of the most widely used machine learning techniques due to its maturity and ease of use. Based upon the seminal ID3 rule induction technique [186], it generates a hierarchical tree of decision rules from a set of test data which may be used to classify unseen data into pre-labeled classes.

The advantage of the C4.5 algorithm over other statistical classifiers such as artificial neural networks is that the decision rules generated by the C4.5 algorithm are readable and deterministic, closely matching the human thought process if the test result space was to be manually segmented.

For this application, as the result space is linearly separable it is also possible to manually define production rules (in `if-then` format). The use of an automated classifier, however, is advantageous as it allows training of rules automatically. For instance, if additional training data was made available in the future through access to additional data sources, the classifier could be easily retrained with little manual effort.

Quinlan has provided source code for C4.5, in C format [187]. This was used to train the the decision tree, although it was necessary to build a decision tree executive that could run on a sensor node. Therefore, a cross-platform C4.5 executive was implemented to run in both a PC and SunSPOT environment. This implementation is described in Section B.4.

7.3.2 b) Training the classifier

As described in Section 5.8, experimental data was gathered from three high-voltage PD test cells which represent the following defects: floating electrode in SF_6, rolling particle in SF_6 and protrusion in SF_6. The training process used 50% of the test data for training and 50% for validation, with a 1.1% error rate. This generated a binary decision tree and class label file, which could be executed by the C4.5 executive. This classification capability was then wrapped within an application agent.

7.3.3 Application agents

The partial discharge application consists of two application agents, which together perform the task of PD emission sampling, storage, classification and dissemination. An overview of the SubSense system architecture executing on the sensor node is shown in Figure 7.7.

7.3.3 c) Partial discharge classifier agent

The data capture agent provides a standard interface to the PD detector, receiving sample vectors each time a PD pulse is detected. The agent consists of the following behaviours:

LoadDecisionTreeBehaviour: loads the binary decision tree from flash memory, parsing it into executable form and placing it into the PD agent internal data store;

PartialDischargeClassifierBehaviour: executing every time a partial discharge event occurs, this behaviour receives the PD emission frequency components, passing them to a new `ClassifyBehaviour`; and,

ClassifyBehaviour: executes the C4.5 decision tree using proportional PD frequency data, saving the PD classification to the `InterpretedData` record store by spawning a subsequent `StoreRecordBehaviour`.

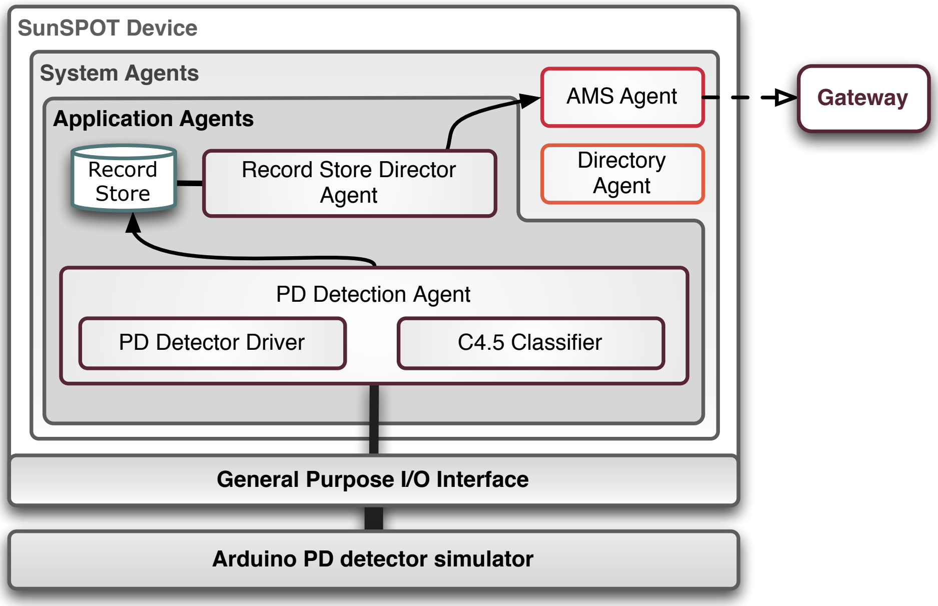


Figure 7.7: SubSense configuration for partial discharge monitoring application

As the classifier used within this agent is identical to one used for PC-based diagnostics, it generates exactly the same results. In this case, however, diagnosis is achieved immediately on the sensor which can be stored locally and transmitted in lieu of the original raw PD data.

7.3.3 c) Record store director agent

The Record Store Director Agent, described in Section 7.2.1 a) above, was reused for this application albeit configured to forward data from the `InterpretedData` record store. This agent forwards interpreted PD data to a base station where it can be displayed to monitoring engineers.

7.3.4 SubSense configuration

The partial discharge diagnostic agents are executed using the configuration file shown in Figure 7.8. This configuration instructs the RecordStoreDirectorAgent to send interpreted partial discharge diagnostic information to the gateway using packet type 30, which indicates that it represents an `InterpretedData` concept, allowing it to be properly converted to an agent fact upon reception.

```
subsense.basestation.address=0014.4F01.0000.53D3:50

#PD sense agents
subsense.agent.org.poweragents.sunspot.pdsense.PDSenseArduinoAgent=PD Sense Test Agent
subsense.agent.org.poweragents.sunspot.agent.RecordStoreDirectorAgent=Archiver,InterpretedData,30
```

Figure 7.8: SubSense configuration file for a partial discharge diagnostic application

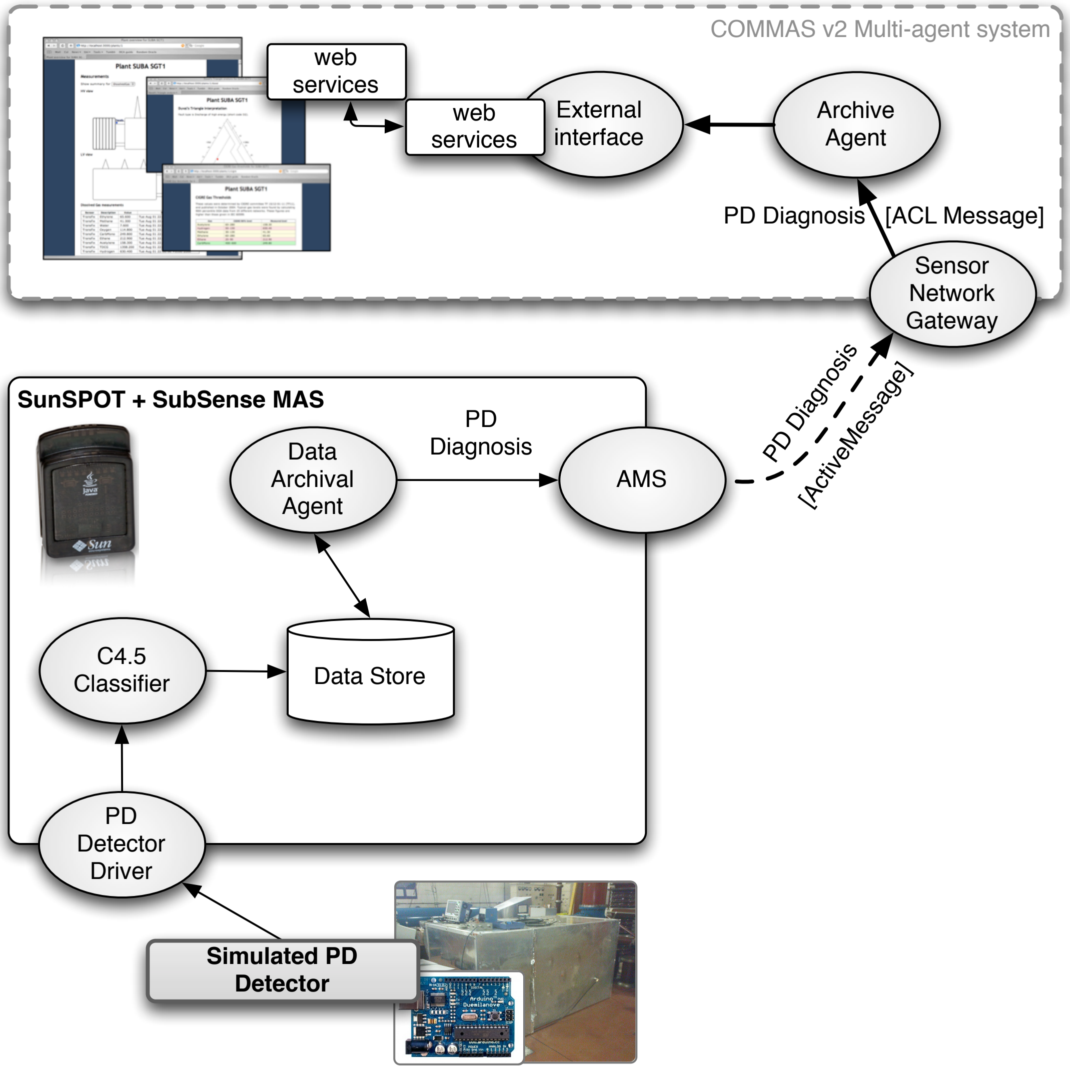


Figure 7.9: SubSense agent, communication path, and system architecture for a PD diagnostic application

7.3.5 System operation

The entire system architecture can be seen in Figure 7.9. Taking the same approach as the previous case study, the resultant sensor data and interpreted information flow can be observed. Using the PD detector simulator, measurement data is introduced to the sensor node via serial bus, in the form:

```
a31,47,22,z
```

This particular measurement, representing 31% energy in the 0-400MHz band, 47% energy in the 400-800MHz band and 22% in the 800-3200MHz band, is from a Rolling Particle defect measured during the frequency-based detector tests. The defect is immediately measured by the PDSenseArduinoAgent, which interprets the frequency data using the C4.5 classifier to generate a defect diagnosis, here represented in human-readable string format:

```
Thu Jun 17 15:23:20 BST 2010, 1276784600318,
        1 Particle in SF6, 0.31, 0.47, 0.22,
```

This data is automatically forwarded to the sensor network gateway in the following bit efficient form:

$$\overbrace{[1E]}^{packet\ \ type}\quad \overbrace{[00][32]}^{\substack{interpretation\\ type}}\quad \overbrace{[00][00][01][29][46][4a][e5][0c]}^{timestamp}\overbrace{[3f][f0][00][00][00][00][00][00]}^{result}$$

Upon reception, the sensor network gateway identifies the packet as interpreted data, converting it to the ontology fact in FIPA-SL syntax shown in Figure 7.10. This fact can then be disseminated to monitoring engineers through the engineer's interface.

```
(PDDefectDiagnosis
   :sourceMeasurementTime (Timestamp
      :millisSinceEpoch 1276784600318)
   :plant (PowerSystemResource
      :resourceIdentifier (ObjectIdentifier
         :aliasName SunSPOT Simulated PD Sensor
         :localName 0014.4F01.0000.5419
         :identifiedObjectName SS2))
   :faultDescription Particle in SF6)
```

Figure 7.10: PD defect diagnosis fact in FIPA-SL syntax translated by the sensor network gateway

Table 7.3 shows a comparison between the transmitted bit-efficient packet generated from the on-sensor diagnosis against the FIPA-SL fact reconstructed by the sensor network gateway. Here, we see that the data payload size can be reduced by 93% using the bit-efficient active message protocol while employing agent-based data management and diagnosis on the sensor.

Table 7.3: Comparison of bit-efficient to FIPA-SL encoded `InterpretedData` facts, received and generated by the sensor network gateway

Encoding medium	Message size (bytes)	Size of SL message (%)
Active message	19	6.66
FIPA-SL	285	100

7.3.6 Discussion

The approach to partial discharge diagnostics shown in this case study is specifically tailored for low-power sensor network operation. Through this approach, this case study has demonstrated that it is possible to deploy diagnostic agents on-sensor for condition monitoring applications, moving towards the integration of all components of a condition monitoring system in a single package. This particular application has shown that a 93% reduction in transmitted data volume can be achieved over FIPA-protocols using a bit-efficient protocol with the sensor network gateway.

The cross-platform C4.5 classifier used here demonstrates that lightweight diagnostic techniques can be used for sensor network applications. This component can be readily reused with other applications using Quinlan's binary decision tree format.

Due to issues with integration between PD detector and sensor node, a microcontroller-based PD detector simulator was designed for end-to-end systems testing. This is reused in the following case study, which applies the agent-based diagnostic sensor methodology to an existing condition monitoring application which was previously implemented on a PC.

7.4 On-sensor defect identification for the COMMAS system

The final case study illustrates how the agent-based sensor network methodology can be applied to an existing condition monitoring system. The COMMAS system, introduced in Section 3.5.1, is a multi-agent diagnostic system designed for partial discharge diagnosis. It has been deployed in many configurations [184], and integrated as part of the COMMAS v2 system [62]. To date, the COMMAS system has been deployed for up to two plant items; this case study demonstrates that it is possible to deploy the data capture and defect identification components of COMMAS onto a wireless sensor, which leads to

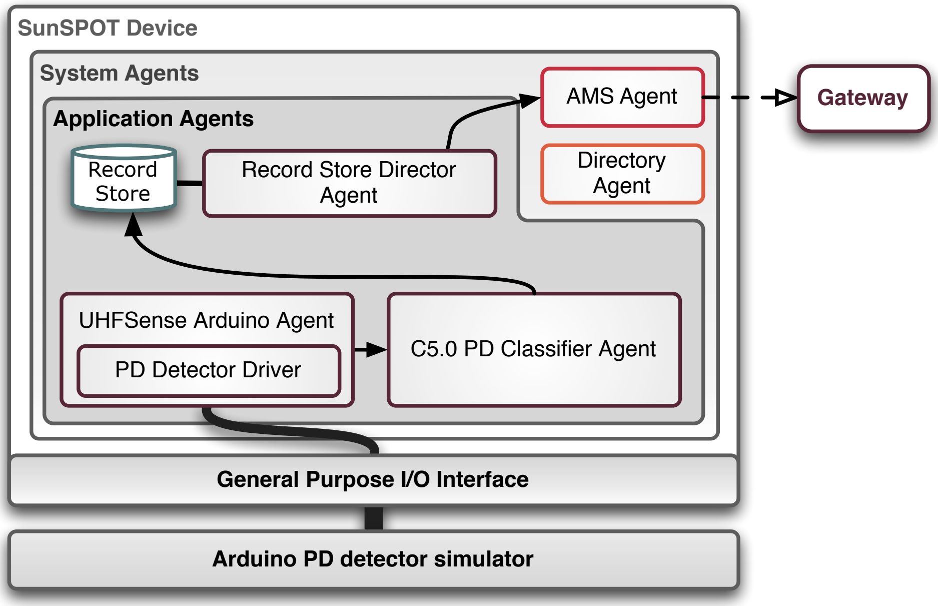


Figure 7.11: SubSense configuration for sensor-network enhanced COMMAS system

increased scalability, and in turn enables the COMMAS system to monitor a large number of plant items.

The sensor node configuration used for this case study re-uses the PD detector simulator used in the previous case study. To demonstrate the combination of simple measurement and interpreted data generation, a battery level monitoring agent was added to the system to monitor the sensor node. Figure 7.11 shows the SubSense agent architecture for the COMMAS application.

7.4.1 Application agents

The application consists of three agents, performing data capture, management and diagnosis. As with the previous examples, the record store director agent is used to monitor for new data and send it to the gateway for dissemination. The additional application agents are:

7.4.1 a) Battery Monitoring Agent

The battery monitoring agent is a general-purpose agent deployed to relay the battery level after a predefined period, in this case 10 minutes.

7.4.1 a) UHF Sense Test Agent

The UHF Sense test agent implements a subset of the COMMAS interpretation capabilities, receiving phase-resolved partial discharge data from the PD detector simulator with which performs on-sensor defect identification. Upon reception of data, the 3200 values representing 1-second of PD pulse data (50 cycles, 64 samples per phase) are used to generate a representative vector of features used for diagnosis. The agent achieves this with two behaviours:

FeatureVectorCalculatorBehaviour: Amended from the COMMAS system to run on the SunSPOT platform. This behaviour generates 101 features from a 1-second snapshot of PD data. This 'feature vector' is the input for each of the diagnostic behaviours within the COMMAS system; and,

C5FeatureVectorClassifierBehaviour: This behaviour is amended from the classifier developed for the COMMAS system, identifying PD defects using heuristics derived from test data using the C5.0 algorithm [188] which is an incremental enhancement to the C4.5 algorithm discussed in the previous section. This behaviour generates `InterpretedData` facts.

7.4.2 SubSense configuration

The SubSense configuration file used for this application is as shown in Figure 7.8. In this application, the record store director agent is configured to send data with two packet types—`10` and `30`, representing both raw sensor measurements and interpreted data. However, its is only battery level measurements that are transmitted in raw form: communication of PD information is limited to interpreted data to minimise data transmissions.

7.4.3 System architecture

Figure 7.13 shows the entire system architecture detailing the agents used for this application.

