# TOPOLOGY CONTROL AND DATA HANDLING IN WIRELESS SENSOR NETWORKS

Lam Ling Shum

Department of Electronic and Electrical Engineering
University College London

Thesis submitted for the degree of Doctor of Philosophy
at the University College London.

Supervisor: Dr. John Mitchell
Second Supervisor: Dr. Yang Yang

# DECLARATION

Date: **August 2009**

Author: **Lam Ling Shum**

Title: **Topology Control and Data Handling in Wireless Sensor Networks**

Department: **Electronic and Electrical Engineering**

Degree: **Ph.D.** Year: **2009**

I have read and understood the College and Departments statements and guidelines concerning plagiarism.

I declare that all material described in this report is all my own work except where explicitly and individually indicated in the text. This includes ideas described in the text, figures and computer programs.

Name:

Signature:

Date:

_____

Signature

*To Bryan, Mark and Sophia*

# Table of Contents

# Acknowledgement

There are a number of people to whom I owe my gratitude to for this PhD thesis. Without them, I would not have come this far with my PhD.

First, I would like to thank Dr. John Mitchell, for "adopting" me as his PhD student half way through the PhD. Not only has he given me valuable advice on my work, he has also put a lot of effort to secure my funding since my first supervisor left the university. Without his support I would never have been able to continue with the PhD.

I would like to thank Dr. Matthew Britton for supporting me and for offering me a job in his start-up company at a crucial time when I was waiting for the funding to continue my PhD. The job has allowed me to continue to research on Wireless Sensor Network, given me the opportunity to experience working on real WSN projects and provided me with the essential finance for living and childcare.

Next, I would like to thank Dr. Lionel Sacks, my first supervisor, who has chosen to further his career outside UCL during my PhD. He has inspired the majority of work in this thesis and has given me the opportunity to work for the SECOAS project and start my PhD with him.

To my husband Bryan, I would like to thank him for his immense support morally and financially. Thanks also for going through my thesis word by word looking for mistake, which cannot be very interesting to an accountant.

To my colleagues and friends, to quote a few, Ibiso, Moses, Hamed, Raul and Richard at the department, thank you all for the endless debates that lead to better research. Thank you for going through the ups and downs of being a PhD student with me.

# Abstract

Our work in this thesis have provided two distinctive contributions to WSNs in the areas of data handling and topology control.

In the area of data handling, we have demonstrated a solution to improve the power efficiency whilst preserving the important data features by data compression and the use of an adaptive sampling strategy, which are applicable to the specific application for oceanography monitoring required by the SECOAS project. Our work on oceanographic data analysis is important for the understanding of the data we are dealing with, such that suitable strategies can be deployed and system performance can be analysed. The Basic Adaptive Sampling Scheduler (BASS) algorithm uses the statistics of the data to adjust the sampling behaviour in a sensor node according to the environment in order to conserve energy and minimise detection delay.

The motivation of topology control (TC) is to maintain the connectivity of the network, to reduce node degree to ease congestion in a collision-based medium access scheme; and to reduce power consumption in the sensor nodes. We have developed an algorithm Subgraph Topology Control (STC) that is distributed and does not require additional equipment to be implemented on the SECOAS nodes. STC uses a metric called subgraph number, which measures the 2-hops connectivity in the neighbourhood of a node. It is found that STC consistently forms topologies that have lower node degrees and higher probabilities of connectivity, as compared to k-Neighbours, an alternative algorithm that does not rely on special hardware on sensor node. Moreover, STC also gives better results in terms of the minimum degree in the network, which implies that the network structure is more robust to a single point of failure. As STC is an iterative algorithm, it is very scalable and adaptive and is well suited for the SECOAS applications.

# List of Figures

# List of Tables

# List of Abbreviation

**a.a.s** asymptotically almost surely

**ACF** Autocorrelation Function

**ACSE** Advanced Communications Systems Engineering group

**ADC** Analogue to Digital Convertor

**BASS** Basic Adaptive Sampling Scheduler

**CBTC** Cone Base Topology Control

**CENS** Centre for Embedded Networked Sensing

**CSMA-CA** Carrier Sensed Medium Access - Collision Avoidance

**CTR** Critical Transmission Range

**DCT** Discrete Cosine Transform

**DND** Deviation from Normal Distribution

**DT** Delaunay Triangulation

**DTI** Department of Trade and Industry

**DWT** Discrete Wavelet Transform

**EWMA** Exponentially-Weighted Moving Average

**FDMA** Frequency Division Multiple Access

**fGn** Fractional Gaussian Noise

**FRC** Fixed Ratio of Largest Coefficient

**FTU** Formazin Turbidity Units

**GG** Gabriel Graph

**GRG** Geometric Random Graph

**IP** Internet Protocol

**IWT** Inverse Wavelet Transform

**LB** Lower bound

**LMST** Local Minimum Spanning Tree

**LOS** Line of Sight

**LRD** Long Range Dependency

**LSE** Least Square Estimate

**MAC** Medium Access Control

**MCU** Micro-controller Unit

**MRA** Multi-Resolution Analysis

**NIWA** National Institute of Water and Atmospheric Research

**RMS** Root Mean Square

**RNG** Relative Neighbourhood Graph

**R/S statistics** Rescaled Range statistics

**SDF** Spectral Density Function

**SECOAS** Self-Organised Collegiate Sensor Network project

**SG** Subgraph

**STC** Subgraph Topology Control

**TC** Topology Control

**TDMA** Time Division Multiple Access

**UB** Upper Bound

**UDG** Unit Disk Graph

**w.h.p.** with high probability

**WLSE** Wavelet Weighted Least Square Fit Estimator

**WSN** Wireless Sensor Network

**YG** Yao Graph

# Part I

# Introduction

# Chapter 1

# Introduction

## 1.1 Motivations

With the advancement of low-cost microprocessing chips and wireless technology, infrastructure free ad-hoc networks have become attractive alternatives to more traditional wired sensor monitoring in terms of cost, readiness of deployment, robustness and flexibility. Sensors equipped with wireless communication devices can be deployed on bridges, glaciers, in the ocean, on landfills, and other remote locations that are difficult to gain access to for environmental monitoring purposes. A network can be set up in a day with data being sent back to the users almost immediately. Network maintenance involves only replacing the faulty sensors or the used up batteries once every few years. And the network will still function properly with a few sensors missing.

Wireless Sensor Network (WSN) has opened up new challenges for the researchers. Sensor nodes are low-cost devices that have limited processing and memory power. Most sensor nodes are battery operated since a mains power supply is scarce in our researched environment (for example, the ocean). Moreover, to support scalability, sensor nodes are required to be autonomous and self-organising. Preferably, the network would have no communication infrastructure to maintain and would be able to function with a few failing nodes, a changing environment and node mobility.

This thesis is based on the research work for the Self-Organised Collegiate Sensor Network (SECOAS) project and similar environmental monitoring applications (see Section 2.4). The SECOAS project aimed to explore the benefit of using WSN in oceanographic monitoring. Traditionally, oceanography monitoring is achieved by a small number (often only one) of expensive and high precision sensing unit. Sensing data are retrieved directly from the equipment at the end of the experiment after the unit is recovered. The implementation of a WSN provides an alternative solution by deploying a larger number (10-100) of disposable sensor nodes in one site. The nodes are equipped with sensors with less precision, however, the network as a whole provides better spatial resolution of the area to the oceanographers and the user can have near-to-real-time access to the data.

Our first task is to develop a power efficient data handling technique to relay data and alerts back to the user. It is important to understand the user requirements and the environment that we are dealing with, such that we can evaluate the performance of the data handling algorithms and make sure that the requirements are met. This is the reason why the analytical work of the SECOAS oceanographic data is important. On the other hand, it is preferable that the developed algorithms can be reused in other WSN projects as well as the SECOAS project. Hence, the algorithms are required to be as generic as possible and independent of the data features.

Relaying raw data back to the user is expensive in term of power consumption and in-network processing of data is encouraged. This is because sending and receiving data via radio uses a lot more power than data processing in a sensor node. Moreover, a long distance radio path costs more energy than a multi-hop path from the same source to destination. Neighbour coordination is generally used to improve a sensor node's view on the local measuring environment. As a mean to conserve power, it is preferable to send compressed data, reports and alerts rather than raw data back to the user to reduce the number of nodes-to-basestation communications in the

network.

Whilst it is possible to compress data in a sensor node based on redundancy or irrelevancy in the data, a more radical solution is to minimise the number of samples taken at the time. Such a strategy can reduce the power used for both sampling and relaying data back to the user. For example, if the resolution required for the temperature reading is $0.5^oC$, then a new sample is only required when the temperature has risen or dropped $0.5^oC$ from the last reading. Another application is event detection where readings are only important when they cross certain thresholds. Both of these applications require the sampling algorithm to either predict or track the rate of change of the environment and determine when sampling is needed.

Most existing sampling algorithms require the use of statistical prediction models, which are heavily dependent on the data characteristics of the measured environment. The models require training and re-training to maintain the accuracy of the predictions, which imposes high processing and memory requirements on the sensor nodes. An alternative sampling method is sought for the SECOAS project. This must also be capable of being used on other WSN projects, have lesser processing and memory requirements and be simple to implement.

The second part of the thesis researches on topology control (TC), which is the automatic construction of a reliable communication network in WSN by varying the transmission power of each sensor node. In addition to the general requirement of reducing the total power consumption in the network for communication, another motivation of TC is to reduce the node degree (number of neighbours) of sensor nodes in order to control the collision rate in the networking layer and simplify routing and broadcasting mechanisms.

The challenge of TC is that WSNs are irregular networks where nodes are mostly deployed at points of interest. The placements are restricted by the deployment environment but relaying nodes may be added to strengthen the network structure.

Hence, node density varies in different parts of the network.

As WSNs are required to be distributed, scalable and self-organising, sensor nodes can only evaluate local connectivity by exchanging messages with their local neighbours. Long distance communication is discouraged due to high energy cost. It is important to find a metric that can represent local connectivity and provide additional insight into the whole network. TC optimisation should result in a connected and resilient network topology.

Currently, there are TC algorithms that guarantee connectivity in the network, however, they depend on information, e.g., position estimate and directional information of neighbouring nodes that requires special hardware to implement. The k-neighbours algorithm only uses the number of neighbours as its TC metric, which is readily available in a sensor node. However, k-neighbours does not guarantee connectivity and is prone to the *island effect*, where a highly-clustered group of nodes is disconnected from the rest of the network. Therefore, we explore an algorithm which does not require special hardware to implement and gives better results in terms of connectivity than k-neighbours. Moreover, we would like to improve on the reliability of the network structure such that redundancy routes are available in the network. These criteria are important in the SECOAS project, where communications are easily affected by the weather and sea conditions.

## 1.2 Thesis Structure

This thesis has two distinctive parts researching on the data handling and topology control aspects of WSN. Data analysis, data compression and sampling scheme, which are directly related to the SECOAS project can be found in Chapters 3 and 4 in Part II of the thesis. The development of a topology control algorithm and a thorough performance analysis can be found in Chapters 5 to 8 in Part III of the thesis.

The contents of the individual chapters are listed below.

- In Chapter 2, we introduce the general aspects of WSN common to both the data handling and topology control parts of our research, which include the characteristics, literature review and modelling of the WSN. We also introduce the philosophy and background of the SECOAS project in this chapter.

- In Chapter 3, the results of data analysis of the temporal oceanography data for temperature, salinity, wave pressure and sediment concentration are reported. The analysis has led to the development of a threshold-based wavelet compression strategy to compress the raw data based on the quality requirement.

- In Chapter 4, we introduce a novel sampling technique, BASS, using exponentially-weighted moving average (EWMA) for the determination of sampling frequency at that instance. The algorithm is compared to eSense [LCS06] with regards to their efficiency of capturing events, sampling rates and preliminary requirements.

- In Chapter 5, we describe the motivations, background theories of topology control in wireless sensor network followed by a literature research of the existing algorithms.

- In Chapter 6, we introduce Subgraph Topology Control (STC) and propose the use of the metric *Subgraph Number* for measuring local connectivity. We then describe the basic algorithm and the feedback mechanism of STC for event detection.

- In Chapter 7, the parameters of STC are analysed by varying the network size and density so that we can understand the capacity of STC under various environmental conditions. The performance of the algorithm is benchmark against the Critical Transmission Range (CTR) [SB03] and k-Neighbours algorithms [BLRS06] for their ability to maintain connectivity and the resulting

node degree and their power efficiency.

- In Chapter 8, we investigate the behaviour of STC in dynamic conditions similar to the actual deployment scenario. We simulate the situations where nodes are added or removed and when the nodes are deployed one-by-one.

## 1.3 Contributions

My personal contributions to the SECOAS project and research in WSN for environmental monitoring are listed as follow.