7.4.4 System operation

By combining data management and defect identification on-sensor, significant savings are made in terms of data transmission. Table 7.4 shows a comparison of the raw data sizes from the original 1-second 3200-byte PD pulse data

```
subsense.basestation.address=0014.4F01.0000.53D3:50

# General purpose agents
subsense.agent.org.poweragents.sunspot.agent.BatteryMonitoringAgent = Battery Monitor Agent

# COMMAS agents
subsense.agent.org.poweragents.sunspot.uhfsense.UHFSenseArduinoAgent = UHF Sense Test Agent
subsense.agent.org.poweragents.sunspot.agent.RecordStoreDirectorAgent =
        Archiver,InterpretedData,30,BatteryLevelStore,10
```

Figure 7.12: SubSense configuration file for a partial discharge diagnostic application

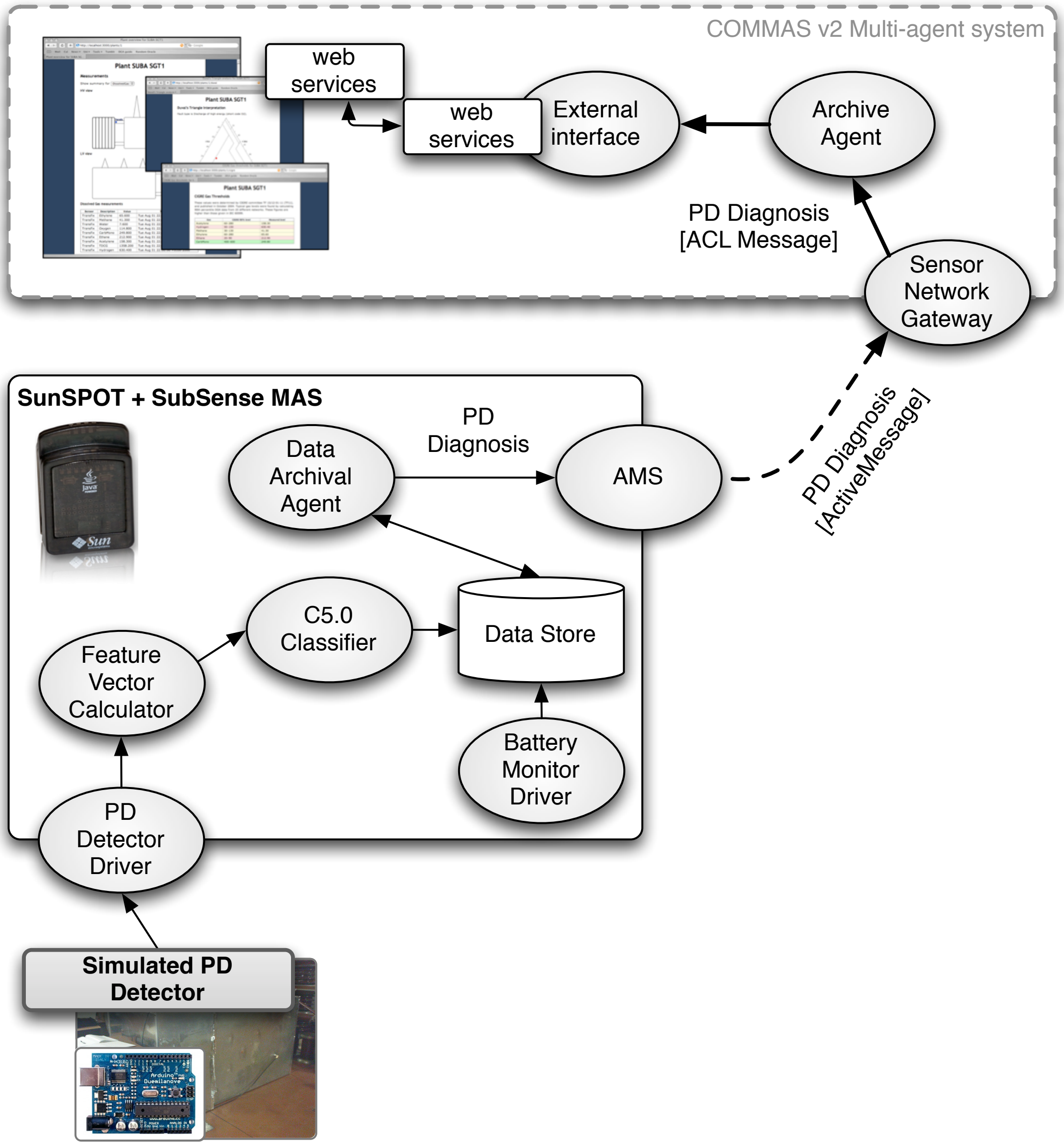


Figure 7.13: SubSense agent, communication path, and system architecture for the enhanced sensor-network enabled COMMAS application

Table 7.4: Comparison of bit-efficient to FIPA-SL encoded `InterpretedData` facts, received and generated by the sensor network gateway

Encoding medium	Data size (bytes)
Phase-resolved sample data	3200
Feature vector	808
Interpreted Data Active message	19
FIPA-SL	285

to the received `InterpretedData` fact generated from the bit-efficient message received by the sensor network gateway. Here, the PD defect is identified immediately, and transmitted in only 19 bytes. The transmitted interpreted data packet is less than 0.6% of the size of the raw 3200-byte phase resolved data; this translates to a 3 orders of magnitude reduction in communications load, decreasing energy consumption and enabling more diagnostic sensors to be deployed within the same network.

7.4.5 Discussion

The final case study has re-implemented a portion of the COMMAS condition monitoring system on a wireless sensor node. By pushing COMMAS data monitoring and interpretation agents down to the sensor level, techniques that could previously only be run on-line connected to a PC running on-site can now be executed on-sensor, capable of diagnosing partial discharge pulses immediately as soon as they are captured. This also reduces the communications burden placed on the monitoring network, as an identified PD can be transmitted using under 0.6% of the bandwidth of raw phase-resolved PD data.

This case study also demonstrates sensor node battery levels monitored along side partial discharges. This highlights the versatility of the multi-agent approach to building diagnostic sensor network applications; SubSense systems cannot only be expanded with the addition of diagnostic behaviours, but can also execute multiple diagnostic applications in parallel on a single sensor node.

7.5 Summary

In this chapter, three case studies have been presented to illustrate different scenarios in which diagnostic condition monitoring applications using wireless sensor may be employed: the first being a laboratory-based temperature and vibration monitoring system; the second being a laboratory-based partial discharge system employing the frequency-based diagnostic technique presented in Chapter 5, and; the third, an integrated deployment of monitoring and interpretation agents from the COMMAS diagnostic system, using a simulated partial discharge detector.

The second and third case studies represent the agent-based software components of an end-to-end wireless condition monitoring system, albeit without the on-line, physical partial discharge detectors. In future, the simulated PD detectors could be replaced with on-line devices—this falls under 'further work' in this area, discussed in the next chapter.

The case studies have demonstrated that condition monitoring systems can be extended down to the sensor level. Low power condition monitoring techniques can be used for substation condition monitoring applications, and multi-agent systems for data capture and diagnosis are effective for wireless sensor network applications. Using the sensor network gateway allows sensor networks and FIPA-compliant agent systems to integrate together within a unified agent architecture. This approach provides a simplified approach to building diagnostic systems, suitable for the next generation of substation condition monitoring applications.

Chapter 8

Conclusions and further work

"A witty saying proves nothing."

- **Voltaire, 1694-1778**

8.1 Conclusions

Condition monitoring of substations can mitigate against unforeseen outages, increase plant lifetime, and help reduce operational costs to the utility. While the need for more condition monitoring sensors is well documented in the literature, several key issues limit the wide-scale deployment of multi-sensor systems. Such systems will not only generate large volumes of data which inevitably must be interpreted, but they must also integrate with other condition monitoring systems so that plant health may be established in a uniform and timely manner.

Intelligent techniques have been widely applied to the issue of data volume and complexity to build online diagnostic systems. The application of multi-agent systems has been proven to simplify the integration of multiple techniques, as well as provide a robust platform that is suited to distributed operation. However, such systems have only been applied to single assets, and have not addressed the issue of a communications infrastructure required to support new CM sensors added to a wide range of plant.

This thesis has demonstrated the use of wireless sensor network technology to provide a mechanism for wide-scale condition monitoring within substations, and the use of low-power sensors in tandem with existing monitoring schemes to increase overall monitoring coverage and knowledge of plant health. A review of condition monitoring, multi-agent systems and wireless sensor networks demonstrated the feasibility of creating agent-based wireless networks for substation monitoring applications, establishing the following fundamental requirements:

- The need for low-power condition monitoring sensors for wide-scale deployment which can operate as part of a wireless sensor network;
- A suitable software architecture to support data management and in-network processing; and,
- Effective integration with existing condition monitoring systems.

To demonstrate that existing monitoring techniques may be adapted and reused in a low-power, resource constrained setting, the fundamentals of the RF technique were adapted so that PD detection may be carried out within the constraints of a wireless sensor network. A partial discharge diagnostic technique using a frequency-based approach was developed as a low-power analogue to complement existing monitoring systems. This technique captures the spectral energies within three preset frequency bands, which have been shown to contain defect-specific information that may be used for defect identification and classification. Through the creation of a physical detector, laboratory experimentation has shown that the technique can identify multiple defects by exploiting both defect-specific frequency components and the effects on the observed frequency spectra due to the geometric relationship between the partial discharge source and monitoring sensors. The key research outcomes established through taking this approach are:

- Spectral measurements of multiple concurrent defects fall into separate clusters, either due to the intrinsic frequency properties of the PD source, or due to the geometric effects on the observed frequency spectrum of a PD emission;

- The frequency-based technique can be used to identify multiple concurrent PD sources; and,

- Partial discharge detection and analysis is feasible in a wireless sensor network context, as demonstrated through the creation of an integrated low-power PD detector.

To interpret the measurement data, a novel visualisation technique was developed to allow deterministic data interpretation. By normalising the frequency components against the signal magnitude to create a set of proportional spectral energies, a ternary plot was used to visualise 3-dimensional frequency data on a 2-dimensional simplex. As well as providing a means to manually view the measurement data generated by the detector, the method demonstrated that the test data could be linearly separated, and therefore could be used to build an online classifier.

A rule-based classifier was constructed using the C4.5 algorithm and trained using a set of test data generated from SF_6 partial discharge test cells within a laboratory environment. This classifier was developed to be cross-platform, executable on both a PC and a wireless sensor node. A key advantage of on-sensor diagnosis is the reduction in volume of measurement data, which is a necessity for sensor network applications where the cost of wireless data transmission is orders of magnitude higher than on-board processing. Rather than transmit raw data back to a base station for analysis, statistical features or defect classifications can be transmitted, delivering the required information as a greatly reduced payload.

To allow this technique to be readily deployed onto a sensor node as a diagnostic software agent, the SubSense multi-agent toolkit for the SunSPOT platform was developed. The platform is in the prototype stage, although the case studies detailed in Chapter 7 have demonstrated that it is reusable and suitable for multiple condition monitoring applications, including new low-power approaches and the enhancement of existing systems using sensor networks.

Wireless sensor networks allow the rapid deployment of sensors with minimal infrastructure requirements. As well as being a platform for novel low-power sensors, they must integrate with existing systems to maximise the benefit to monitoring engineers. Following an agent-based approach, a wireless sensor network gateway agent was created to integrate wireless and wired systems, capable of translating between the bit-efficient messaging required within sensor networks and the rich, speech-act-based messaging suite used within FIPA multi-agent systems.

The research presented in this thesis addresses a wide range of issues relating substation monitoring, encompassing measurement, detection, defect identification, diagnostics and integration across the entire spectrum of a condition monitoring system. This thesis has demonstrated that existing monitoring techniques can be adapted to a low-power approach, intelligent techniques can be deployed at the sensor level, and wireless sensor networks can be employed by industry to build enhanced, widely-deployable, next generation substation condition monitoring networks.

Throughout the execution of the research detailed in this thesis, several future research opportunities were established which are outside the scope of this document, but directly extend this body of work. These are proposed in the following section.

8.2 Further work

8.2.1 Extending the frequency-based RF technique

The frequency-based technique described in Chapter 5 was developed in a laboratory using an air-filled cuboid aluminium test tank. The geometry of this tank does not accurately model GIS or oil-filled transformers, therefore it is difficult to ascertain the exact effects on tank geometry on frequency spectra. To gain a better understanding of these effects, two separate approaches may be taken: firstly, spectral analysis of historical partial discharge measurements may be carried out, and secondly, further laboratory experimentation may be carried out using high-voltage test apparatus in a configuration that more closely resembles in-service plant. At the University of Strathclyde, a new testing apparatus which more closely resembles a physical transformer is nearing completion, and could be used in the future for further tests.

8.2.1 a) Frequency analysis of wide-band RF PD spectra

To-date, the frequency-based technique described in this thesis has only been applied to the experimental apparatus within a laboratory environment. To expand the technique, it should be applied to archived wide-band data from in-service transformers and GIS. The benefits of this are twofold: firstly, by analysing real defects over a long period, it may be possible to confirm that the frequency spectra of a defect emission does not change over time [53]. Secondly, by performing frequency analysis on measurements containing previously known defects, it will be possible to develop the frequency-based method and visualisation technique further; if several different same-type defects are seen to have similar frequency fingerprints, then it strengthens the argument for automated frequency-based analysis of PDs. It may also be beneficial to simulate a larger number of frequency bands to establish whether this provides any optimisation over the existing technique.

8.2.1 a) Mixture-model analysis of frequency data

The C4.5 classification technique works by linear separation of the result space, by generating a decision tree that is both readable and deterministic. While this rule-induction technique is well suited to regular, linearly separable data, it has a tendency to over-fit the training set and does not make any accommodation for noise.

A mixture model, which is a probabilistic model using a mixture distribution, may be applied in place of the C4.5 classifier, which would split the observations according to their proportions to create a number of distributions representing each cluster. Each data point would then have a membership function attributing it to each distribution, assigning a probability of a correct classification. The Dirichlet distribution may be well suited to this application as it is suited to proportional data. By training a Dirichlet mixture model from a large training set, a more accurate analysis of PD pulse frequency components may be achieved.

8.2.2 Implementing the UHF method on a sensor node

While the IEC standard recommends a sampling frequency upwards of 9kHz, the UHF method has been successfully implemented at a sampling rate of 3200Hz [26]. In view of this, it may be possible to implement the UHF phase-resolved technique on a wireless sensor node, by integrating low-power PD detection hardware with suitable data processing and analysis techniques. The physical hardware required to detect PD magnitude is similar to that of the frequency-based detector described in Chapter 5, although several changes to the original design are required to accommodate the change in approach.

If successfully implemented, the data-driven and knowledge-based diagnostic techniques previously applied to UHF PD data (described in Section 3.5.1) may be adapted and reused on a sensor node, and the detector could directly drive the diagnostic application described in the third case study in the previous chapter. The benefits of an integrated detector of this type match those of the frequency-based detector, and by applying a parallel technique to the same underlying technology, the two approaches can be compared and possibly integrated in the future leading to a more robust and accurate diagnostic method. There are, however, several issues that have been identified that must be addressed for this approach to be feasible.

8.2.2 b) Phase resolution

In both the IEC and UHF phase resolved plot methods, the phase angle of the electrical phase under observation is required so that an absolute reference may be used with which to perform analysis. The current breed of PD detectors use a tethered zero-crossing detector to do this, which consists of a detection unit connected to a VT connected to an energised conductor. For wireless sensor network applications, this poses several issues. If, for instance, a transformer is to be instrumented with multiple sensors which each require synchronisation, multiple wired connections to a VT would be required, which is impractical because of safety regulations surrounding high-voltage plant. There are a number of possible solutions which address this issue, however these are theoretical and their efficacy is still an open research question.

Firstly, a wireless zero crossing detector could detect the zero crossing point of the connected phase without a VT connection. Such a device, in its simplest form, would consist of a plate placed within an electric field. The variations in the electric field would induce an alternating voltage on the plate which could be conditioned before sampling by the sensor node. Using this approach, such a device would ideally be integrated with each sensor node. This approach may also be used as an application-specific approach to sensor node time synchronisation which is one of the fundamental issues facing sensor network deployments, as discussed in Chapter 4.

Alternatively, it may be possible to diagnose phase resolved plots without an absolute phase reference. This approach may be implementable purely in software, removing the need for an external zero-crossing detector. However it is currently a theoretical approach which has not yet been addressed in the literature.

8.2.2 b) Detector triggering

The detector described in Chapter 5 uses an external trigger which is activated when PD activity exceeds a preset magnitude. This approach cannot be used for the phase-resolved method, as the UHF phase-resolved method requires that sampling must be carried out over a 1-second period, starting from the zero-crossing point. In a tethered wideband system where energy is abundant, sampling can be carried out continuously, and periods of interest can be analysed as necessary. Under the energy constraints of a wireless sensing application, it is infeasible for a detector to record and analyse UHF data in this way as the energy requirements will be too large. An optimised method of detection must be sought, so that sensing and diagnosis can maximised within the energy constraints of the device.

If a piece of plant is undergoing continuous partial discharge, the optimal sampling period may be solely chosen based upon the energy resources available to the node. However, other factors may make partial discharge activity irregular such as varying loads and changes in ambient temperature. A physical model could be created that incorporates available energy with external factors, which could be used to plan when to sleep and when to wake, sample and diagnose. This model may be as simple as a set of heuristics, although further research is required to establish the complexity of this interesting optimisation problem.

8.2.3 A domain-specific approach to sensor network synchronisation

Discussed previously in Section 4.3.2, sychronisation of sensor nodes is a key requirement for widescale sensor network deployments. Domain-specific approaches are recommended in [126], however there has not yet been a generic domain-specific approach to power system wireless sensor networks.

One approach may be to synchronise sensor clocks to the fundamental power frequency. This method is used within commercial clocks, where standard operational variations in the power frequency are compensated over a 24 hour period so that the time may be kept accurate to the nearest second over a 24 hours period. For partial discharge detection which uses the time-of-flight information for defect location, a zero crossing detector is used to synchronise detected pulses to a common phase reference [44]. Such detectors use a wired connection attached to the phase under observation via a voltage transformer (VT). Theoretically, a wireless zero crossing detector could be implemented that could detect the zero crossing point of the connected phase without a VT connection by detecting changes in the ambient electric field. If such a system could be implemented requiring a nominal amount of energy to operate, it may provide a domain-specific time synchronisation method that was both unique to the power engineering domain and sufficiently effective to ensure that all substation sensor nodes were autonomously able to synchronise their clocks to the nearest second on demand.

To compensate against local power frequency deviations, local phasor measurement unit (PMU) data may be used to compensate against clock skew when sensor data arrives at the edge of the wired network. This concept is purely theoretical at this point, but may provide fruitful for wireless power system sensing applications in the future.

8.2.4 Integrating power and sensor network ontologies

In Section 4.5, the issues of integrating sensor network and condition monitoring ontologies has been explored. CIM has already been recommended as a basis for the power engineering upper ontology [70], and a suitable sensor network ontology must also be identified, such as IEEE 1451 or SensorML.

Once this has been achieved, mapping between the two ontologies can be carried out to allow data exchange between the two domains. However, ontology mapping can be a non-trivial task, as discussed in Section 3.4.2. In [162], mapping between two power engineering ontologies is approached, where the authors find that mappings must be implemented by hand, which results in a time-consuming and therefore expensive process. Mapping between two mature and standardised models such as CIM and SensorML, for instance, would only need to be carried out once, with amendments only made when new versions of either standard were released. The initial mapping process would be time-consuming, however once this has been achieved the benefits to multi-agent sensing applications are clear. Alternatively, both ontologies could individually be integrated with SUMO, the suggested upper merged ontology [92]. In the long run, this may be more beneficial as SUMO aims to be the topmost ontology in an ontology hierarchy.

8.2.5 Network model agent

If power and sensor network ontologies can be integrated, models for both sensor networks and electrical networks can be combined. If this can be achieved, it may be possible to use these models to improve the static message translation method used within the sensor network gateway. A 'network model agent' is proposed which can use existing domain knowledge combined with reasoning capabilities to answer arbitrary queries on sensor and power networks.

Such capabilities could be integrated directly with the sensor network gateway, however, by deploying it as a separate service, it could not only provide contextual information to the sensor network gateway but also supply domain knowledge as a generic service to other agents. Under this scheme, application agents can be programmed to search for plant and network information periodically, removing the need to supply them explicitly with prerequisite site-specific knowledge. This would be beneficial to all agent-based substation diagnostic applications as it would remove the need for agents to be manually provided with domain information at startup. Properly configured, they would search out domain knowledge which would be provided by the network model agent.

As we have seen in Section 3.4.4, for agents to be fully flexible they should ideally be able to comprehend any message that they receive. As an alternative to static message parsing, the use of a semantic reasoner such as the JADE Semantic Framework [105] could allow agents to evaluate arbitrary logic statements through a complete understanding of the FIPA-ACL and FIPA-SL semantics. Semantic reasoning capabilities can potentially add even more benefit to a network model agent, by incorporating a theorem prover, or backwards-chaining reasoner, combined with models of both the the electrical and sensor networks described previously.

Theorem provers operate by solving proofs of logic statements containing well-formed formulas that comply with some previously known world model. A suitably capable theorem prover is able to answer arbitrary logic questions against its knowledge base, so long as its models are valid and it can evaluate each predicate appropriately. It is proposed that the network model agent embodies such a capability, combining a theorem prover with models of the electrical and sensor networks, with appropriate links between models. This is illustrated in Figure 8.1.

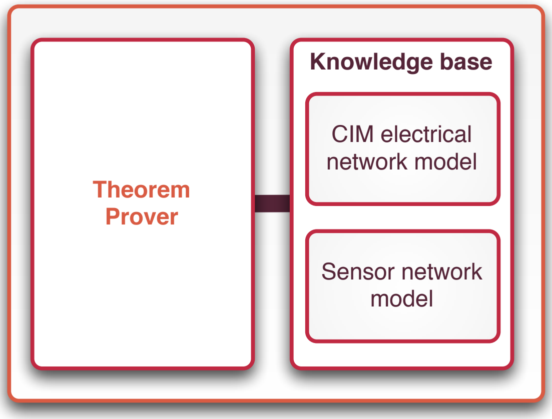


Figure 8.1: An overview of the proposed 'Network model agent' components

To develop this idea further, consider the following basic use case. When using the IEEE 802.15.4 communication protocol, each message includes a 64-bit IEEE 802 MAC address of the sender, unique for each sensor node. Assuming each sensor node is monitoring either a single power system resource or an ambient measurement, the MAC address of a node may be used as an index key with which to query against a model of the sensor network to establish the particular plant—or the substation, in the case of ambient measurements—under observation. The Active Message type ID can also be used to retrieve the ontology concept from the network model agent pertaining to the fact to be created. With this information, it will be possible to retrieve all necessary information to construct an agent fact from the message contents. Table 8.1 shows the contents of a mock partial discharge anomaly message (in human readable form rather than the bit-efficient form used in the previous chapters).

For the sensor network gateway to construct an ontology fact from the ac-

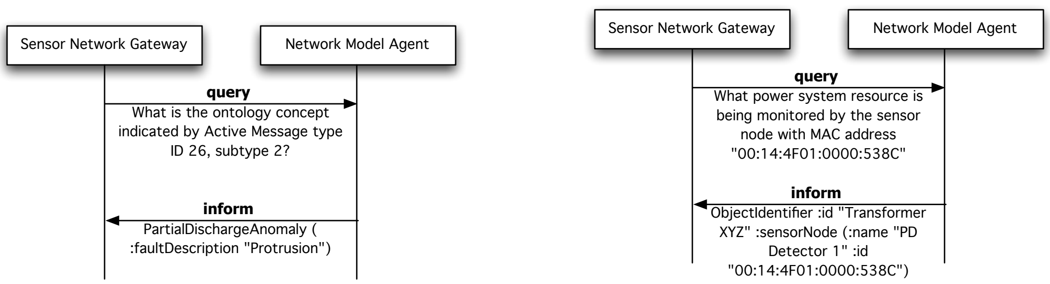
Table 8.1: An example Active Message for a partial discharge anomaly, in human readable form

mac address	typeid	payload
00:14:4F01:0000:538C	30	2 1257264163

tive message, it must query the network model agent to establish: 1) the ontology fact type, and 2) which plant item the sensor node which sent the message is attached to. A diagram illustrating these interactions can be seen in Figure 8.2. These have been simplified for readability, where the query has been represented in English rather than FIPA-SL. The process behind these interactions are as follows:

- The sensor network gateway receives a message with active message type ID 30. It does not recognise this type, so it queries the network model agent to establish which concept it relates to. The network model agent retrieves this information from its knowledge base, and returns it to the gateway.

- The sensor network gateway asks for plant information pertaining to the MAC address in the received message. The theorem prover then constructs a number of reasoning steps to identify the *PowerSystemResource* matching the address, iterating through them until a solution is found. In this case, the theorem prover returns a solution to the query, detailing the plant under observation. This solution is then sent in reply to the sensor network gateway, so it may combine the response with the sensor reading to contextualise the data.



(a) Message ID query interaction

(b) PowerSystemResource query interaction

Figure 8.2: Interaction diagrams between SNG and NMA agents for sensor network data integration

```
(PartialDischargeAnomaly (
   :faultDescription "Protrusion"
   :sourceMeasurementTime "1257264163"
   :powerSystemResource (
      :objectIdentifier (
         :id "Transformer XYZ"
         :sensorNode (
            :name "PD detector 1"
            :id "00:14:4F01:0000:538C"
         )
      )
   )
)
```

Figure 8.3: A PD diagnosis message, constructed as an ontology fact using plant data from the NMA

Upon receiving this information, the sensor network gateway can then integrate the sensor network data with the information received from the network model agent. This is illustrated in Figure 8.3, where we can see the fully constructed *PartialDischargeAnomaly* fact, represented in three colours to indicate the information source:

- Fault information retrieved from the original message
- Concept information retrieved in the first interaction (Figure 8.2(a))
- *PowerSystemResource* information retrieved in the second interaction (Figure 8.2(b))

Integrating existing data models and combining it with reasoning capabilities is a non-trivial task, however it would greatly benefit sensor network operation and if the reasoning capabilities extended to the entire FIPA-ACL and FIPA-SL semantics, it could act as a general-purpose knowledge repository, directly benefiting other agent-based applications within the power domain.

8.2.6 Industrial field trials

To date, the low-power partial discharge detector has been tested solely in a laboratory environment. With appropriate changes to the detector to make it field ready, it may be tested in an online setting as part of an industrial field trial. Such a test would require UHF transducers to be previously installed on the plant under inspection; based upon the availability of installed UHF sensors, this would most likely be on a GIS unit.

To fully test the partial discharge detector, it is recommended that a field deployment also includes wideband capturing of UHF signals. Operating alongside the low-power detector, an oscilloscope would capture wide-band data which could be analysed off-line and compared to the online measurements captured by the PD detector. This approach would allow the accuracy and susceptibility to noise of the low-power detector to be determined through a relative performance comparison of the two systems.

Referring back to Section 2.2.1, one of the key requirements for condition monitoring systems is *reliability*. In Section 4.6, we have seen that this has been addressed in part for wireless substation communications, however, many other factors contribute to reliability in this area, including advances in sensor network technology, advances in energy harvesting, and advances in diagnostics. Reliable wireless condition monitoring systems can only practically be achieved through the combination of these components in field trials to ensure that the technology is robust enough to operate in the potentially environmentally and electromagnetically harsh environment of the substation.

Appendix A

Supplementary PD detector circuit diagrams

This appendix contains supplementary circuit diagrams for the frequency-based partial discharge detector discussed in Chapter 5. Figure A.1 shows the single-channel circuit diagram, and Figure A.2 shows a diagrammatic mock-up of the full 3-channel detector.

All circuit design, layout, prototyping and fabrication was carried out by the author for the purposes of the research detailed in this thesis.

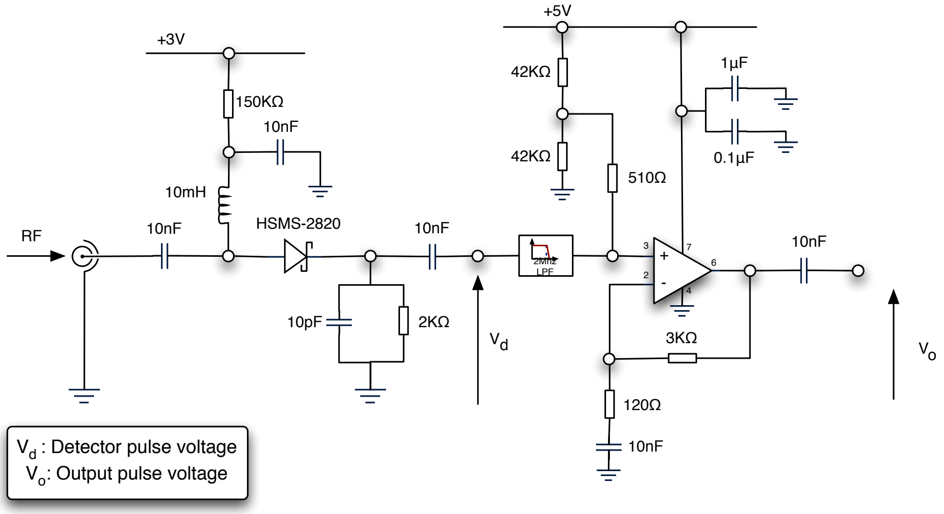


Figure A.1: Single channel circuit diagram for the frequency-based partial discharge detector

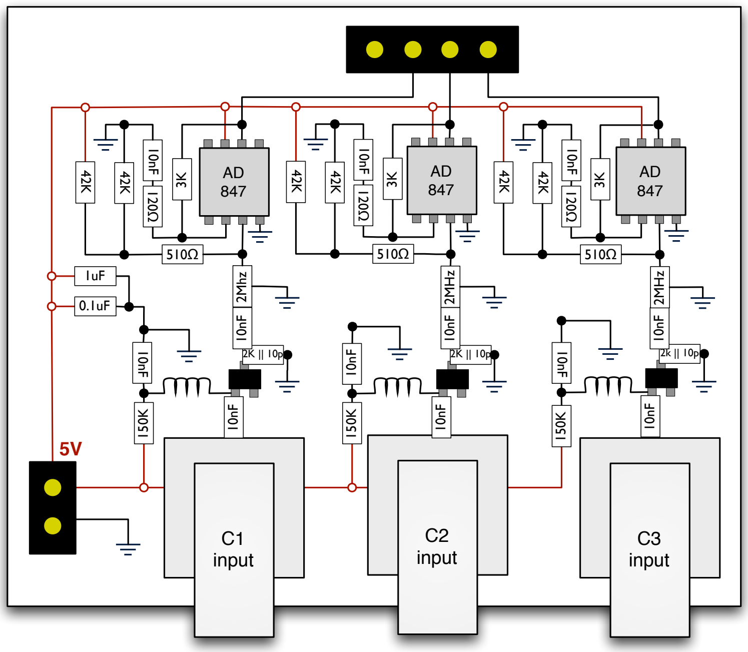


Figure A.2: Layout of the frequency-based partial discharge detector

Appendix B

SubSense toolkit overview and user manual

B.1 Introduction

This appendix serves as a system specification and user manual for the SubSense multi-agent toolkit. This toolkit, designed for the SunSPOT wireless sensor platform, was used as the basis for the case studies described in Chapter 7. After an overview of the toolkit architecture, its components will be described in full, including use cases and code examples that demonstrate how the toolkit can be used for other applications in the future. The appendix also includes an overview of the cross-platform C4.5 classifier written for the SubSense platform.

The following three documents should be used as reference when using this guide, as they cover the basics of the SunSPOT platform, the theory of its operation, and a developer's guide:

- Sun Microsystems, "SunSPOT Owners Manual - Red Release 5.0." Available from `https://www.sunspotworld.com/docs`, June 2009
- Sun Microsystems, "SunSPOT Theory of Operation - Red Release 5.0." Available from `http://www.sunspotworld.com/docs`, June 2009
- Sun Microsystems, "Sun Small Programmable Object Technology (Sun SPOT) Developers Guide." Available from `http://www.sunspotworld.com/docs`, May 2009

In addition to these documents, community support for the SunSPOT platform is available in the SunSPOT discussion forum at:

- `http://www.sunspotworld.com/forums`

It should be noted that in the code examples within this guide, some lines of code are omitted from the original source for brevity, such as package imports and header comments. Any classes or methods referred to in the software examples or UML class diagrams which are not discussed in this guide are part of the SunSPOT software API: refer to the documents listed above for more information on these.

B.2 System overview

The SubSense toolkit is a multi-agent system designed for the SunSPOT wireless sensor platform. It was designed specifically for use in diagnostic applications, however it is sufficiently generic to be used as a general-purpose platform to prototype wireless sensor applications using the multi-agent software methodology.

For system designers familiar with the JADE platform [76], the architecture and programming conventions will be familiar, however it is targeted for resource-constrained wireless sensor networks so there are some components which will be unfamiliar to experienced agent developers. The toolkit is designed to be run on individual nodes running sensing and diagnostic applications, communicating data between each other, and ultimately a sensor network gateway.

The SubSense toolkit is not directly compatible with either JADE or JADE-LEAP agents (for the reasons discussed in Section 4.5), however applications built using the SubSense toolkit can be easily integrated with FIPA-compliant multi-agent systems using the SubSense-to-FIPA Sensor Network Gateway, which is discussed in Section 6.4.

B.3 System architecture

The system architecture, shown in Figure B.1, is split into four main functional groups:

- Agent Platform
- Agents
- Behaviours
- Functional Components

The toolkit provides the entire system-level functionality, as well as a number of prototype and off-the-shelf agent and behaviours that can either be used directly or built upon. The 'functional components' are a catch-all term for sensors, SunSPOT API features and other modules such as diagnostic capabilities, which are wrapped as behaviours before being combined within an agent. Each of the four toolkit subsystems will now be discussed in turn.

B.3.1 SubSense agent platform

At the top of the component hierarchy (shown in Figure B.1) is the `SubSenseApp` class. This class is described as a 'Midlet' which is a Java Mobile environ-

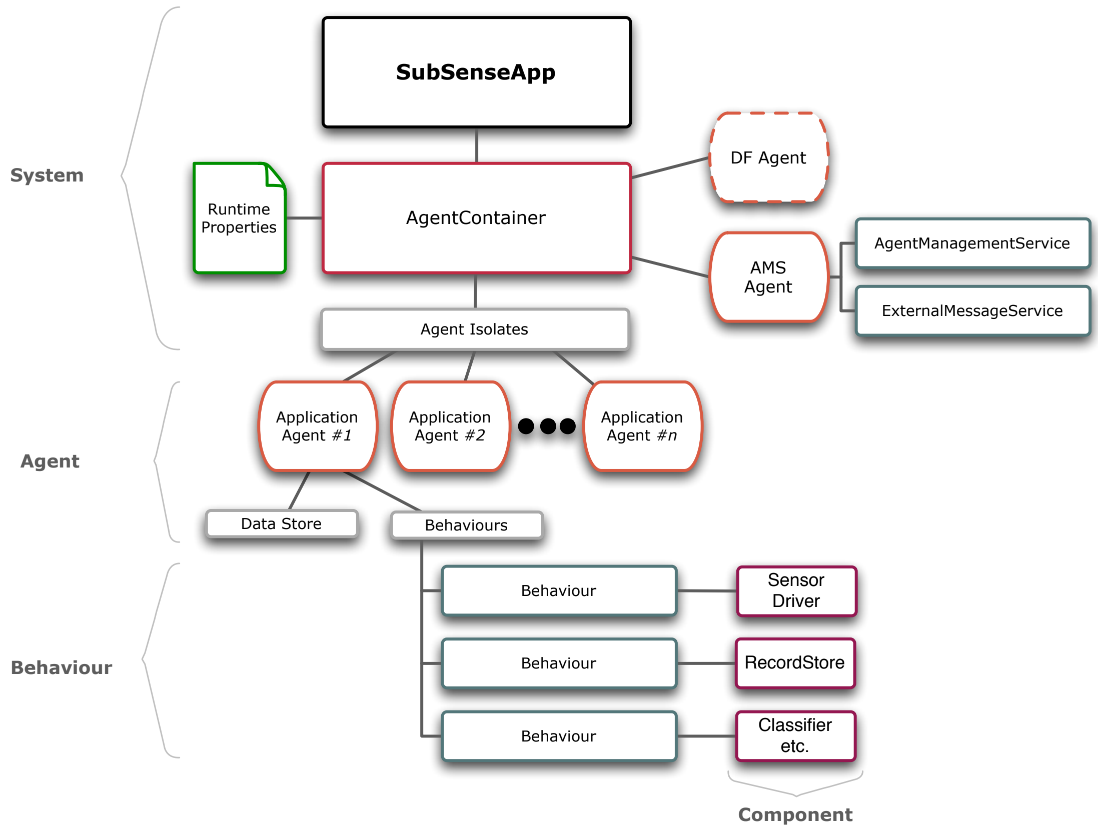


Figure B.1: SubSense architecture component overview ment application. In Java Mobile environments all applications must extend the MIDLet class, to to ensure they will run on compatible devices such as the SunSPOT. More information on MIDLets, can be found in the SunSPOT Owners Manual [189].