- Developed a robust Subgraph Topology Control (STC) algorithm for SECOAS type of environmental monitoring applications. (Chapter 6-8). STC was an iterative algorithm which constructed a communication network in a distributed manner, maintained a good network connectivity and reduced power consumption in the network. The performance of STC was simulated against a popular algorithm k-Neighbours, which had similar implementation requirement as STC. The results showed that STC achieved better network connectivity, lower physical degree and higher minimum degree compared to k-Neighbours algorithm. The adaptability of STC was also simulated against addition and removal of nodes in the network.

- Analysed the temporal oceanographic data (Chapter 3) for the SECOAS project to identify characteristics in the data that may be useful in the development of a data compression algorithm and a sampling strategy. Times series of the data were observed, spectral analysis was used to investigate the wave and tidal characteristic of the data. Multi-resolution analysis was performed on the sediment concentration data to prepare for the wavelet threshold compression in section 3.5. Three different self-similarity estimators were also evaluated for their accuracy when used on data with trends and periodic features.

- Developed Basic Adaptive sampling Scheduler (BASS) for the SECOAS project (Chapter 4), which scheduled the sensor nodes to sample more often when events were likely to happen. The performance of BASS was simulated against eSense, a model-based adaptive sampling algorithm for their event detection and sampling rates.

- Developed a wavelet threshold compression strategy for the SECOAS project (Chapter 3). The compression strategy used wavelet transform and run-length zeros coding techniques. The performance of the compression strategy was tested against the sediment concentration data analysed in Section 3.4.

## 1.4 Publications

The research in this thesis has led to the following publications.

- L.Shum and J.Mitchell, "An Adaptive Subgraph-based Topology Control Algorithm for Wireless Sensor Networks" IEEE Journal on Selected Areas in Communications (JSAC) - Simple Wireless Sensor Networking Solutions (submitted).

- L.Shum and L. Sacks, "Localised Topology Measurement and Control in Wireless Sensor Network", London Communications Symposium, 2006.

- L.Zhang, L.Shum, L.Sacks, "Network Topological Analysis Utilizing Significent Profiles for Sensor network application", London Communications Symposium, 2005.

- L.Shum and L.Sacks, "A Data-centric and Statistical-based Random Sensing Scheduler for Rare Event Detection in Distributed Wireless Sensor Networks", London Communications Symposium, 2005.

- M.Britton, L.Shum, L.Sacks, H.Haddadi, "A biologically-inspired approach to designing wireless sensor networks", proceedings of the second European workshop on wireless sensor networks, Istanbul, 2005.

- I.Wokoma, L.Shum, L.Sacks, I.Marshall, "A Biologically-inspired clustering algorithm dependent on spatial data in sensor networks", poster section, proceedings of the second European workshop on wireless sensor networks, Istanbul, 2005.

- L. Shum and L. Sacks, "Data Analysis and Investigation of Self-Similarity in Oceanographic Sediment Data", London Communications Symposium, 2004.

- L. Shum, I. Wokoma, T. Adebutu, A. Marbini, L. Sacks and M. Britton, "Distributed Algorithm Implementation and Interaction in Wireless Sensor Networks", 2nd International Workshop on Sensor and Actor Network Protocols and Applications 2004.

# Chapter 2

# Wireless Sensor Networks

**Nomenclature in this chapter**

| symbol | description | equation | page no. |
|---|---|---|---|
| $d$ | distance | 2.3.1 | 29 |
| $P_r$ | receive power | 2.3.1 | 29 |
| $P_t$ | transmission power | 2.3.1 | 29 |
| $\alpha$ | path loss exponent | 2.3.1 | 29 |
| $\delta$ | dimension of the network | - | 30 |
| $l$ | length of the world | - | 30 |
| $n$ | number of nodes | | 30 |
| $r$ | transmission range | - | 30 |
| $r_{max}$ | maximum transmission range | - | 30 |

Wireless sensor networks (WSN) are a particular type of ad hoc networks, in which the nodes are sensors equipped with wireless transmission capability. Hence, they have the characteristics, requirements and limitations of an ad hoc network [San05a].

The term ad hoc network describes a type of wireless network without a fixed infrastructure. Conventional wireless networks including WiFi and cellular networks have supporting backbones and are hierarchical. Nodes communicate with each other via the base stations. In an ad hoc network the nodes can communicate with each other directly via multi-hops paths. Usually the network does not have any

coordinating node and hence, ad hoc networks are decentralised, self-organised and self-healing. Messages may be duplicated on the way to the base station to provide extra resilience [ASSC02].

A WSN is usually composed of a large number of sensing nodes in the order of tens, hundreds or even thousands scattered in a sensor field and one or a few base stations/ sinks, which connect the sensor networks to the users via the Internet or other networks. The nodes are deployed either inside the observed phenomenon or very close to it. Sensor nodes are equipped with sensing, data processing and communicating components to accomplish their tasks. Each of the sensor nodes are capable of collecting data and routing the data back to the sink by multi-hopping, as illustrated in Figure 2.1.

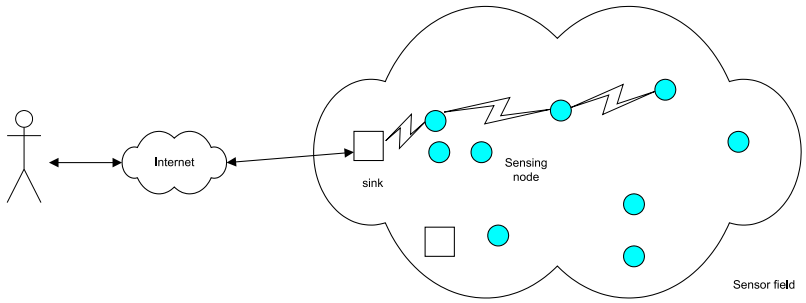



Figure 2.1: Basic structure of a wireless sensor network

The work in this thesis is based on the Self-Organised Collegiate Sensor Network (SECOAS) project (see Section 2.4). In this Chapter, we first describe the general characteristics, motivations and structure of common wireless sensor network for environmental monitoring in Section 2.1 and 2.2. We then discuss the WSN modelling techniques used in this thesis in Section 2.3. Finally we introduce the goal, system design and research objectives of the SECOAS project in Section 2.4.

## 2.1 Characteristics

The design of a WSN can be divided into *data handling* and communication aspects. Data handling tackles areas including the sensing interface, data sampling rate, data compression and fusion, clustering, decision making and result reporting. The communication aspects of a WSN include the radio module, network synchronisation, Medium Access Control (MAC) scheduling, topology control and networking strategy and information dissemination.

The protocol stacks of a WSN can be variable with different applications. Figure 2.2 is the protocol stack used in the SECOAS project. The communication and data parts of the stack run in parallel in the lower layers providing the basic infrastructure to support the implementation of the higher layer intelligent algorithms, which includes data dissemination and network topology adaptations.

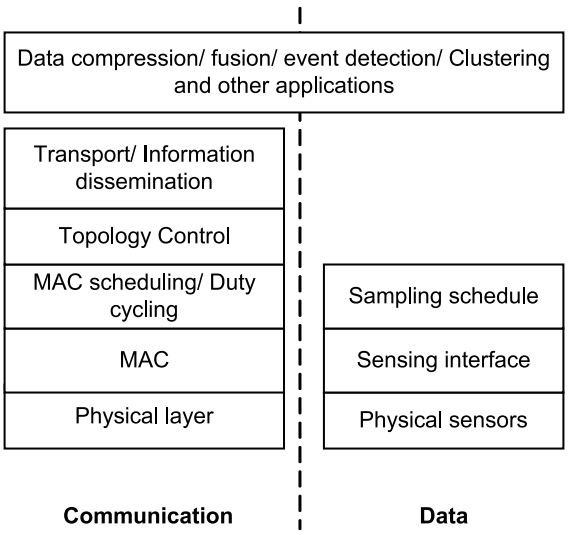



Figure 2.2: Example protocol stack for WSN

- *Physical layer*: IEEE 802.15.4 WPAN has suggested 3 frequency bands for WSN operations, 16 channels in the 2.4GHz ISM band, 10 channels in the 915MHz band and one channel in the 868MHz band. In general, the minimum output power required to transmit a signal over a distance $d$ is proportional to

$d^n$ where $2 \leq n \leq 4$ depending on the transmission environment. WSN normally has low lying antenna and near-ground channels and thus, $n$ is normally closer to 4 [PK00]. The higher frequency band relies mainly on line-of-sight paths. Multi-hop relaying in ad-hoc network can normally ease this problem as every node can serve as a router and nodes are separated by a short distance.

- *MAC layer*: There are a few competing MAC technologies. IEEE 805.15.4 suggested carrier sense medium access - collision avoidance (CSMA-CA) as the medium access technology. Nodes listen to the wireless channel to see if it is free for transmission. Other TDMA/ FDMA based MAC control has been proposed in [SCI+01] [Net06]. These schemes involve time and frequency slot allocation and have strict synchronisation and delay requirements. TDMA/FDMA based schemes achieve better energy efficient. However, these schemes are more complicated to implement and have stricter time synchronisation requirement and may not be achievable in all systems.

- *MAC scheduling*: MAC scheduling is a power saving scheme that puts the sensor nodes (or the transceiver module) to sleep in synchronisation, such that the nodes wake up and listen at approximately the same time (network-wide synchronisation is not usually necessary). The basis of MAC scheduling is network synchronisation. The nodes in the network, or at least in the neighbourhood must have the same notion of a clock to determine the timing of the scheduling cycle. The nodes are required to start the transmission within the wake cycle and hence, idle listening time is reduced. Proposed schemes include Sensor-MAC(SMAC) [YHE02], Wise-MAC [EEHDP04], Timeout-MAC (TMAC) [LKR04] and Dynamic Sensor-MAC(DSMAC) [LQW04].

- *Topology Control*: Topology control determines the transmission power of individual nodes and how they connect to the neighbouring nodes. The basic

idea is to form a connected network with satisfactory delay requirement while minimising the power consumption in the network. A throughout review and research on topology control is included in Part III.

- *Network and transport*: The networking and transport layer provides services for the information to pass across the network to other nodes, and to and from the sink. We consider the scenario where most traffic goes from node to sink, sink to node or node to neighbours (for synchronisation and algorithmic optimisation). Hence, the requirement of routing is not mandatory. Simpler transport mechanisms such as flooding, directional flooding and broadcast trees [HKB99] are preferred in most WSN networks, which also provide redundancies to the network. Gossiping [MTL88][WLP+02], directed diffusion [IGE+03], backbone building such as SPIN [HKB99], or simple routing such as EAR [SGAP00] are other commonly used strategies to disseminating messages in the network.

- *Data handling*: The data handling problem in WSN varies from application to application. For example, some applications require clustering for grouping the nodes that measure similar characteristics into the same clusters [WSSM05], data compression [GEH02] for minimising the amount of raw data required for transmission, data fusion for fusing data coming from different sensors, or event detection for detecting forest fire, landslide, etc. Some of these projects are reviewed in Section 3.1.

## 2.2 Challenges of Remote Environmental Monitoring

The work in this thesis focuses on a particularly type of sensor network application, *remote environmental monitoring*. This type of research can be very different from

other types of WSN applications including body sensing networks or in-building monitoring.

Traditionally data are logged locally by one of a variety of methods including a paper plotter or some form of on-site memory, and then manually downloaded at specific intervals to a central location for analysis. Pervasive wireless sensor networks have provided a cheaper, more robust and enhanced alternative to the data logging method. A group of sensors are deployed in the field and the data are transmitted via the some gateways to the user. Not only can the user get access to nearly-real time data from the sensor network, the group of sensors has also provided further spatial resolution for analysis. Moreover, the sensor network is autonomous and has built-in intelligence enabling it to alert the user when the data are different to what is normally observed.

One examples of these applications is habitat monitoring. The Center for Embedded Network Sensing (CENS) has several projects to monitor ecology and biological life forms [SOP+04] and other environmental applications [RSE+06]. Berkeley University also has a habitat monitoring project at Great Duck Island [SPMC04].

WSNs are also used to monitor and forecast environmental phenomenons. Such projects include the CORIE project which studies the Columbia river estuary [SBM+00], the NASAs project in Antarctica [PMR+05] for glacial monitoring and Volcano Sensorweb [SC07] for monitoring volcano eruption. In the UK, the Envisense centre, part of the Next Wave Technologies and Markets programme sponsored by the DTI, has three projects undertaking remote environmental monitoring applications in hazardous locations. FLOODNET aims to provide flood warning in the UK, SECOAS [SBW+03] monitors coastal erosion around small islands intended for wind-farms, and GLACSWEB [MHO04] aims to develop a system to work in a glacial environment. All of these may be transferable to other remote environments.

In addition to the common problems faced by all WSN applications, WSNs for

remote environmental monitoring have additional challenges due to the environment and location they are deployed in. Overall, the challenges for WSNs undertaking environmental monitoring can be summarised as follows.

- Power Management

  Most WSNs have a requirement of increased battery life-time. In remote monitoring applications, this becomes more important as the location may be very difficult to get access to, and hence, it is very difficult to change the batteries or deploy a new system.