The SubSenseApp class has two core functions; firstly to load the runtime properties file 'subsense.properties' which contains agent and platform configuration details, and secondly, to instantiate the agent container within which the application agents reside. The UML diagram for the SubSenseApp and AgentContainer classes are shown in Figure B.2, and an example properties file is shown in Figure B.3.

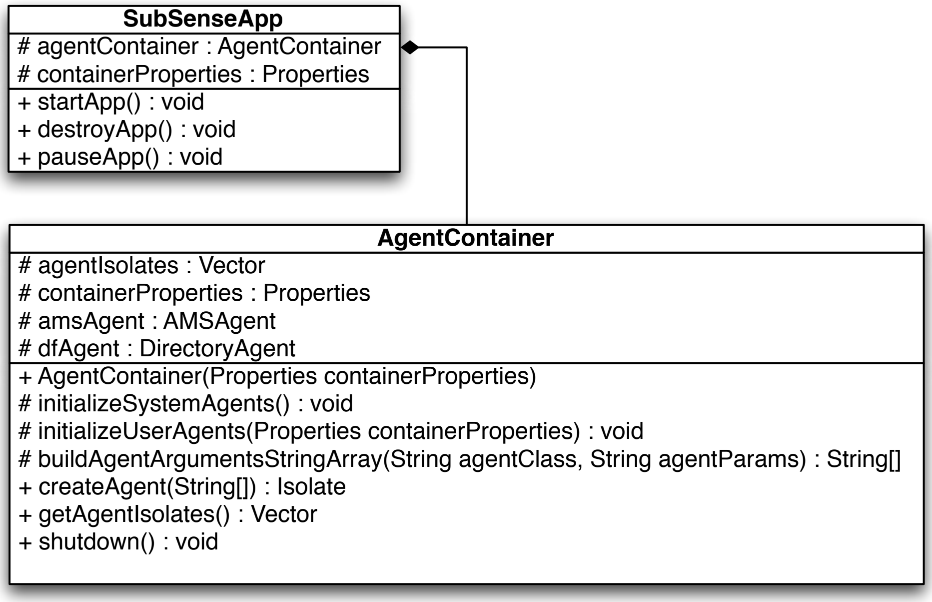


Figure B.2: SubSense system component UML diagram

B.3.1 a) Subsense.properties

The subsense.properties allows different agent configurations to be deployed at runtime, an example of which is shown in Figure B.3. In this properties file, the base station network address must be defined to allow sensor nodes to connect to the gateway. Individual application agents are defined by Java class name, followed by invocation arguments. This allows for agents to be deployed in various configurations depending on the application, specifying which agents to load at startup, and how they should be configured. To illustrate how this is used: during the development stages of this thesis this method simplified the switch between temperature monitoring and partial discharge monitoring applications by simply changing a few lines in the properties file, rather than rewriting or recompiling any source code. The settings supported in the properties file are:

subsense.basestation.address: A MAC address indicating the network address of the sensor network gateway (Ideally this setting should be discovered automatically: this feature may be added in the future); and,

subsense.agent.*: Settings starting with this format indicate runtime agent declarations, with the * representing a wild-card expression which can be used to load specific agents. The format for this is as follows:

```
subsense.agent.<agent class> =
<Agent Name>,[comma separated arguments]
```

So for instance, the string:

```
subsense.agent.mypackage.MyAgent = foo,Temperature
```

indicates an agent who's class is '`mypackage.MyAgent`', name is '`foo`' and first configuration argument is '`Temperature`'.

B.3.1 a) The agent container

The agent container handles agent startup and shutdown functionality, offering a programmatic method to instantiate and manage new agents. The structure of the agent container class is shown in Figure B.2. Upon instantiation, the agent container first loads the system agents—the Agent Management System agent and Directory Agent, described below—followed by application-specific agents. All agents on a particular node are stored within a single agent container, allowing them to be managed centrally.

Within the SubSense system, agents take advantage of the Java Isolate API [192], which provides a sandboxed environment in which individual agents can operate. Using this facility ensures that agent operation is truly autonomous; agents can only interact with each other through messaging or through shared data stores, and their run state can only be modified by either themselves or the agent container. The two exceptions to this are the system agents which live directly in the container, allowing them to programmatically interact with the container.

```
# define basestation address
subsense.basestation.address = 0014.4F01.0000.53D3

# instantiate temperature monitor
subsense.agent.org.poweragents.sunspot.agent.TemperatureMonitoringAgent
        = Temp Vibration Agent

# Load a record store director agent to send temperature
# and vibration data to the basestation as a measurement packet.
subsense.agent.org.poweragents.sunspot.agent.RecordStoreDirectorAgent
        = Archiver,TemperatureStore,10
```

Figure B.3: An example SubSense configuration file

The use of the Isolate API theoretically also allows agents to migrate between hosts, effectively implementing a mobile agent pattern. This feature has not been implemented, however, this ability could be used in the future for live deployment of new agents to remote nodes as the underlying SunSPOT platform natively supports it.

B.3.2 Agent model

A number of agent prototypes are provided within the toolkit. A UML diagram showing their class hierarchy is shown in Figure B.4. The agent class functionality can be summarised as follows:

Agent: the base class for each agent within SubSense. Contains basic methods common to all agents, supporting AMS registration and inter-agent communications using the SunSPOT inter-isolate messaging API. The `onStartup()` method must be implemented by all subclass agents, which is executed upon agent startup. Typically, agent behaviours are instantiated within this method, an example of which is shown in the following section;

ServiceAgent: contains additional methods to register an agent `ServiceDescription` with the Directory agent, which contains details of a particular service provided by the agent. Also opens a channel through which other agents can access the service. The Application agents should typically implement the `ServiceAgent` class to simplify creating services that can be accessed by other agents;

SystemAgent: an abstract prototype class for the AMS and Directory Agents;

AMSAgent: The Agent Management Service agent contains a table of `WhitePagesEntry` instances representing each of the agents. This allows agents to be identified from their name and also their instantiation arguments set at runtime. The AMS agent also contains two system behaviours, the `AgentManagementService` which listens to agent registration requests, and the `ExternalMessageService` which allows agents to communicate with other sensor nodes and the sensor network gateway; and,

DirectoryAgent: The Directory agent contains a table of agent `ServiceDescription` instances representing the services present on the local sensor node. Each agent extending the `ServiceAgent` prototype registers a `ServiceDescription` with the Directory agent, allowing other agents to discover the service.

B.3.2 b) Example agent implementation

An example of an agent extending the `ServiceAgent` prototype is shown in Listing B.1. In this example, the `RecordStoreDirector` behaviour uses multiple instances of the `ArchiveBehaviour` (discussed later in Section B.3.3) to listen to new records being persisted within the SunSPOT flash memory.

In this example, we can see that the `onStartup()` method is implemented, defining a `ServiceDescription` before using agent arguments to create a number of instances of `ArchiveBehaviour`. Even though a single method is implemented, the underlying `Agent` and `ServiceAgent` functionality provides full messaging capabilities and automatic registration with the Agent Management System Agent. Here, we can see that the bare minimum that is required to develop a functional service agent is to implement the `onStartup()` method. This example demonstrates the simplicity of building an agent with the SubSense toolkit.

B.3.3 Agent behaviours

The agent behavioural model within SubSense is based loosely upon that of the JADE toolkit, where agents may possess an arbitrary number and combination of behaviours depending on their required functionality. These behaviours are combined by the agent designer to fulfil the design requirements of each particular agent. Within JADE, each agent is modelled as a single execution thread which has certain limitations; namely that behaviours cannot run concurrently, and it is possible for an agent to become 'deadlocked' if two behaviours interact with each other unexpectedly. Within the SubSense system, each behaviour is modelled as a single thread. This allows greater control of behaviour execution, and allows the use of native power management facilities where the SunSPOT automatically enters a shallow sleep state when all agent behaviours are in an idle state.

The following list of prototype behaviours are provided within SubSense to allow rapid prototyping of applications, illustrated in Figure B.5:

Behaviour: this is the basic prototype for all behaviour classes, linking the

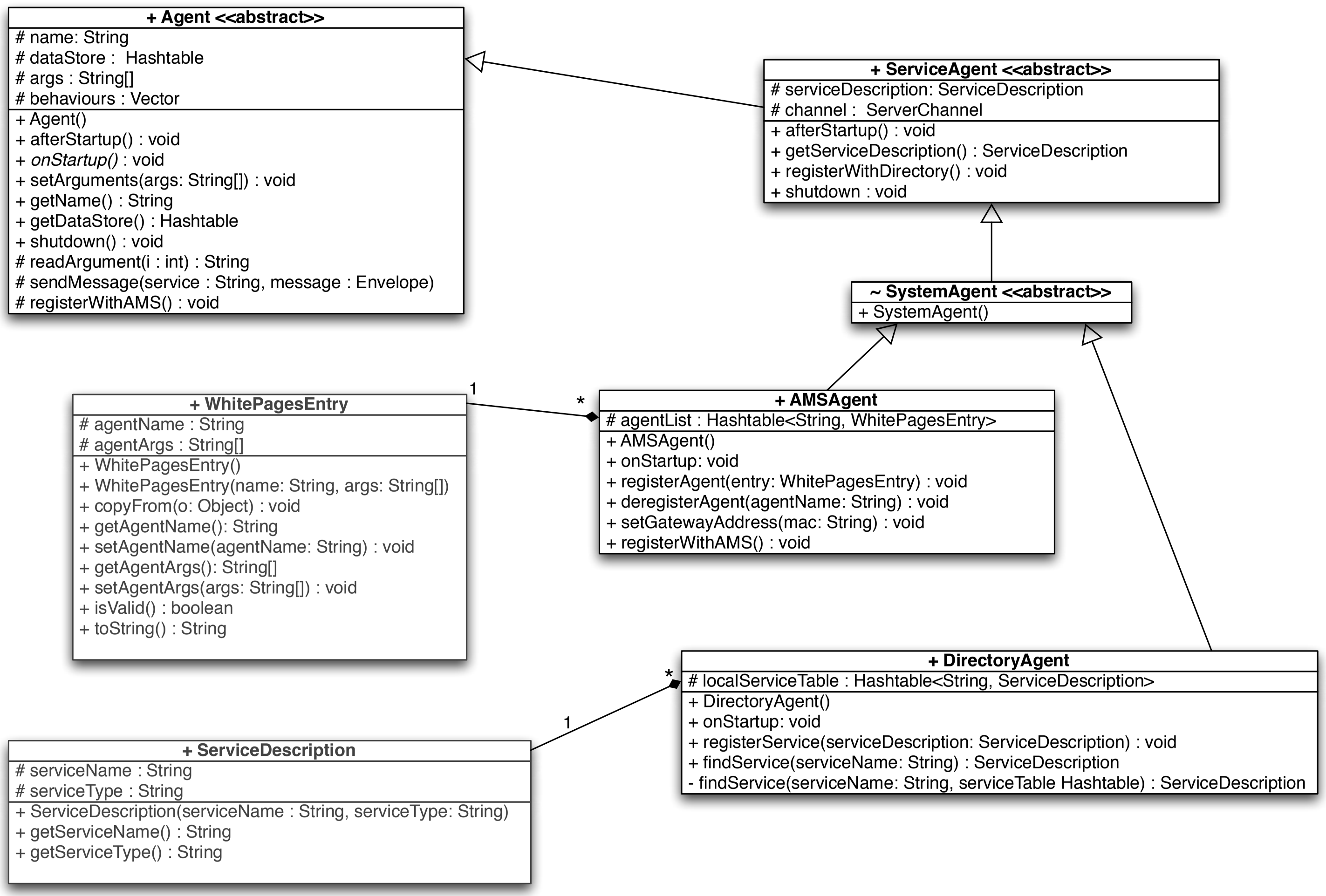


Figure B.4: SubSense agent hierarchy UML diagram

```java
public class RecordStoreDirectorAgent extends ServiceAgent {

  /**
   * Constructs a new RecordStore Director Agent.
   *
   * @param args must be an array of minimum length 2, where
   * the first entry is the agent name, and remaining arguments
   * are the names of {@link DataStore DataStores} to archive.
   * @throws AgentException if at least one  agent datastore
   * name is not provided,  or if the archive behaviours cannot
   * be instantiated.
   */
  public void onStartup() throws AgentException {
    this.serviceDescription = new ServiceDescription(
        "RecordStoreDirector");

    //there should be at least 2 args.   {name, recordstore(s)..}
    if (args != null && args.length >= 2) {
      //loop through the remaining arguments and pick out the
         datastores to archive
      for (int i = 1; i < args.length; i++) {
        try {
          //add a new behaviour for each datastore.   this sends
             the data to the AMS for archival
          this.addBehaviour(new ArchiveBehaviour(args[i]));
        } catch (Exception e) {
          throw new AgentException(e.getMessage());
        }
      }
    } else {
      throw new AgentException("At least one datastore name
         required");
    }
  }
}
```

Listing B.1: RecordStoreDirectorAgent sample

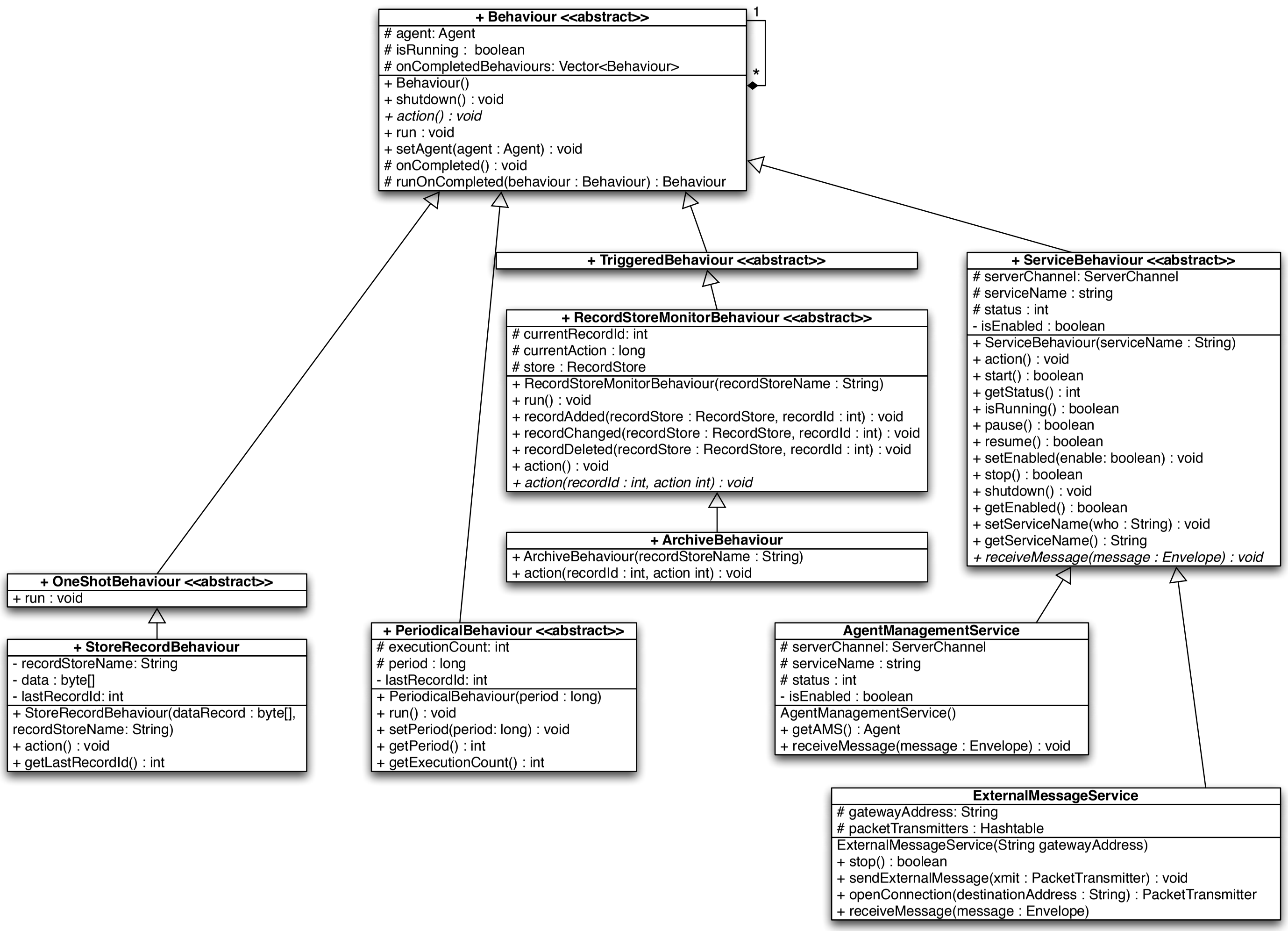


Figure B.5: Agent behaviour UML hierarchy agent model with the underlying Java thread model and providing method stubs required for behaviour implementations;

OneShotBehaviour: provides the basis for a one-shot, single execution behaviour which is discarded after execution;

PeriodicalBehaviour: allows periodic execution of an agent action based upon a period in milliseconds specified either upon instantiation or during operation. This behaviour takes advantage of the SunSPOT thread model, so that the SunSPOT device will enter into sleep mode during execution events if no other behaviours are active;

ServiceBehaviour: provides a basis for service-oriented behaviours, which exploit the native inter-agent messaging provided on the SunSPOT by the Isolate API; and,

TriggeredBehaviour: triggered by some external source, such as a record store or sensor.

Through the extension of the prototype classes described above, a number of general-purpose behaviours have been built, which can be added to application agents to quickly build up complex behaviour patterns. These are:

StoreRecordBehaviour: extends the one-shot behaviour, allowing agent data to be persisted into flash memory;

RecordStoreMonitorBehaviour: extends the triggered behaviour, providing blackboard-like subscription to persistent record stores;

ArchiveBehaviour: provides a simple mechanism for transmission of agent data to the base station;

ExternalMessageService: sends bit-efficient messages to other sensor nodes across the wireless link; and,

AgentManagementService: manages a table of all locally running agents, capable of pausing or shutting them down as required by the platform.

To illustrate how a behaviour is constructed, the `ArchiveBehaviour` mentioned previously is shown in Listing B.2. Here, we can see that the `action(int recordId, int action)` method is where all of the heavy lifting takes place. The `ArchiveBehaviour` class inherits from the `RecordStoreMonitorBehaviour` which, omitted for brevity, calls the action method whenever a change is made to the record store under observation. In this behaviour, whenever a record is added, indicated by the `RECORD_ADDED` indicator, a new `ArchiveDataEnvelope` is constructed and forwarded to the external message service before being sent to the basestation.

Through the combination of these behaviour classes, complex patterns can be created involving sensing, data storage and communication. By abstracting these functions at the behavioural level, application developers can mix-and-match abilities depending on the application.

B.3.4 Functional components

The SunSPOT API provides a number of components which can be leveraged for agent operation, including data storage and communication. Host agents typically interact with these through behaviours, which are discussed in the next section. The main components used within the toolkit are for data storage through the Record Store API, messaging through the inter-isolate messaging API, and physical sensors through the SunSPOT `IDriver` interface. This section includes an overview of each of these with brief code examples.

B.3.4 d) Record stores

The J2ME specification provides a persistent record storage system, using a publish-subscribe model. This is a basic implementation of 'Linda'-like blackboard communications, discussed in Section 4.5.1. Through this system, agents can easily share data with each other through named record stores, automatically receiving notification when new data of interest is available or has been modified.

In figure B.6, an example of three record stores is shown which store `Temperature`, `Vibration` and `Voltage` data. As the record stores implement a publish-subscribe model, it is possible for agents to subscribe to record stores and receive updates when new data is written to them. In this example, we can see that;

1. **Agent B** registers itself as a listener for the `Voltage` record store.
2. **Agent A** writes a new record to the store,
3. **Agent B** will automatically be informed of the new record.

```java
/**
 * Listens to records added to a particular record store, and
 * sends them to the  AMS agent to be sent to an external
 * destination for archival.  This behaviour is run each time
 * a new entry is added to a record store, at which point it
 * asks the AMS to send it to an archive.
 */
public class ArchiveBehaviour extends RecordStoreMonitorBehaviour
    {

  /**
   * Attempts to open the record store with the given name,
   * throwing an exception if it is not possible.
   *
   * @param recordStoreName is the name of the record store
   * to monitor.
   * @throws RecordStoreException if the record store cannot
   * be opened or is not found.
   */
  public ArchiveBehaviour(String recordStoreName) throws
      RecordStoreException {
    super(recordStoreName);
  }

  /*
   * @see org.poweragents.squawk.agent.behaviour.
       RecordStoreMonitorBehaviour#action(int, int)
   */
  public void action(int recordId, int action) {
    switch(action) {
      case RECORD_ADDED: {
        try {
          //forward the data to the ams agent for archival
          byte[] data = this.store.getRecord(recordId);
          Envelope env = new ArchiveDataEnvelope(data);
          this.agent.sendMessage(ExternalMessageService.
              EXTERNAL_MESSAGE_SERVICE_NAME, env);

          this.store.deleteRecord(recordId);
        } catch (Exception e) {
          System.out.println("Could not archive data: "+e);
        }

        break;
      }
    }
  }

  /**
   * Do nothing upon deletion.
   */
  public final void recordDeleted(RecordStore recordStore, int
      recordId) { }
}
```

Listing B.2: Archive Behaviour source code

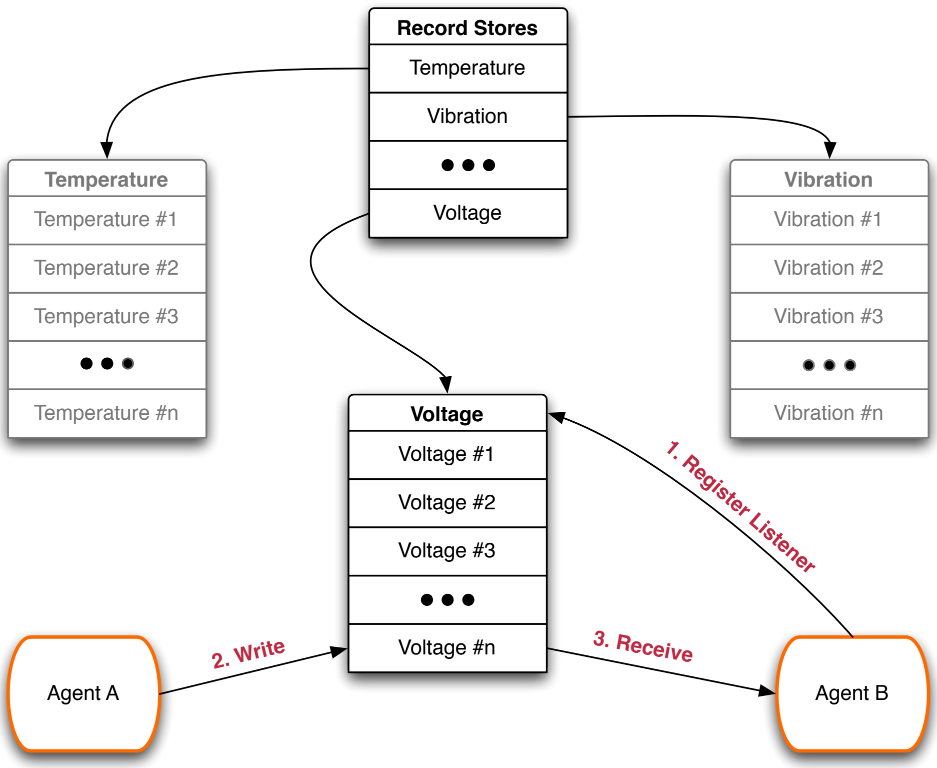


Figure B.6: Three SunSPOT Record Stores, with agent interaction between two agents and a Voltage measurement store

This is a useful method of sharing sensor data within a sensor node, as it has low overheads and ensures that only one copy of the data is stored per node.

B.3.4 d) Agent Messaging

Within SubSense, agent services can be accessed by other agents using messages, supported by the SunSPOT inter-isolate messaging API. To illustrate how the messaging works, consider the following code example, taken from the `Agent` class:

```
/**
 * Sends a message to the service with the given name.
 *
 * @param service is the name of the service to contact.
 * @param message is the message envelope.
 */
protected void sendMessage(String service, Envelope message)
    throws AgentServiceException {
  Channel serverChannel = null;
```

```
    try {
      serverChannel = Channel.lookup(service);
      serverChannel.send(message);
    } catch (IOException e) {
      throw new AgentServiceException("Cannot find service with
          name :"
          + service);
    } finally {
      if (serverChannel != null) serverChannel.close();
    }
  }
```

Listing B.3: `sendMessage` method from the Agent class

In this method, we can see that for an agent to send a message it must know the name of the service, and package up the message payload in an `Envelope` instance. If a valid service name—same as those indexed by the system Directory Agent–and `Envelope` instance is provided, the agent will connect to the service and the message will be successfully delivered. The `Envelope` class istelf is an abstract class so cannot be instantiated, however, the following subclasses are provided for use within the SunSPOT API:

ByteArrayEnvelope: Wrapping an array of bytes to be sent between agents;

ObjectEnvelope: Wrapping an arbitrary object implementing the `ICopiable` interface;

RequestEnvelope: Implementing a remote procedure call pattern: wrapping a command to be executed by the receiving agent; and,

ReplyEnvelope: A message representing the reply from a `RequestEnvelope`.

These classes can be used directly or extended for specific applications. The following code example in Listing B.4 shows an example of how the `ByteArrayEnvelope` can be sub-classed specifically for archived data transmission:

In this class, additional methods are provided to read the data and message types independently. This is one example of an `Envelope` implementation, however it is possible to create custom envelope classes for specific data types to communicate any type of data between running agents.

```java
public class ArchiveDataEnvelope extends ByteArrayEnvelope {

  /**
   * Constructs a new archive data envelope
   */
  public ArchiveDataEnvelope(ActiveMessage msg) {
    super(msg);
  }

  /**
   * Returns the data payload from the active message.
   */
  public byte[] getData() {
    return ((ActiveMessage) this.getContents()).getData();
  }

  /**
   * Returns the active message type ID.
   */
  public int getMessageType() {
    return ((ActiveMessage) this.getContents()).getMessageType();
  }
}
```

Listing B.4: An example of a custom `Envelope` class

B.3.4 d) Sensor Drivers

To simplify the process of sensing within the SubSense platform, an extensible driver model was created that could be re-used across applications. Figure B.7 shows a UML hierarchy containing three sensor drivers, for a thermometer, battery monitor and PD detector. Each of these drivers inherit from the `AbstractSampler` class, which is a prototype that allows listeners to be attached to specific sensors. This mechanism ensures that each time a sample is taken, all `ISampleListener` instances registered against a particular sensor are updated. To integrate this model with agent behaviours, two additional reactive behaviour prototypes were created specifically for use for sensing: `InterruptedSensorDataBehaviour` and `PolledSensorData-Behaviour`. These allow sensing behaviours to be triggered either based upon an external interrupt, for instance when a voltage threshold is crossed, or periodically after a specified time period.

Using this model, sensor drivers can be easily constructed. Extending the `AbstractSampler`, the `SimpleDriver` class is the basis for all drivers which take a single measurement per sample. Both the `ThermometerDriver` and `BatteryMonitorDriver` extend this class, integrating code to address specific physical devices. The `AbstractMultiChannelSampler` class is used for drivers which sample more than one channel concurrently. An example of this is the `PDDetectorDriver`, which samples three voltage values concurrently, representing the relative energy levels across three frequency bands of a partial discharge pulse.

In practice, the SubSense driver model extends the underlying SunSPOT API to allows physical drivers to be easily controlled by reactive agent behaviours.

B.4 Cross-platform C4.5 classifier

To enable PD classification to be achieved on the SubSense platform, it was necessary to build a suitable classifier which would run on the SunSPOT / J2ME platform. This classifier had to match the functionality of Quinlan's original C4.5 classifier which was written in C [185]—not compatible with the SunSPOT architecture. Rather than build a platform-specific implementation for the SunSPOT, a cross-platform classifier was built to run on both the SunSPOT mobile Java platform and the standard Java desktop / server environment. This is beneficial as it allows the same code to be reused for multiple applications, and allows the embedded implementation to easily be validated against the desktop version.

An abridged version of the C4.5 loader and classifier classes is shown in Figure B.8. This contains the core classes, with peripheral classes omitted for brevity. For the implementation to be generic, it has to be completely application-agnostic, loading binary decision trees at runtime. To achieve this in a cross-platform manner, two separate binary file loaders were written. These load binary files generated by Quinlan's decision tree generator, constructing a decision tree which is traversed to obtain a classification. This method allows for decision trees to be built and tested on a PC environment before being deployed onto a sensor node.

B.4.1 SubSense implementation overview

The C4.5 classifier package was wrapped as two agent behaviours to enable it to be used within SubSense:

LoadDecisionTreeBehaviour: This one-shot behaviour wraps the `Record-StoreBinaryTreeLoader` behaviour to load the binary decision tree

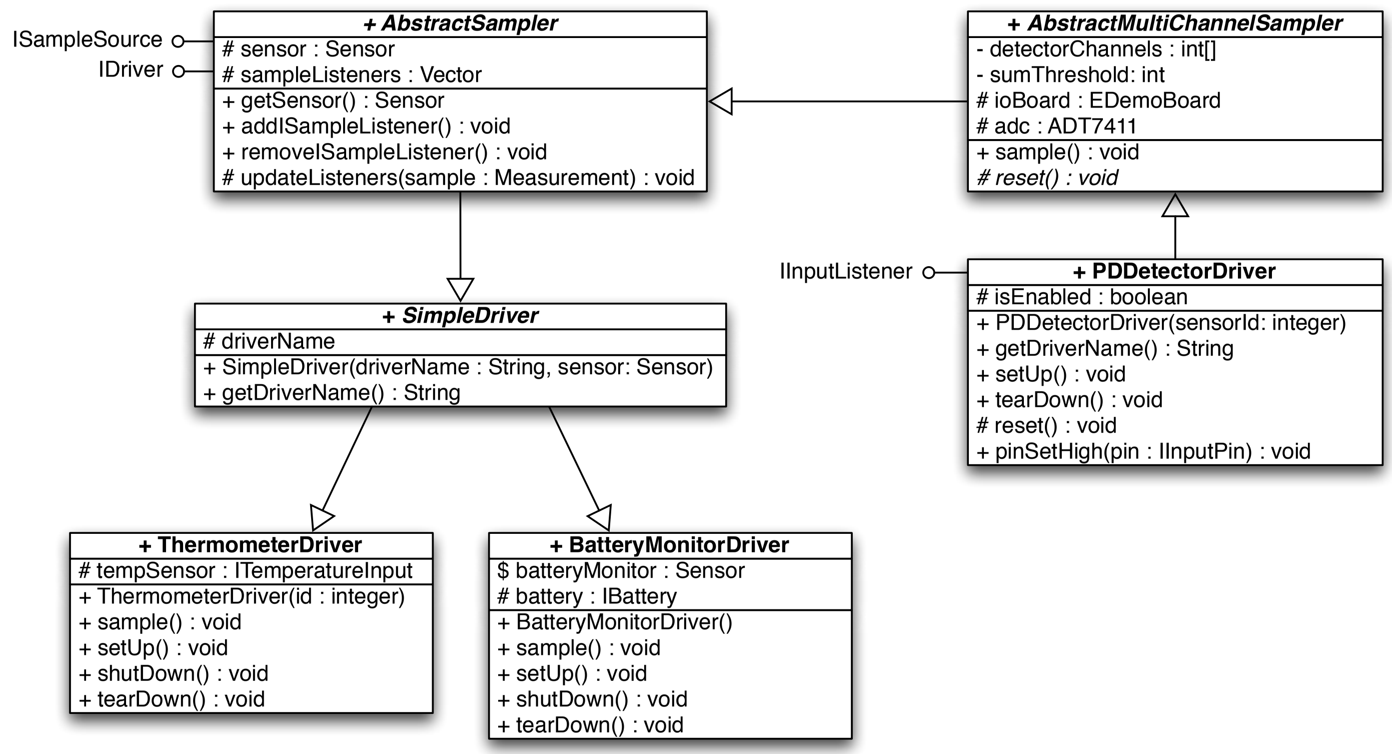


Figure B.7: UML hierarchy of a partial discharge detector driver

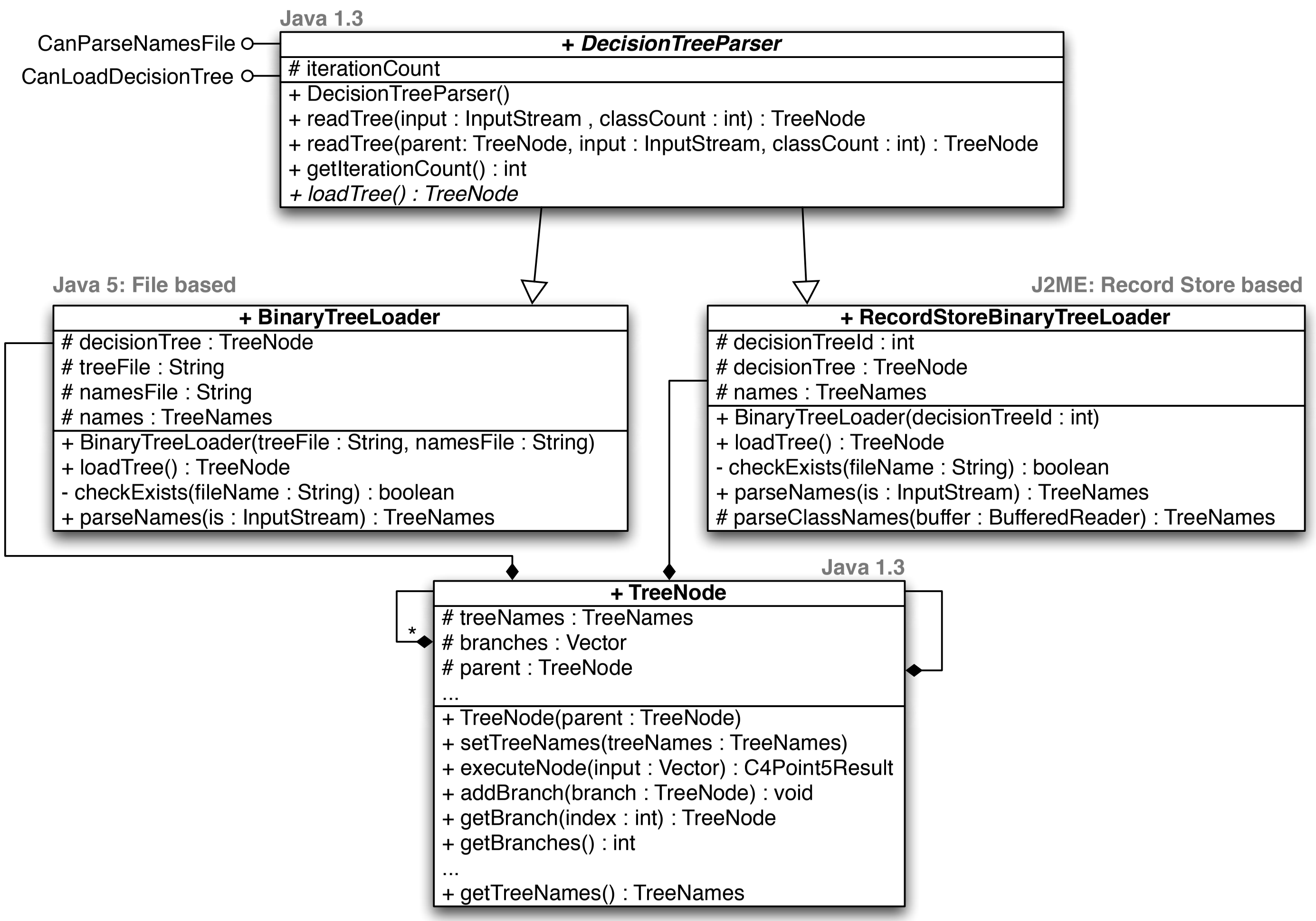


Figure B.8: UML hierarchy of the cross-platform C4.5 classifier. Here, two decision tree loaders are implemented to allow the classifier to execute on both PC and sensor node platforms file from flash memory. The binary file is parsed and a decision tree is generated in memory representing the C4.5 production rules. This decision tree is stored within the owner agent's data store; and,

ClassifyBehaviour: A one-shot behaviour which uses the decision tree and a new sensor measurement to generate a classification. Once a classification has been achieved, it is stored within the `InterpretedData` record store.