- Standardisation

  At the moment there are very few standards governing the design and implementation of WSNs. IEEE has developed the 802.15.4 standards for low data-rate WSN applications, and Zigbee has additional items for routing and security measurements. However, WSN applications cover a very wide area and have very different communication and data requirements, which has made standardisation difficult.

- Scalability

  WSNs are required to be self-organising and self-healing, and must be able to cope with configuration and topology changes. Hence, a sensor network should be distributed. The size of a sensor network can range from a few nodes to hundreds to thousands of nodes and the nodes can be mobile. Hence, scalability is essential for WSN protocols.

- Security

  Security is necessary in WSN to protect the data gathered and malicious tampering to the system. However, as with most algorithms in WSN, the security system is required to be simple and distributed as sensor nodes have limited processing power and memory resources.

- Communication

  The systems are often operated in hazardous locations such as ocean, glacier or volcano. Radio communication can be a major challenge as the environment can vary significantly over time. Moreover, nodes can be damaged or destroyed, hence the communication protocol should be fast adapting to the changes of the network.

- Remote management

  As access to the deployment area can be difficult, it is important to have facilities for remote management to cater for policy changing or even software updating. The telemetry to transfer the data from the sensor network to the user has to be robust and reliable.

WSNs undertaking environmental monitoring have different characteristics, requirement and challenges compared to other applications. The algorithms developed in this thesis are intended to fulfil the objectives of environmental monitoring, and may not be suitable for other WSN applications including body sensing or intelligent building.

## 2.3 Modelling Wireless Sensor Networks

The wide range of sensor network applications have created big challenges for the research. The differences in applications imply that researchers can be dealing with very different WSN characteristics. In the case of modelling the networking topology, these include placement of nodes, mobility, infrastructure, deployment environment, etc.

Hence, we need to define a specific set of characteristics and requirements that our model and algorithms are intended to work with, on the other hand it is important to keep the algorithms as generic as possible such that they can be used in a wide range

of applications with small modifications. The models we have used are adapted from environmental monitoring, which has the following characteristics.

- The network has irregular/random node placement. We assume that for environmental monitoring purpose, nodes are either purposefully placed at locations which are best for monitoring, or they are randomly scattered to fulfil a density requirement. In most applications it is more likely that the nodes are specially placed for maintenance, controllability and health and safety issues. However, we have generalised the node placement as random in our simulation model. The deployment space can be 1-dimensional (along rail embankment), 2-dimensional (on a plain), 3-dimensional (on trees and posts).

- The node has little or no mobility. This is very different from body sensor application where nodes are attached to entities that are mobile. In environmental monitoring we deploy the nodes at a point and the nodes themselves do not have mobility element. However, the position of the nodes may still move depending on the environment it is deployed in, such as drifting in the sea or wind. The environmental changes may also cause changes to the network connectivity such as temporal unavailability of a radio connection.

- The model does not assume any infrastructure and most deployments are assumed to be outdoors and remote places. All nodes are equipped with the same, or similar hardware. The gateways in the network may possess more functionality, processing power and memory but we assume they have the same radio transceiver and have the same contribution to networking as other sensor nodes.

### 2.3.1 The Wireless Channel

The wireless channels in our simulated environment are represented by the *log-distance path model* for deriving the relationship of transmit $P_t$ and receive power $P_r$ and the distance $d$ between the nodes.

The log-distance model has been derived combining analytical and empirical methods. Empirical methods are based on field measurements and reverse curve fitting on the experimental data. The model can be adjusted to suit different transmission media by the parameter $\alpha$, which is called the path loss exponent or distance-power gradient. The path-loss model is [Rap02],

$$P_r(d) \propto \frac{P_t}{d^{\alpha}}. \tag{2.3.1}$$

The radio coverage region in this model is a disk of radius proportional to $\sqrt[\alpha]{P_t}$ centered at the transmitter. The value of $\alpha$ depends on the environmental conditions, and it has been experimentally evaluated in many scenarios [Rap02]. Table 2.1 summarised these values.

| Environment | $\alpha$ |
|---|---|
| Free space | 2 |
| Urban area | 2.7-3.5 |
| Indoor Line of sight (LOS) | 1.6-1.8 |
| Indoor no LOS | 4-6 |

Table 2.1: Values of the distance-power gradient for log-distance model in the different environments

There are limitations to the accuracy of the log-distance model to the actual environment as the log-distance propagation model predicts only the average received power at a certain distance. However, for our purpose the model is adequate to represent the relationship between transmission power and distance of communication, and for the comparison of the efficiency of different algorithms. $\alpha$ is set to 2 for all

the simulations in this thesis as the SECOAS environment is an open space in the ocean.

### 2.3.2 The Communication Graph

The communication graph defines the network topology, which includes the set of wireless links the nodes can use to communicate with each other. The presence of a wireless link is dependent on transmission power, the distance between the nodes and environment factors. The former two factors are accounted for in the log-distance model as described in section 2.3.1.

Sensor nodes are often equipped with omni-directional antennas because of their low cost. Nodes within the transmission range can be 'heard' in one hop and are *neighbours* to each other. In an asymmetric transmission environment where nodes are transmitting with different power, neighbour connections are often made symmetric by simply truncating all the asymmetric links.

We apply graph notation in representing our communication graphs. We denote $n$ as the number of sensor nodes located in a bounded space $[0, l]^{\delta}$, where $l > 0$ and $\delta = 1, 2, 3$ . Each node $u$ has a transmission range $r_u \in (0, r_{max}]$, where $r_{max}$ is the physical maximum transmission range of each node. The transmission range of node $u$ denotes the range within which data transmitted by $u$ can be correctly received. Note that in real radio equipment this transmission range is likely to be discrete with a number of steps. A node $v$ is a 1-hop neighbour of $u$ if it is within the communication range of $u$, and an edge $(u, v)$ exists from $u$ to $v$. If $u$ is also a neighbour of $v$, then the link is said to be bi-directional and the pair of nodes are symmetric neighbours.

A network with all the nodes transmitting at $r_{max}$ is a maximum graph. A graph with all the nodes transmitting with $r$ is called $r$-homogenous, or simply homogenous.

A homogenous graph is undirected[1].

Because of the irregular and random nature of the deployment of sensor nodes for environmental monitoring, we adopted a special case of *geometric random graph* (GRG) as our model. This assumes that nodes are randomly located in the space with uniform distribution and the nodes are connected if they are within the communication range of each other. In the study of sampling strategy in Chapter 4, the graph is homogenous and a *unit-disk graph* (UDG) may be used to describe the model. A UDG is a graph in which every two nodes are connected with an edge if and only if they are at distance of at most 1 unit. The unit distance in this case is $r$. The rules for node connectivity for our topology control models are more complicated than a UDG graph. However, the basic idea is similar.