B.5 Summary

This guide outlines the architecture of the SubSense toolkit, and gives an insight of how its components may be used and extended to create agent-based wireless sensing applications. The toolkit was designed for substation condition monitoring applications, however it is completely generic in nature and may be used in the future, along with this guide, to build any agent-based application for the SunSPOT platform.

Appendix C

Microcontroller-based PD detector simulator

C.1 Introduction

This appendix gives a short description of a microcontroller application used to simulate the frequency-based partial discharge detector discussed in Chapter 5, and a standard RF PD detector using the phase-resolved method. These detectors are simulated in the Case Study chapter in Sections 7.3 and 7.4 respectively.

The simulator application was constructed to test the SubSense toolkit, discussed in Chapter 6 and Appendix B. The system allows defect data to be introduced into the diagnostic system in a controlled manner as if a physical detector was connected, without the need for high-voltage equipment but still using a physical interface. Referring back to the four components of a condition monitoring system used throughout this thesis (discussed initially in Section 2.2), the microcontroller is used to simulate the sensing and data capture components, specifically for an embedded diagnostic environment.

The application is built using a desktop Java application coupled with an application running on an Arduino microcontroller. The Arduino is a fully open-source microcontroller architecture designed for small-scale embedded applications, and like the SunSPOT devices used for the SubSense toolkit described in the previous appendix, all physical functions are abstracted behind an application programming interface enabling embedded applications to be built using standard programming techniques. For more information on the Arduino platform, please consult:

- "Arduino open-source electronics prototyping platform." Available online at `http://www.arduino.cc`, Feb. 2010

C.2 System overview

The PD detector simulator is capable of working in two configurations. Connecting to the SunSPOT device via a serial cable, the simulator can supplying spectral energy data for use with the frequency-based diagnostic method, and also supply phase-resolved partial discharge data used by the diagnostic methods developed within the COMMAS system. Figure C.1 shows the Arduino microcontroller connected to the SunSPOT device, with a USB tether connected to the Arduino for user interaction.

C.3 System operation

Taking the phase-resolved PD application as an example, when the application is run the user is presented with the screen shown in Figure C.2. Choosing one of these options then sends data across the serial link using a simple serial protocol.

C.3.1 Serial protocol

The serial protocol uses comma-separated string encoded data values with start and end packet characters 'a' and 'z' . An example of a PD pulse 3-channel spectral energy sample is:

```
a31,47,22,z
```

This protocol can be used for packets of any length, so for phase-resolved data which contain 3200 values, a truncated packet looks like:

```
a1,2,3,[omitted values],3198,3199,3200,z
```

As is, this protocol only allows a single type of PD detector to be simulated at any one time. There is no differentiation between the types of data sent using this application at the packet level - when simulating PD data it is necessary to run the correct data monitoring agent on the SubSense system to ensure that data is received properly. The application running on the Arduino device is a simple serial echo application. This can be seen in Figure C.3.

C.3.2 Integration with SubSense

A customised driver for the SubSense platform was written to allow serial data to be read by SubSense monitoring agents. A generic serial data driver prototype was created which was used for the basis of both the frequency-based and phase-resolved data drivers. The UML diagram for these drivers is shown in Figure C.4. Wrapping these drivers in standard monitoring agents, this allows the PD simulator to 'plug-in' to the SubSense system without any modification: to the rest of the system it appears as if an actual PD detector is present.

C.4 Summary

The microcontroller-based partial discharge detector simulator allows for partial discharge data to be injected into the SubSense diagnostic system without the use of high-voltage equipment. This system is demonstrated in operation within the Case Study chapter in Sections 7.3 and 7.4.

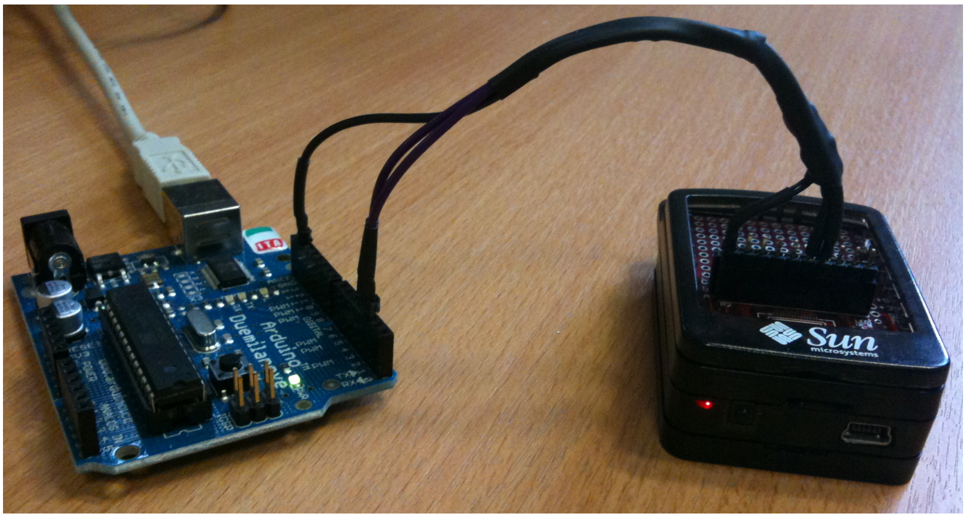


Figure C.1: Arduino microcontroller connected to a SunSPOT sensor node

```
Choose an defect from the list to send via the serial link

 1) Bad Contact
 2) Protrusion
 3) Rolling Particle
 4) Surface Discharge
 5) Floating Electrode
 6) Suspended Particle

 9) Exit app

 > 1
 > Sending data from BC106.l3d...
```

Figure C.2: PD detector simulator user interface, showing a simulated bad contact defect being transmitted across the serial link

```
// Sends batches of PD data across the serial link from a Java application.
#include <NewSoftSerial.h>

const int rxPin = 2;
const int txPin = 3;
const int DATA_LENGTH = 3200;

int currentNumber;

NewSoftSerial sunspotSerial(rxPin, txPin);

//sets up the application.  opens serial ports and prints welcome
void setup() {
  sunspotSerial.begin(115200);
  Serial.begin(115200);
  printWelcome();
}

//main program loop.  echoes data from a PC to the SunSPOT
void loop() {
  char input;

  if (Serial.available()) {
    input = (char) Serial.read();

    sunspotSerial.print(input);
    delayMicroseconds(150);  //wait for 100 microseconds before getting the next char
  }
}
```

Figure C.3: Arduino serial echo script for PD simulator application

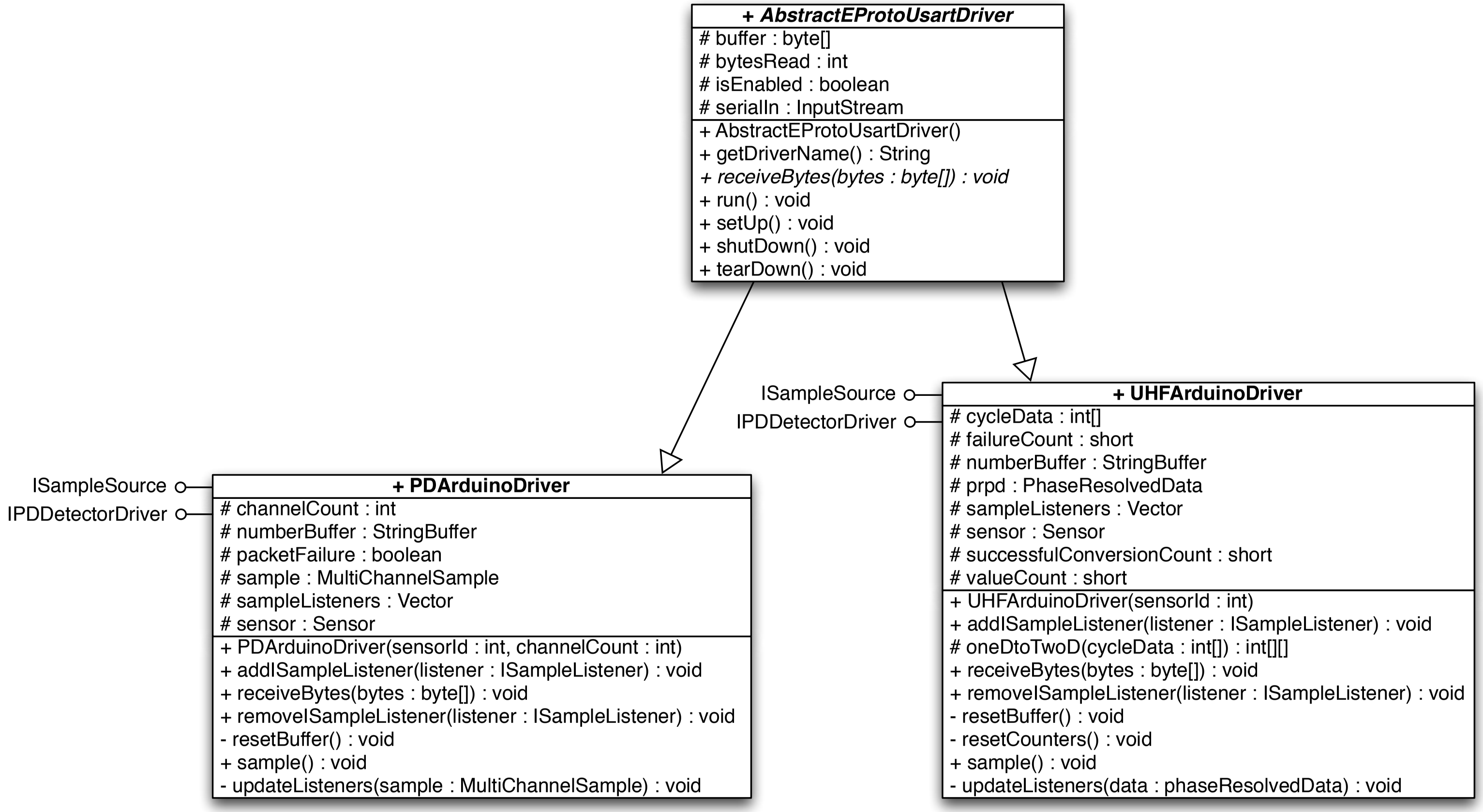


Figure C.4: PD simulator application driver UML hierarchy

References

[1] United Nations Development Program (UNDP), *Human Development Report 2007/2008*. United Nations Development Program, 2007/2008 ed., 2007. (cited on page 2)

[2] P. C. Baker, S. D. J. McArthur, and M. D. Judd, "A Frequency-Based RF Partial Discharge Detector for Low-Power Wireless Sensing," *IEEE Transactions on Dielectrics and Electrical Insulation*, vol. 17, pp. 133 –140, Feb. 2010.

[3] P. C. Baker, S. D. J. McArthur, and M. D. Judd, "Data Management of On-Line Partial Discharge Monitoring Using Wireless Sensor Nodes Integrated with a Multi-Agent System," in *Intelligent Systems Applications to Power Systems, 2007. ISAP 2007. International Conference on*, Nov. 2007.

[4] P. C. Baker, B. Stephen, M. D. Judd, and S. D. J. McArthur, "Development of an integrated low-power RF partial discharge detector," in *Electrical Insulation Conference, 2009. EIC 2009. IEEE*, pp. 273 –277, May 2009.

[5] P. C. Baker, V. M. Catterson, and S. D. J. McArthur, "Integrating an agent-based wireless sensor network within an existing multi-agent condition monitoring system," in *Proceedings of: Intelligent Systems Applications to Power Systems, 2009. ISAP 2009. International Conference on*, Dec. 2009.

[6] P. C. Baker, S. D. J. Rudd, S. McArthur, and M. D. Judd, "A conceptual design for a low-power wireless UHF detector," in *Proceedings of: Third UHVNet Colloquium on technologies for future high voltage infrastructure, 2010*, Jan. 2010.

[7] M. Zhu, P. C. Baker, N. M. Roscoe, M. D. Judd, and J. Fitch, "Alternative power sources for autonomous sensors in high voltage plant," in *Proceedings of the 2009 IEEE Electrical Insulation Conference (EIC), Montreal Canada*, pp. 36–40, June 2009. (cited on page 63)

[8] A. J. Mair, R. R. Soman, P. C. Baker, E. M. Davidson, S. D. J. McArthur, S. K. Srivastava, M. Andrus, and D. A. Cartes, "Progress in the development of adaptive control for shipboard power systems through modelling and simulations,"

in *Grand Challenges in Modeling and Simulation, Proceedings of.*, Society for Modelling and Simulation International, July 2010.

[9] H. Meinecke, "High voltage gas insulated switchgear: an overview," in *IEE Colloquium on Gas-Insulated Switchgear at transmission and distribution levels*, vol. 3, Nov. 1995. (cited on page 10)

[10] D. Warne, *Newnes electrical power engineer's handbook*. Elsevier, 2nd ed., 2005. (cited on pages 10, 11, 17)

[11] P. Bolin and H. Koch, "Gas insulated substation GIS," in *IEEE Power Engineering Society General Meeting, 2006*, p. 3, 2006. (cited on page 10)

[12] Y. Han and Y. H. Song, "Condition monitoring techniques for electrical equipment-a literature survey," *IEEE Transactions on Power Delivery, Vol. 18, No. 1*, pp. 4–13, Jan. 2003. (cited on pages 12, 13, 16)

[13] M. Wang, A. Vandermaar, and K. Srivastava, "Review of condition assessment of power transformers in service," *IEEE Electrical insulation magazine*, vol. 18, no. 6, pp. 12–25, 2002. (cited on pages 13, 14, 15)

[14] C. Bengtsson, "Status and trends in transformer monitoring," *IEEE Transactions on Power delivery*, vol. 11, no. 3, pp. 1379–1384, 1996. (cited on pages 13, 27, 126)

[15] O. C. Seevers, *Power Systems Handbook: Design, Operation and Maintenance*. Prentice-Hall, 1991. (cited on page 15)

[16] L. Pettersson, "Estimation of the remaining service life of power transformers and their insulation," *CIGRE Electra*, vol. 133, pp. 65–71, 1990. (cited on page 15)

[17] A. Bossi, J. Dind, J. Frisson, U. Khoudiakov, H. Light, D. Narke, Y. Tournier, and J. Verdon, "An international survey on failures in large power transformers in service," *CIGRE Electra*, vol. 88, 1983. (cited on page 16)

[18] Switchgear Committee of the IEEE Power Engineering Society, "IEEE Guide for Diagnostics and Failure Investigation of Power Circuit Breakers," Tech. Rep. C37.10TM-1995 (R2002), IEEE, May 2008. (cited on page 16)

[19] S. M. Strachan, S. D. J. McArthur, B. Stephen, J. R. McDonald, and A. Campbell, "Providing decision support for the condition-based maintenance of circuit breakers through data mining of trip coil current signatures," *IEEE Transactions on Power Delivery*, vol. 22, no. 1, pp. 178–186, 2007. (cited on pages 17, 28)

[20] CIGRE Task Force 15.03.07 of Working Group 15.03, "Long-term performance of SF6 insulated systems," Tech. Rep. 15-301, CIGRE, 2002. (cited on page 17)

[21] R. James and Q. Su, *Condition Assessment of High Voltage Insulation in Power System Equipment*. Institution of Engineering & Technology (IET), 2008. (cited on pages 17, 83)

[22] IEEE, "IEEE Guide for the Interpretation of Gases Generated in Oil-immersed Transformers," 1991. IEEE Standard C57.104-1991. (cited on page 18)

[23] IEC, "Mineral Oil-Impregnated Electrical Equipment in Service - Guide to the Interpretation of Dissolved and Free Gases analysis," 1999. Document IEC 60599. (cited on page 18)

[24] M. Duval, "A review of faults detectable by gas-in-oil analysis in transformers," *IEEE Electrical Insulation Magazine*, vol. 18, no. 3, pp. 8–17, 2002. (cited on pages 18, 19, 98)

[25] M. J. Heathcote, *The J&P Transformer Book. Chapter 8: Operation and Maintenance*. Newnes, 12th ed., 1998. (cited on page 20)

[26] J. S. Pearson, O. Farish, B. F. Hampton, M. D. Judd, D. Templeton, B. M. Pryor, and I. M. Welsh, "Partial Discharge Diagnostics for Gas insulated Substations," *IEEE Transactions on Dielectrics and Electrical Insulation*, vol. 2, pp. 893–905, Oct. 1995. (cited on pages 20, 24, 153)

[27] CIGRE Working Group 15.03, "Diagnostic Methods for GIS Insulating Systems," Tech. Rep. 15/23-01, CIGRE, Aug. 1992. (cited on page 20)

[28] CIGRE Task Force 15/11 33.03.02, "Knowledge rules for partial discharge diagnosis in service," tech. rep., CIGRE, Apr. 2003. (cited on pages 20, 21, 22, 26)

[29] G. P. Cleary and M. D. Judd, "UHF and current pulse measurements of partial discharge activity in mineral oil," *IEE Proceedings-Science, Measurement and Technology*, vol. 153, no. 2, pp. 47–54, 2006. (cited on pages 21, 82, 83)

[30] IEC, "High-Voltage Test Techniques - Partial Discharge Measurements," 2000. Document IEC 60270. (cited on pages 22, 83, 89)

[31] G. Vaillancourt and R. Malewski, "Digital acquisition and processing of partial discharges duringacceptance test of HV transformers," *IEEE Transactions on Power Delivery*, vol. 4, no. 1, pp. 421–427, 1989. (cited on page 22)

[32] A. Contin, A. Cavallini, G. C. Montanari, G. Pasini, and F. Puletti, "Digital detection and fuzzy classification of partial discharge signals," *Dielectrics and Electrical Insulation, IEEE Transactions on*, vol. 9, no. 3, pp. 335–348, 2002. (cited on pages 22, 85)