These models have limitations representing the communication of a real sensor network. One of them is the assumption of perfectly regular radio coverage. The radio coverage varies largely by the environment (e.g. open air, forest, in-building, hill, sea etc) that the nodes are in and therefore the radio coverage region is likely to be irregular. A map based simulation model can be used if the actual deployment area is fixed. Moreover, the model assumes that nodes are distributed randomly in the field. In reality they may be distributed in a certain way to enhance network connectivity and communication. For example, one may set the minimum and maximum distance allowed between 2 nodes in deploying a WSN. These rules can be added to the models to make them closer to the actual deployment.

## 2.4 The SECOAS project

The work presented in this part of the thesis contributes to the Self-Organised Collegiate Sensor Network (SECOAS) project, one of projects funded by the UK Department of Trade and Industry (DTI) as part of the NextWave initiative. Its objective

[1] bi-directional/symmetric

is to investigate the use of self-organising sensor networks for environmental monitoring. SECOAS aims to achieve this by building a platform for distributed algorithms and emphasising on the development of the algorithms based on the spatial partitioning of the phenomena of interest. The target environments are by their nature turbulent and dynamic; the locations are not always safe and easily accessible. The combination of the need for an autonomous system and the limited power supply makes centralised management mechanisms unreliable and ineffective for such networks. Distributed and decentralised approaches observable in biological models show an adaptable style of network management suitable for the dynamic nature of the environments. The localised nature of algorithms will allow simple modification of individual algorithms without influencing other algorithms [SBW+03].

### 2.4.1 Project Introduction

The SECOAS sensor network was deployed at Scroby sands off the coast of Great Yarmouth and its purpose will be to monitor the impact of a newly developed wind farm on coastal processes in the area. A map of Scroby Sands can be found in Appendix A.

There is a need for a new monitoring approach for oceanography. Traditional methods for this kind of environmental monitoring involve using costly and immobile sea-bed landers that reside in fixed locations for long periods of time. Users are required to retrieve the equipment in order to download a large amount of data that give little spatial characteristics. This method has the following disadvantages.

- The high precision equipment is expensive and therefore, only one or a few can be deployed.
- The data collected lack spatial resolution.
- Data are collected in regular bursts. Interesting features may be missed in the

intervals. The burst frequency caters for worst weather so most of the time energy is wasted to collect data too frequently.

- It has a single point of failure. Data are collected at the end of the exercise only. If the equipment is lost or damaged by trawlers, effort can be wasted.
- Equipment needs to be recovered at the end of the data gathering exercise.

The SECOAS project provides an alternative solution by deployment of a large number of low-cost sensor nodes forming a sensor network. These nodes will provide for temporal and spatial resolution of data to the users. Data are transmitted via wireless radio links back to the user in regular intervals. The network will be optimised to lengthen battery life by adaptive scheduling of tasks as well as intelligent sampling strategy. The nodes are low-cost and hence can be disposable at the end of the exercise if preferred. The SECOAS project also serves as a test bed for the implementation of distributed algorithms.

### 2.4.2 System Design Overview

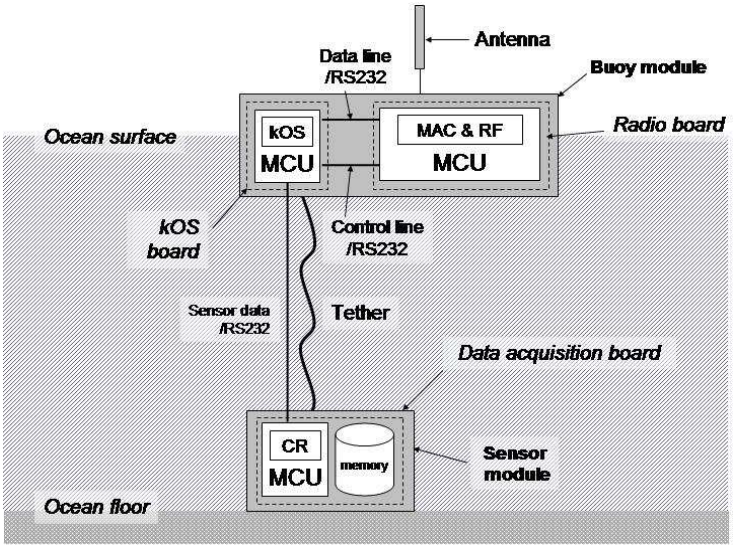



Figure 2.3: SECOAS sensor node architecture. MCU stands for micro-controller unit

A sensor node is physically divided into two parts, the sensor module that resides in the bottom of the sea, and the buoy module that floats on the surface. The sensor module houses a sensor micro-controller unit (MCU) that is designed for the operation of the sensors with averaging of data to remove excessive noise. The buoy module houses a radio MCU for radio communication and a MCU board with a kOS (kind-of Operation System) for the operation of intelligent algorithms. Figure 2.3 shows the main components in a SECOAS node. The buoy is tethered to the submerged sensor module which logs data locally. The sensor module passes data to the kOS module, which either directly relates the data, or processes the data and passes the results to the base station.

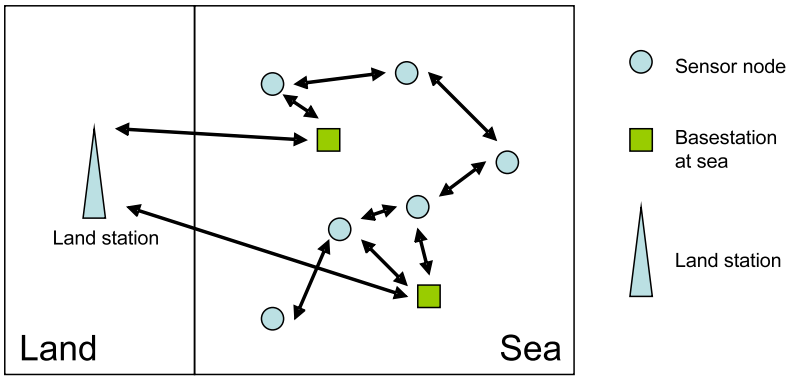



Figure 2.4: Network scenario of SECOAS

Figure 2.4 illustrates the SECOAS network architecture, where sensor nodes and base stations are shown. The sensor nodes and the base stations form an ad hoc network. The sensor nodes transmit their data to the base stations, which will then transmit the data back to the land station. The base stations are sensor nodes equipped with additional functionalities, batteries supply and have larger communication range. The data are then sent back to the users from the land station via a wired network.

The protocol stack for a SECOAS sensor node is depicted in Figure 2.2. kOS is a light-weight operating system that provides scheduling, parameter sharing, hardware

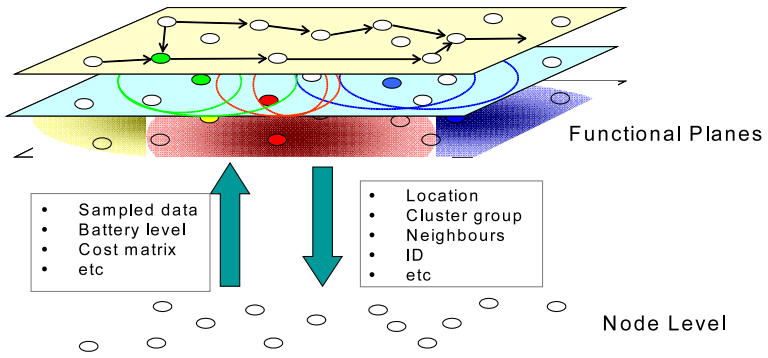



Figure 2.5: Layers of data handling aspects in SECOAS

interfaces and other facilities to the intelligent layer of the project. All the distributed algorithms are situated above the kOS layer. Vertically the protocol is divided into network related and sensing/ data related protocols. The 2 planes united at upper layer and the facilities provided by the lower layers are used by applications such as data fusion and location estimates, where spatial coordination is required in the protocol.

On the radio side of the project, Network synchronisation and MAC scheduling (sleep and wake cycles) are situated on top of the MAC layer for energy conservation. Nodes are only allowed to transmit during the wake-up period of a cycle. Above the layers, Topology Control is used for determining the transmission power of each node, for generation of a reliable network while minimising the node degree (number of neighbours within transmission range) and energy usage of the network. Networking and Transport strategy are on the upper layer of the communication stack.

For the data handling aspect of the project, temporal data processing uses data passed from the sensor module for compression in time and produces parameters for the clustering algorithm. The clustering algorithm also uses location information algorithm for estimating the location of sensor nodes. Spatial data processing, including data fusion and sampling strategy, sits on top of the clustering algorithms as it uses both the cluster information and temporally compressed data for operation.

The relation of the algorithms is depicted in Figure 2.5.

Figure 2.5 also represents the relationship between sensor nodes and task formations of the distributed algorithms, which are considered as separate functional planes in the network. Within a sensor node, individual algorithms communicate and exchange parameters with other algorithms. They are, however, separate entities in the network, which have functionalities and characteristics of their own. Each algorithm requires inter-node messaging to accomplish their task in the network.

The 3 types of information passing across the network are data, network policies and algorithmic messages. They are generated by different sources, have different destinations and hence, they should be distributed by separate transport mechanisms.

| Traffic type | Sources $->$ destination | Dissemination strategy |
|---|---|---|
| Data | Sensor nodes $->$ base station | Point-to-point/Gossip |
| Data | Base station $->$ land station | Point-to-point/Gossip |
| Network Policies | Base stations $->$ sensor nodes | Broadcast/Gossip |
| Algorithmic messages | Sensor nodes $->$ sensor nodes | One-hop broadcasting |

Table 2.2: Types of information and dissemination strategies

Table 2.2 illustrates the types of information in the network and their distribution strategies. Sensor data, results, alerts and reports are originated from the sensor nodes and destined to the base station. Although point-to-point communication, such as routing, is more suitable for this type of traffic, gossip can achieve the task and is simpler to implement as the nodes are not required to have a knowledge of the network structure. Network policies have multiple recipients and therefore, a broadcast or gossip strategy is suitable for the distribution of this information. Algorithmic messages are often sent to the direct neighbours only. Hence, simple one-hop broadcasting is adequate for this type of messages.

Information dissemination in WSNs requires a decentralised and scalable solutions. Gossip protocol is chosen in SECOAS for distributing both data and policies

in the network through a sequence of delayed communication events. The nodes forward messages to a subset of their surrounding neighbours that are chosen probabilistically [LM99]. Gossip protocols are ideal for systems with weak consistency because delays in propagating new messages are acceptable [SBW+03] and the eventual message delivery to all the nodes is guaranteed regardless of the network size or the prospect of node failure, as long as the final network is connected.

For the maintenance of the network-wise parameters that are common to all the nodes, a hash table is used in each node for keeping the latest parametric values and policy. The hash table has an entry for every parameter. A hash function [2] is then used to calculate the hash value based on the entries in the hash table. Hash values are exchanges between neighbours periodically. When a node receives a different hash value from its own, it sends out its entire hash table in the next transmission. A node which receives a hash table will compare the time attached to every entry and update its entries with the newer ones.

For example, in our proposed algorithm Basic Adaptive Sampling Scheduler (BASS), which will be discussed in Chapter 4, parameter $k_1$ is used to adjust the minimum sampling rate in nodes. It is a network-wise policy to be set by the user and has an entry in the hash table. If the user decides to change the value after the WSN has been deployed, the new value will be gossipped in the network and the hash entries for $k_1$ will be overwritten. However, in BASS the parameter $d_i$, which measures the reading difference between neighbouring nodes, is not be included in the table because $d_i$ changes with each reading and is not the same for all the nodes.

The use of hash table is good for maintaining parameters that do not change very often, and it is guaranteed that the node will receive the latest parameters eventually if the network is fully connected [WLP+02].

[2] A hash function can be any equation that uses the hash table entries to calculate the final hash value. For example, hash function can be the sum of all the bytes in the hash table.

### 2.4.3 Algorithmic Development in SECOAS

SECOAS is a collaboration project involving multiple parties working on different areas of the problem including radio, networking, data handling, sensors and the physical casing of the equipment. The Advanced Communications Systems Engineering (ACSE) group at UCL is responsible for the design of kOS, data handling and node positioning of the project (See Figure 2.2). The group aims to build an intelligent system with the following characteristics and contributions.

- Use of distributed algorithms concept

  Distributed algorithms are adopted in this project because a global knowledge of the system is not required and it allows for easy addition, removal and alteration of individual algorithms [SWA+04]. It makes the integration of algorithms in the system a easier task.

- Clustering approach to feature extraction

  The objective for clustering in WSN is to find interesting groupings of the sensor readings such that strategies can be applied to individual group. Since the interesting groupings of features would change with time, the strategies need to be adaptive and dynamic.