[33] L. Hao, P. L. Lewin, and S. G. Swingler, "Identification of multiple partial discharge sources," in *Condition Monitoring and Diagnosis, 2008. CMD 2008. International Conference on*, pp. 118–121, April 2008. (cited on page 22)

[34] S. M. Strachan, S. D. J. McArthur, M. D. Judd, and J. R. McDonald, "Incremental Knowledge-Based Approach to Partial Discharge Diagnosis in oil Filled Power Transformers," in *Intelligent Systems Application to Power Systems, 2005. Proceedings of the 13th International Conference on*, 2005. (cited on pages 22, 24)

[35] A. Cavallini, A. Contin, G. C. Montanari, and F. Puletti, "Advanced PD inference in on-field measurements. part I. noise rejection," *Dielectrics and Electrical Insulation, IEEE Transactions on*, vol. 10, no. 2, pp. 216–224, 2003. (cited on page 22)

[36] E. Howells and E. Norton, "Detection of partial discharges in transformers using acoustic emission techniques," *IEEE Transactions on Power Apparatus and Systems*, pp. 1538–1549, 1978. (cited on page 22)

[37] E. Grossmann and K. Feser, "Sensitive online PD-measurements of onsite oil/paper-insulated devices by means of optimized acoustic emission techniques (AET)," *IEEE Transactions on Power Delivery*, vol. 20, no. 1, pp. 158–162, 2005. (cited on page 23)

[38] S. Markalous, S. Tenbohlen, and K. Feser, "Detection and location of partial discharges in power transformers using acoustic and electromagnetic signals," *Dielectrics and Electrical Insulation, IEEE Transactions on*, vol. 15, pp. 1576–1583, December 2008. (cited on page 23)

[39] A. Reid, M. Judd, and C. Johnstone, "Development of UHF Transformer Probe Sensors for On-Line Partial Discharge Measurement," in *Proceedings of the 44th International Universities Power Engineering Conference*, (University of Strathclyde), Sept. 2009. (cited on page 23)

[40] B. F. Hampton and R. J. Meats, "Diagnostic measurements at uhf in gas insulated substations," *Generation, Transmission and Distribution, IEE Proceedings*, vol. 135, pp. 137–145, Mar 1988. (cited on pages 23, 91, 94)

[41] M. D. Judd, O. Farish, and B. F. Hampton, "The excitation of UHF signals by partial discharges in GIS," *Dielectrics and Electrical Insulation, IEEE Transactions on*, vol. 3, pp. 213–228, Apr. 1996. (cited on page 24)

[42] G. P. Cleary and M. D. Judd, "An investigation of discharges in oil insulation using UHF PD detection," in *Dielectric Liquids, 2002. ICDL 2002. Proceedings of 2002 IEEE 14th International Conference on*, pp. 341–344, 2002. (cited on page 24)

[43] M. D. Judd, G. P. Cleary, and C. J. Bennoch, "Applying UHF Partial Discharge Detection to Power Transformers," *IEEE Power Engineering Review*, Aug. 2002. (cited on page 24)

[44] M. D. Judd, L. Yang, and I. B. B. Hunter, "Partial Discharge Monitoring for Power Transformers using UHF Sensors Part 1: Sensors and Signal interpretation," *IEEE Electrical Insulation Magazine*, vol. 21, pp. 5–14, Mar. 2005. (cited on pages 24, 25, 87, 155)

[45] M. D. Judd, S. D. J. McArthur, A. J. Reid, V. M. Catterson, L. Yang, B. Jacobson, K. O. Svensson, and M. Gunnarsson, "Investigation of Radiometric Partial Discharge Detection for use in Switched HVDC Testing," *Power Engineering Society General Meeting, 2006. IEEE*, June 2006. (cited on page 24)

[46] Diagnostic Monitoring Systems (DMS), "Smartcoupler pd detection system." Available from `http://www.roheinc.com/Products/DMSPartialDischargeMonitoring/tabid/170/Default.aspx`, 2010. (cited on page 24)

[47] S. M. Strachan, G. Jahn, S. D. J. McArthur, and J. R. McDonald, "Intelligent Diagnosis of Defects Responsible for Partial Discharge Activity Detected in Power Transformers," in *Intelligent Systems Applications to Power Systems, 2003. ISAP 2003.*, Sept. 2003. (cited on pages 24, 47)

[48] J. Zhang, G. Jones, J. Spencer, P. Jarman, I. Kemp, Z. Wang, P. Lewin, and R. Aggarwal, "Chromatic classification of RF signals produced by electrical discharges in HV transformers," *IEE Proceedings-Generation, Transmission and Distribution*, vol. 152, no. 5, pp. 629–634, 2005. (cited on page 24)

[49] J. Zhang, G. R. Jones, A. G. Deakin, and J. W. Spencer, "Chromatic processing of DGA data produced by partial discharges for the prognosis of HV transformer behaviour," *Measurement Science and Technology*, vol. 16, no. 2, pp. 556–561, 2005. (cited on page 25)

[50] G. R. Jones, J. W. Spencer, L. M. Shpanin, and A. G. Deakin, “A chromatic analysis of current interrupters,” in *Gas Discharges and Their Applications, 2008. GD 2008. 17th International Conference on*, pp. 153 –156, sept. 2008. (cited on page 25)

[51] T. Pinpart, J. E. Fletcher, and M. D. Judd, “Methods for distinguishing between partial discharges based on the uhf detection technique,” in *Condition Monitoring and Diagnosis, 2008. CMD 2008. International Conference on*, pp. 1060–1064, Apr. 2008. (cited on pages 25, 84)

[52] M. D. Judd, L. Yang, and I. B. B. Hunter, “Partial Discharge Monitoring for Power Transformers using UHF Sensors Part 2: Field experience,” *IEEE Electrical Insulation Magazine*, vol. 21, pp. 5–13, May 2005. (cited on page 25)

[53] S. Meijer, *Partial discharge diagnosis of high-voltage gas-insulated systems*. PhD thesis, Technische Universiteit Delft, Nov. 2001. (cited on pages 26, 85, 87, 152)

[54] K. T. Muthanna, A. Sarkar, K. Das, and K. Waldner, “Transformer Insulation Life Assessment,” *IEEE Transactions on Power Delivery*, vol. 21, pp. 150–156, Jan. 2006. (cited on page 27)

[55] W. Lampe, L. Pettersson, C. Ovren, and B. Wahlstrom, “Hot-spot measurements in power transformers,” *International Conference on Large High Voltage Electric Systems*, vol. 1, Aug. 1984. (cited on page 27)

[56] IEEE, “IEEE Guide for Loading Mineral-Oil-Immersed Transformers,” 1996. IEEE Standard C57.91-1995. (cited on page 27)

[57] IEC, “Loading guide for oil-immersed power transformers,” 1991. Document IEC 60354. (cited on page 27)

[58] C. Booth and J. McDonald, “The use of artificial neural networks for condition monitoring of electrical power transformers,” *Neurocomputing*, vol. 23, no. 1-3, pp. 97–109, 1998. (cited on page 27)

[59] B. Garcia, J. C. Burgos, and A. M. Alonso, “Transformer tank vibration modeling as a method of detecting winding deformations—part II: experimental verification,” *IEEE Transactions on Power Delivery*, vol. 21, no. 1, pp. 164–169, 2006. (cited on page 27)

[60] B. Garcia, J. C. Burgos, and A. M. Alonso, “Transformer tank vibration modeling as a method of detecting winding deformations-part I: theoretical foundation,” *IEEE Transactions on Power Delivery*, vol. 21, no. 1, pp. 157–163, 2006. (cited on page 27)

[61] T. Bengtsson, H. Kols, L. Martinsson, M. Foata, F. Leonard, C. Rajotte, and J. Aubin, "Acoustic diagnosis of tap changers," *International Council on Large Electric Systems (CIGRÉ), Paris, France*, 1996. (cited on page 27)

[62] V. M. Catterson, S. E. Rudd, S. D. J. McArthur, and G. Moss, "On-line transformer condition monitoring through diagnostics and anomaly detection," in *Intelligent Systems Applications to Power Systems, 2009. ISAP 2009. International Conference on*, IEEE, Nov. 2009. (cited on pages 27, 41, 52, 53, 141)

[63] Areva T&D, "Areva CBwatch-2 Modular Circuit Breaker Monitoring System." Available from `http://www.areva-td.com/solutions/US_402_CBWatch+-+2+Modular+Circuit+Breaker+Monitoring+System.html`, 2010. (cited on page 28)

[64] J. Dewulf, T. Jung, J. Dupraz, and G. Montillet, "A development and application of circuit breakers diagnostic and monitoring," in *2003 IEEE PES Transmission and Distribution Conference and Exposition*, vol. 2, 2003. (cited on page 28)

[65] M. Kezunovic, Z. Ren, G. Latisko, D. Sevcik, J. Lucey, W. Cook, and E. Koch, "Automated monitoring and analysis of circuit breaker operation," *IEEE Transactions on Power Delivery*, vol. 20, no. 3, pp. 1910–1918, 2005. (cited on page 28)

[66] CIGRE Working Group A2.27i, "Recommendations for condition monitoring and condition assessment facilities for transformers," Tech. Rep. Brochure 343, CIGRE (Paris), Apr. 2008. (cited on pages 29, 87)

[67] P. J. Moore, I. E. Portugues, and I. A. Glover, "Partial discharge investigation of a power transformer using wireless wideband radio-frequency measurements," *IEEE Transactions on Power Delivery*, vol. 21, no. 1, pp. 528–530, 2006. (cited on page 29)

[68] T. K. Saha, "Review of Modern Diagnostic Techniques for Assessing Insulation Condition in aged Transformers," *IEEE Transactions on Dielectrics and Electrical Insulation*, vol. 10, pp. 903–917, Oct. 2003.

[69] M. J. Heathcote, *The J&P Transformer Book*. Newnes, 12th ed., 1998.

[70] S. D. J. McArthur, E. M. Davidson, V. M. Catterson, A. L. Dimeas, N. D. Hatziargyriou, F. Ponci, and T. Funabashi, "Multi-Agent Systems for Power Engineering Applications—Part 1: Concepts, Approaches and Technical Challenges," *IEEE Transactions on Power Systems*, vol. 22, pp. 1743–1752, Nov. 2007. (cited on pages 32, 46, 155)

[71] S. D. J. McArthur, E. M. Davidson, V. M. Catterson, A. L. Dimeas, N. D. Hatziargyriou, F. Ponci, and T. Funabashi, "Multi-Agent Systems for Power Engineering Applications—Part 2: Technologies, Standards and Tools for Building Multi-Agent Systems," *IEEE Transactions on Power Systems*, vol. 22, pp. 1753–1759, Nov. 2007. (cited on pages 32, 36, 41, 46, 52)

[72] M. Wooldridge and N. R. Jennings, "Intelligent agents: Theory and Practice," *Knowledge Engineering Review*, vol. 10, no. 2, pp. 115–152, 1995. (cited on page 32)

[73] Foundation for Intelligent Physical Agents, "FIPA Standards Repository." Available from `http://www.fipa.org/`, 2007. (cited on page 33)

[74] Foundation for Intelligent Physical Agents, "FIPA agent Management Specification." Avaliable from `http://fipa.org/specs/fipa00023/`, 2004. (cited on page 33)

[75] F. Bellifemine, G. Caire, A. Poggi, and G. Rimassa, "JADE: A software framework for developing multi-agent applications. Lessons learned," *Information and Software Technology*, vol. 50, no. 1-2, pp. 10 – 21, 2008. (cited on page 34)

[76] F. Bellifemine, G. Caire, and D. Greenwood, *Developing Multi-Agent Systems with JADE*. John Wiley and Sons Ltd, 1st ed., 2007. (cited on pages 34, 165)

[77] Foundation for Intelligent Physical Agents, "FIPA ACL Message Structure Specification." Available from `http://fipa.org/specs/fipa00061/`, 2002. (cited on page 34)

[78] Foundation for Intelligent Physical Agents, "FIPA SL Content Language Specification." Available from `http://fipa.org/specs/fipa00008/`, 2002. (cited on pages 35, 36, 38)

[79] Foundation for Intelligent Physical Agents, "FIPA Agent Communication Language Representation Specification." Available from `http://www.fipa.org/repository/aclreps.php3`, 2002. (cited on pages 35, 39)

[80] Foundation for Intelligent Physical Agents, "Fipa communicative act library specification." Available from `http://www.fipa.org/specs/fipa00037/`, Dec. 2002. (cited on pages 35, 38)

[81] Foundation for Intelligent Physical Agents, "FIPA interaction Protocol Specifications." Available from `http://fipa.org/repository/ips.html`, 2003. (cited on page 36)

[82] Foundation for Intelligent Physical Agents, "FIPA Request Interaction Protocol Specification." Available from http://www.fipa.org/specs/fipa00026/, Dec. 2002. (cited on pages 36, 37)

[83] M. R. Genesereth, R. E. Fikes, *et al.*, "Knowledge Interchange Format, version 3.0 reference manual," Tech. Rep. Logic 92-1, Computer Science Department, Stanford University, June 1992. (cited on page 36)

[84] D. L. McGuinness, F. Van Harmelen, *et al.*, "OWL web ontology language overview," *W3C recommendation*, vol. 10, pp. 2004–03, 2004. (cited on page 36)

[85] B. Schiemann and U. Schreiber, "OWL DL as a FIPA ACL content language," in *Proceedings of the Workshop on Formal Ontology for Communicating Agents*, pp. 73–80, July 2006. (cited on page 36)

[86] W3C, "World Wide Web consortium." http://www.w3.org/Consortium/, Apr. 2010. (cited on page 36)

[87] G. L. Steele, *Common LISP: the language*. Digital Pr, 1990. (cited on page 36)

[88] E. L. Soto, *Agent Communication Using Web Services, a New FIPA Message Transport Service for Jade*, pp. 73–84. Lecture Notes in Computer Science, Springer, Sept. 2007. (cited on page 39)

[89] H. Helin and M. Laukkanen, "Towards Efficient and Reliable Agent Communication in Wireless Environments," *Cooperative Information Agents V, Proceedings of the 5th International Workshop CIA*, pp. 258–263, 2001. (cited on pages 40, 71)

[90] T. R. Gruber *et al.*, "A translation approach to portable ontology specifications," *Knowledge acquisition*, vol. 5, pp. 199–199, 1993. (cited on page 40)

[91] N. F. Noy and D. L. McGuinness, "Ontology Development 101: A Guide to Creating your First ontology," Tech. Rep. KSL-01-05, Stanford Knowledge Systems Laboratory, Mar. 2001. Available from http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html. (cited on page 41)

[92] I. Niles and A. Pease, "Towards a standard upper ontology," in *FOIS '01: Proceedings of the international conference on Formal Ontology in Information Systems*, (New York, NY, USA), pp. 2–9, ACM, 2001. (cited on pages 41, 156)

[93] Q. Zhou and R. Fikes, "A reusable time ontology," in *Proceedings of the AAAI Workshop on Ontologies for the Semantic Web*, 2002. (cited on page 41)

[94] IEC, "Energy Management System Application Program Interface (EMS-API) - Part 301: Common information Model (CIM) base," 2005. Document IEC 61970-301. (cited on pages 41, 76)

[95] Foundation for Intelligent Physical Agents, "FIPA ontology Service Specification." Available from `http://fipa.org/specs/fipa00086/`, 2001. (cited on page 41)

[96] H. Suguri, E. Kodama, M. Miyazaki, H. Nunokawa, and S. Noguchi, "Implementation of FIPA ontology service," in *Proceedings of the Workshop on Ontologies in Agent Systems, 5th International Conference on Autonomous Agents Montreal, Canada*, 2001. (cited on page 41)

[97] A. Kalfoglou and M. Schorlemmer, "Ontology Mapping: the State of the Art," *The Kowledge Engineering Review*, vol. 18, no. 1, pp. 1–31, 2003. (cited on page 42)

[98] V. A. Pham and A. Karmouch, "Mobile software agents: an overview," *IEEE Communications Magazine*, vol. 36, pp. 26–37, July 1998. (cited on page 42)

[99] D. B. Lange and M. Oshima, "Seven good reasons for mobile agents," *Communications of the ACM*, vol. 42, no. 3, pp. 88–89, 1999. (cited on page 42)

[100] J. A. Stankovic, "Distributed computing," Tech. Rep. UM-CS-1992-011, IEEE Computer Society Press, 1992. (cited on page 42)

[101] G. Vigna, "Mobile Agents: Ten Reasons For Failure," in *Mobile Data Management, IEEE International Conference on*, (Los Alamitos, CA, USA), p. 298, IEEE Computer Society, 2004. (cited on page 43)

[102] R. A. Brooks, "Intelligence without representation," *Artificial Intelligence*, vol. 47, no. 1-3, pp. 139–159, 1991. (cited on page 43)

[103] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, ch. 25, pp. 901–942. Artificial Intelligence, Prentice Hall, 2nd ed., 2003. (cited on page 43)

[104] A. S. Rao and M. P. Georgeff, "BDI agents: From theory to practice," in *Proceedings of the first international conference on multi-agent systems (ICMAS-95)*, pp. 312–319, 1995. (cited on page 43)

[105] F. Bellifemine, G. Caire, and D. Greenwood, *Developing Multi-Agent Systems with JADE*, chapter JADE Semantics Framework, pp. 225–245. John Wiley and Sons, 1st ed., 2007. (cited on pages 46, 156)

[106] S. D. J. McArthur, S. M. Strachan, and G. Jahn, "The design of a multi-agent transformer condition monitoring system," *IEEE Transactions on Power Systems, Vol. 19, No. 4*, pp. 1845–1852, Nov. 2004. (cited on page 46)

[107] V. M. Catterson, S. D. J. McArthur, M. D. Judd, and A. S. Zaher, "Managing remote online partial discharge data," *IEEE Transactions on Power Delivery*, vol. 23, no. 4, pp. 1754–1762, 2008. (cited on page 46)

[108] S. Rudd, S. Strachan, M. D. Judd, and S. D. J. McArthur, "An incremental knowledge based approach to the analysis of partial discharge data," in *XVth International Symposium on High Voltage Engineering, Proceedings of*, vol. 5, 2007. (cited on page 47)

[109] V. M. Catterson and S. D. J. McArthur, "Using evidence Combination for Transformer Defect Diagnosis," *International Journal of Innovations in Energy Systems and Power (IJESP)*, vol. Issue 1, Nov. 2006. Invited paper. (cited on pages 47, 48, 51)

[110] E. M. Davidson, S. D. J. McArthur, J. R. McDonald, T. Cumming, and I. Watt, "Applying Multi-Agent System Technology in Practice: Automated Management and analysis of SCADA and Digital Fault Recorder Data," *IEEE Transactions on Power Systems*, vol. 21, pp. 559–567, May 2006. (cited on pages 47, 49)

[111] D. P. Buse and Q. H. Wu, "Mobile agents for remote control of distributed systems," *IEEE Transactions on Industrial Electronics*, vol. 51, Dec. 2004. (cited on page 49)

[112] A. Kezunovic, X. Xu, and D. Won, "Improving circuit breaker maintenance management tasks by applying mobile agent software technology," *Transmission and Distribution Conference and Exhibition 2002: Asia Pacific. IEEE/PES*, vol. 2, 2002. (cited on page 49)

[113] T. Nagata and H. Sasaki, "A Multi-Agent approach to Power System Restoration," *IEEE Transactions on Power Systems*, vol. 17, pp. 457–462, May 2002. (cited on page 50)

[114] J. Solanki, S. Khushalani, and N. Schulz, "A multi-agent solution to distribution systems restoration," *IEEE Transactions on Power Systems*, vol. 22, no. 3, pp. 1026–1034, 2007. (cited on page 50)

[115] I. Baxevanos and D. Labridis, "Software agents situated in primary distribution networks: a cooperative system for fault and power restoration management," *IEEE Transactions on Power Delivery*, vol. 22, no. 4, pp. 2378–2385, 2007. (cited on page 50)

[116] I. S. Baxevanos and D. P. Labridis, "Implementing multiagent systems technology for power distribution network control and protection management," *IEEE Trans. Power Del.*, vol. 22, pp. 433–443, Jan. 2007. (cited on page 51)

[117] E. M. Davidson, S. D. J. McArthur, J. R. McDonald, and P. Taylor, "An Architecture for flexible and autonomous network management systems," in *Proceedings of CIRED 2009*, 2009. (cited on page 51)

[118] J. Heo and K. Lee, "A multi-agent system-based intelligent identification system for power plant control and fault-diagnosis," in *IEEE Power Engineering Society General Meeting, 2006*, p. 6, 2006. (cited on page 51)

[119] J. Heo and K. Lee, "A Multiagent-System-Based Intelligent Reference Governor for Multiobjective Optimal Power Plant Operation," *IEEE Transaction on Energy Conversion*, vol. 23, no. 4, pp. 1082–1092, 2008. (cited on page 51)

[120] X. Song, M. Wu, C. Jermaine, and S. Ranka, "Conditional anomaly detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 5, p. 631, 2007. (cited on page 53)

[121] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks, Volume 38, Issue 4*, pp. 393–422, 2002. (cited on pages 57, 59, 73)

[122] D. Arvind and K. Wong, "Speckled Computing: Disruptive Technology for Networked Information Appliances," in *Consumer Electronics, 2004 IEEE International Symposium on*, Sept. 2004. (cited on page 58)

[123] IEEE 802.15 Working Group for WPAN, "IEEE 802.15.4-2006: Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (WPANs)." Available from `http://standards.ieee.org/getieee802/download/802.15.4-2006.pdf`, 2006. (cited on page 59)

[124] ZigBee Alliance, "Zigbee specification," Tech. Rep. 053474r06, ZigBee Alliance, 2005. (cited on page 59)

[125] G. Pottie and W. Kaiser, "Wireless integrated network sensors," *Communications of the ACM*, vol. 43, no. 5, pp. 51–58, 2000. (cited on page 59)

[126] F. Sivrikaya and B. Yener, "Time synchronization in sensor networks: A survey," *IEEE Network*, vol. 18, pp. 45–50, 2004. (cited on pages 59, 155)

[127] Epsom Toyocom, "FC-255: khz range crystal unit datasheet." Available from `http://www.eea.epson.com/portal/pls/portal/docs/1/1087994.PDF`. (cited on page 60)

[128] M. M. Nordman, M. Lehtonen, and O. Vahamaki, "Managing concurrent duties and time of wireless sensors in electrical power systems," in *IEEE Conference Emerging Technologies and Factory Automation, 2003. Proceedings. ETFA'03*, vol. 1, 2003. (cited on page 60)

[129] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energy-aware wireless microsensor networks," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 40–50, 2002. (cited on page 62)

[130] M. Zhu, M. Judd, and P. J. Moore, "Energy Harvesting in Substations for Powering Autonomous Sensors," in *Proceedings of the 2009 Third International Conference on Sensor Technologies and Applications-Volume 00*, pp. 246–251, IEEE Computer Society, 2009. (cited on page 63)

[131] N. D. Sadanandan and A. H. Eltom, "Power donut system laboratory test and data analysis," *IEEE Southeastcon'90. Proceedings.*, pp. 675–679, 1990. (cited on pages 63, 77)

[132] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. E. Culler, and K. S. J. Pister, "System architecture directions for networked sensors," in *Architectural Support for Programming Languages and Operating Systems*, pp. 93–104, 2000. (cited on page 64)

[133] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D. E. Culler, "The nesC language: A holistic approach to networked embedded systems," *Proc. of the ACM SIGPLAN*, 2003. (cited on page 64)

[134] P. Buonadonna, J. Hill, and D. Culler, "Active message communication for tiny networked sensors," in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM01)*, 2001. (cited on pages 64, 73)

[135] D. Simon, C. Cifuentes, D. Cleal, J. Daniels, and D. White, "Java on the bare metal of wireless sensor devices: the squawk Java virtual machine," in *Proceedings of the 2nd international Conference on Virtual Execution Environments*, p. 88, ACM, 2006. (cited on pages 65, 112)

[136] M. M. Wang, J. N. Cao, J. Li, and S. K. Dasi, "Middleware for wireless sensor networks: A survey," *Journal of Computer Science and Technology*, vol. 23, no. 3, pp. 305–326, 2008. (cited on page 66)

[137] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TinyDB: an acquisitional query processing system for sensor networks," *ACM Transactions on Database Systems (TODS)*, vol. 30, no. 1, p. 173, 2005. (cited on page 67)

[138] P. Levis and D. E. Culler, "Maté: a tiny virtual machine for sensor networks," *ACM SIGOPS Operating Systems Review*, vol. 36, no. 5, pp. 85–95, 2002. (cited on page 67)

[139] C. Curino, M. Giani, M. Giorgetta, A. Giusti, A. Murphy, G. Picco, D. E. e Informazione, and I. Milan, "TinyLIME: bridging mobile and sensor networks through middleware," *Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on*, pp. 61–72, 2005. (cited on page 67)

[140] D. Gelernter, "Generative communication in Linda," *ACM Transactions on Programming Languages and Systems*, vol. 7, no. 1, pp. 80–112, 1985. (cited on page 67)

[141] C.-L. Fok, G.-C. Roman, and C. Lu, "Rapid development and flexible deployment of adaptive wireless sensor network applications," in *24th International Conference on Distributed Computing Systems (ICDCS'05)*, 2005. (cited on pages 67, 69)

[142] H. Qi, S. Iyengar, and K. Chakrabarty, "Distributed multiresolution data integration using mobile agents," in *IEEE Aerospace Conference*, 2001. (cited on page 68)

[143] L. Szumel, J. LeBrun, and J. D. Owens, "Towards a mobile agent framework for sensor networks," in *Second IEEE Workshop on Embedded Networked Sensors (EmNetS-II)*, pp. 79–87, May 2005. (cited on page 68)

[144] D. Georgoulas and K. Blow, "Making motes intelligent: An agent-based approach to wireless sensor networks," *WSEAS on Communications Journal*, pp. 515–212, 2006. (cited on page 68)

[145] M. R. Genesereth and S. P. Ketchpel, "Software agents," *Communications of the ACM*, vol. 37, no. 7, 1997. (cited on page 68)

[146] F. L. Koch and J. J. C. Meyer, "Knowledge-based Autonomous Agents for Pervasive Computing Using AgentLight," *IEEE Distributed Systems Online*, vol. 4, no. 6, 2003. (cited on page 68)

[147] J. Lawrence, "LEAP into Ad-hoc Networks," in *Workshop on Ubiquitous Agents in Wearable, Embedded and Wireless Systems, AAMAS*, 2002. (cited on page 68)

[148] Foundation for Intelligent Physical Agents, "FIPA Agent Message Transport Envelope Representation in Bit Efficient Encoding Specification." Available from http://www.fipa.org/specs/fipa00088/, Dec. 2002. (cited on page 71)

[149] D. Crockford, "The application/json media type for javascript object notation (JSON)," Tech. Rep. RFC 4627, IETF Network Working Group, July 2006. (cited on page 74)

[150] L. Schor, P. Sommer, and R. Wattenhofer, "Towards a Zero-Configuration Wireless Sensor Network Architecture for Smart Buildings," in *BuildSys09, November 3, 2009, Berkeley, CA, USA.*, Nov. 2009. (cited on page 74)

[151] D. Yazar and A. Dunkels, "Efficient Application Integration in IP-Based Sensor Networks," in *BuildSys09, November 3, 2009, Berkeley, CA, USA.*, Nov. 2009. (cited on page 74)

[152] J. Herbert and C.-L. Fok, "Mobile Agent Architecture Integration for a Wireless Sensor Medical Application," *Web Intelligence and International Agent Technology Workshops, 2006. WI-IAT 2006 Workshops. 2006 IEEE/WIC/ACM International Conference on*, pp. 235–238, 2006. (cited on page 75)

[153] S. Rooney, D. Bauer, and P. Scotton, "Edge server software architecture for sensor applications," in *Proc. of the Symposium on Applications and the Internet (SAINT 05) Trento, Italy*, vol. 31, 2005. (cited on page 75)

[154] National Institute of Standards and Technology, "IEEE-P1451 Draft Standard Home Page." http://ieee1451.nist.gov/, June 2005. (cited on page 76)

[155] K. Aberer, M. Hauswirth, and A. Salehi, "Middleware support for the "Internet of Things"," in *5. GI/ITG KuVS Fachgespräch "Drahtlose Sensornetz"*, Universität Stuttgart, July 2006. (cited on page 76)

[156] M. Sveda and R. Vrba, "Integrated smart sensor networking framework for sensor-based appliances," *IEEE Sensors Journal*, vol. 3, pp. 579–586, Oct. 2003. (cited on page 76)

[157] Open Geospatial Consortium Inc., "OpenGIS sensor model language (SensorML) implementation specification," Tech. Rep. OGC 07-000, Open Geospacial Consortium, July 2007. (cited on page 76)

[158] C. Goodwin and D. J. Russomanno, "An ontology-based sensor network prototype environment," in *Proceedings of the 5th International Conference on Information Processing in Sensor Networks*, pp. 1–2, 2006. (cited on page 76)

[159] M. Eid, R. Liscano, and A. El Saddik, "A novel ontology for sensor networks data," in *Computational Intelligence for Measurement Systems and Applications, Proceedings of 2006 IEEE International Conference on*, pp. 75–79, July 2006. (cited on page 76)

[160] S. Avancha, C. Patel, and A. Joshi, "Ontology-driven adaptive sensor networks," *Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference on*, pp. 194–202, 2004. (cited on page 76)

[161] IEC, "Communications networks and Systems in Substations," 2005. Document IEC 61850. (cited on page 76)

[162] V. M. Catterson, E. M. Davidson, and S. D. J. McArthur, "Issues in Integrating Existing Multi-Agent Systems for Power Engineering applications," *Intelligent Systems Application to Power Systems, 2005. Proceedings of the 13th International Conference on*, Nov. 2005. (cited on pages 76, 155)

[163] K. Lau, J. Yee, R. Mayeda, J. Riccomini, M. Ilyin, *et al.*, "Fault sensor device with radio transceiver," Aug. 27 1996. US Patent 5,550,476. (cited on page 77)

[164] K. Rautiainen, K. Kauhaniemi, and O. Vahamaki, "Measurement of quantities of electric line," Sept. 3 2002. US Patent App. 10/232,403. (cited on page 77)

[165] P. J. Moore, I. E. Portugues, and I. A. Glover, "Pollution of the radio spectrum from the generation of impulsive noise by high voltage equipment," *IEE Seminar Digests*, vol. 37, 2002. (cited on page 77)

[166] Q. Shan, I. A. Glover, P. J. Moore, I. E. Portugues, R. J. Watson, and R. Rutherford, "Performance of Zigbee in Electricity Supply Substations," *Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on*, pp. 3866–3869, 2007. (cited on page 77)

[167] M. Qureshi, A. Raza, D. Kumar, S.-S. Kim, U.-S. Song, M.-W. Park, H.-S. Jang, H.-S. Yang, and B.-S. Park, "A survey of communication network paradigms for substation automation," in *Power Line Communications and Its Applications, 2008. ISPLC 2008. IEEE International Symposium on*, pp. 310–315, Apr. 2008. (cited on page 77)

[168] M. M. Nordman and M. Lehtonen, "A wireless sensor concept for managing electrical distribution networks," in *IEEE PES Power Systems Conference and Exposition, 2004*, pp. 1198–1206, 2004. (cited on page 77)

[169] H. Xuyong, M. Shihong, and L. Pei, "A wireless sensor network for distribution bus protection," in *Mobile Data Management, 2007 International Conference on*, pp. 382–386, May 2007. (cited on page 78)

[170] A. Nasipuri, R. Cox, H. Alasti, L. Van der Zel, B. Rodriguez, R. McKosky, and J. A. Graziano, "Wireless sensor network for substation monitoring: design and deployment," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*, pp. 365–366, ACM New York, NY, USA, 2008. (cited on page 78)

[171] Q. Gao and H. Wang, "WSN design in high-voltage transformer substation," *Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on*, pp. 6720–6724, June 2008. (cited on page 78)

[172] D. E. Culler, D. Estrin, and M. Srivastava, "Guest editors introduction: Overview of sensor networks," *IEEE Computer, Vol. 37, No. 8*, Aug. 2004.

[173] T. Pinpart and M. Judd, "Improving uhf partial discharge location in high voltage equipment," in *Proceedings of the 44th International Universities Power Engineering Conference*, (University of Strathclyde), September 2009. (cited on page 83)

[174] A. J. Reid, M. D. Judd, and R. A. Fouracre, "Frequency distribution of RF energy from PD sources and its application in combined RF and IEC60270 measurements," in *Electrical Insulation and Dielectric Phenomena, 2006 IEEE Conference on*, pp. 640–643, 2006. (cited on page 85)

[175] A. Cavallini, M. Conti, A. Contin, and G. C. Montanari, "Advanced PD inference in on-field measurements. part II. identification of defects in solid insulation systems," *Dielectrics and Electrical Insulation, IEEE Transactions on*, vol. 10, no. 3, pp. 528–538, 2003. (cited on page 85)

[176] A. J. Reid, *A New Approach to Partial Discharge Measurements for Testing Electrical Insulation Systems*. PhD thesis, University of Strathclyde, Glasgow, UK, 2007. (cited on page 86)

[177] B. Sharma, *Metal–Semiconductor Schottky Barrier Junctions and Their Applications*. NY: Plenum Press, 1984. (cited on page 89)

[178] P. J. G. Orr, A. J. Reid, and M. D. Judd, "Sensor response characteristics for UHF location of PD sources," in *Condition Monitoring and Diagnosis, 2008. CMD 2008. International Conference on*, pp. 1119–1122, 2008. (cited on page 91)

[179] J. W. Sammon, "A nonlinear mapping for data structure analysis," *IEEE Transactions on computers*, vol. 18, no. 5, pp. 401–409, 1969. (cited on page 98)

[180] I. Jolliffe, *Principal component analysis*. Springer-Verlag, 2002. (cited on page 98)

[181] G. Clayton and S. Winder, *Operational Amplifiers*, ch. 8.9, pp. 213–215. Newnes, 5th ed., 2003. (cited on page 104)

[182] R. Goldman, "Using the LIS3L02AQ Accelerometer," Application Note, Sun Microsystems, Feb. 2007. (cited on page 107)

[183] E. M. Davidson, S. D. J. McArthur, J. R. McDonald, T. Cumming, and I. Watt, "Applying multi-agent system technology in practice: automated management and analysis of scada and digital fault recorder data," *IEEE Transactions on Power Systems*, vol. 61, pp. 559–567, May 2006. (cited on page 111)

[184] V. M. Catterson, *Engineering robustness, flexibility, and accuracy into a multi-agent system for transformer condition monitoring*. PhD thesis, University of Strathclyde, Sept. 2006. (cited on pages 115, 141)

[185] J. R. Quinlan, *C4. 5: Programs for Machine Learning*. Morgan Kaufmann, 1993. (cited on pages 135, 181)

[186] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986. (cited on page 135)

[187] J. R. Quinlan, "Ross Quinlan's personal homepage: C4.5 source code." `http://www.rulequest.com/Personal/`, Dec. 2009. (cited on page 136)

[188] J. R. Quinlan, "C5.0 Online Tutorial." Available from `http://www.rulequest.com`, 2003. (cited on page 143)

[189] Sun Microsystems, "SunSPOT Owners Manual - Red Release 5.0." Available from `https://www.sunspotworld.com/docs`, June 2009. (cited on page 167)

[190] Sun Microsystems, "SunSPOT Theory of Operation - Red Release 5.0." Available from `http://www.sunspotworld.com/docs`, June 2009.

[191] Sun Microsystems, "Sun Small Programmable Object Technology (Sun SPOT) Developers Guide." Available from `http://www.sunspotworld.com/docs`, May 2009.

[192] Sun Microsystems, "Application Isolation API Specification." Online at `http://jcp.org/en/jsr/detail?id=121`, June 2006. Java Specification Request 121. (cited on page 168)

[193] "Arduino open-source electronics prototyping platform." Available online at `http://www.arduino.cc`, Feb. 2010.