- Use of biologically inspired algorithms

  SECOAS applies complex system concept for algorithm development, where simple rules lead to global desirable results. Many characteristics of complex systems are analogue to the sensor network environment as only local knowledge is available to a node and cooperation of the nodes are required to achieve a global objective. Many biological entities are very simple and have very similar characteristics to a distributed system, that the results can be compared and applied to developing algorithms [SBW+03].

## 2.5 Chapter Summary

In this chapter, we discussed the general characteristics and different compositions of wireless sensor networks (WSN) and included an overview on the challenges of WSN in remote environmental monitoring applications. The modelling of communication channel and geometric random graph (GRG) was then introduced. We then described the SECOAS project with a system overview, description of the communication protocol used and research interests of the project.

# Part II

# Data Handling in the SECOAS Environmental Monitoring Project

# Chapter 3

# Data Analysis for Wireless Sensor Network

**Nomenclature used in this chapter**

| symbol | description | equation | page no. |
|---|---|---|---|
| $\alpha$ | parameter for long range dependency | 3.2.1 | 48 |
| $B$ | backward shift operator used in fractionally differenced process | 3.2.10 | 50 |
| $B_H$ | fractional Brownian motion | 3.2.6 | 49 |
| $C$ | channel capacity | 3.5.1 | 89 |
| $C_j$ | wavelet coefficients | 3.5.2 | 90 |
| $C_j^s$ | scaling coefficients | 3.2.13 | 54 |
| $D$ | fractional dimension | – | 51 |
| $\delta$ | long range dependence parameter used in fractional differenced process | 3.2.10 | 50 |
| $\epsilon_t$ | white noise with mean zero and variance $\sigma_\epsilon^2$ | 3.2.10 | 50 |
| $\eta$ | degree of freedom for Chi-square distribution | 3.3.2 | 61 |
| $H$ | Hurst parameter | 3.2.5 | 49 |
| $H_E$ | Entropy | 3.5.1 | 89 |
| $j$ | wavelet octave/dyadic scale | | 53 |
| $k$ | wavelet translation factor | 3.2.12 | 54 |

| symbol | description | equation | page no. |
|---|---|---|---|
| $N$ | number of data or number of wavelet coefficients | – | 54 |
| $N_{comp}$ | number of compressed coefficients | 3.5.5 | 93 |
| $N_j$ | number of wavelet coefficients at octave $j$ | – | 52 |
| $\psi$ | mother wavelet | – | 52 |
| $\Psi$ | frequency transform of wavelet analysing function | – | 62 |
| $r$ | number of vanishing moments | 3.2.2 | 52 |
| $R$ | Range in R/S statistics | – | 56 |
| $S$ | rescaled factor in R/S statistics | – | 56 |
| $S_x$ | spectral density function of sample X | 3.2.1 | 48 |
| $S_X$ | Autocorrelation of sample X | 3.2.2 | 48 |
| $\tau$ | Time lag | 3.2.2 | 48 |
| $\tau_j$ | wavelet scale at octave $j$ | – | 62 |
| $x(\cdot)$ | instance of (discrete) time series | | 54 |
| $X_t$ | time series $X = x(t)$, $t = 1, 2, ....n$, $n$ is total number of samples | 3.2.6 | 49 |

The requirement for long battery life has promoted the work on *in-network processing* in the data handling area of WSN. Instead of just sending a large amount of raw data in the network, data are interpreted and processed. The network only needs to send back the processed information, reports and alarms to the user, which is a very efficient way of conserving energy for long-distance communication.

The major challenge of data handling is that it can be very specific to applications. The data characteristics, goal of data handling and technique applicable can vary a lot for different type of applications, for example, oceanographical monitoring and animal target tracking. Thus, it is very important to understand the specific requirement for the application.

In this chapter, we report on the data analysis work done on Scroby Sands

oceanographic data. Firstly, a review of WSN data handling projects, self-similarity, wavelets and the tools used for time series analysis is included in Section 3.2. In Section 3.3, we evaluate 3 different estimators for self-similarity detection in data in the presence of trends and periods. This is to prepare for the use of the estimators for our analysis in latter sections. Section 3.4 is a thorough time series data analysis on our oceanography data, which includes temperature, conductivity, sediment concentration and pressure. The results of the analysis have formed the foundation of the proposed techniques using wavelet for raw data compression in Section 3.5. The analysis results also assist in the performance analysis of the adaptive sampling scheduler in Chapter 4. In Section 3.5, we propose a wavelet threshold compression technique for the sediment concentration data and the performance is evaluated.

## 3.1 Related Projects

We review the data handling requirements of some sensor network projects and summarised them in Table 3.1. The emphasis of different types of applications are also given in the table.

In the tracking and event detection applications, data are processed in the network and only the results are sent back to the user. Data collection applications aim to gather data for post-processing off-site by the users. Sampled data stored in the sensor nodes are sent back to the user when queried or at regular intervals. Most academic projects including, EYES [EYE] and SCADD [SCA], Centre for Embedded Networked Sensing (CENS), have both in-network processing and relaying of raw data in their research, however, they put emphasis on different aspects of intelligence.

SECOAS is a data gathering project and places emphasis on multi-resolution data collection in relation to quality. The sea environment makes it a challenge for both hardware and software development since all algorithms must cope with the communication constraints of under various weather conditions.

| | Application | Emphasis | Applications | Projects |
|---|---|---|---|---|
| 1 | Tracking | - Target identification<br>- decision fusion<br>- localisation<br>- multi sensor fusion<br>- association | - Military surveillance<br>- habitat monitoring | WINS |
| 2 | Event detection | - multi-sensor fusion<br>- adative sampling<br>- data compression<br>- event-detection<br>- reporting | - Building monitoring<br>- traffic feedback<br>- fault detection<br>- condition-based monitoring<br>- forest fire detection | WINS<br>XYZ<br>DSSN<br>DARPA |
| 3 | Data collection | - Query processing and dissemination<br>- Multi-resolution fusion<br>- in-network aggregation<br>- adaptive sampling<br>- data compression | Data gathering for offline analysis | COUGAR<br>DIMENSIONS<br>TAG<br>SECOAS |

Table 3.1: Classification of sensor network applications

We have researched on some of the data gathering projects in comparison to the objectives and emphasis of the SEOCAS project.

### 3.1.1 TAG

TAG [MFH02], which stands for Tiny AGgregation, is a service built for operating in the TinyOS environment. It aims to provide core aggregation services by the system software for the user. TAG functions similarly to the aggregation functions of Structural Query Language (SQL) of a database. It contains two phases: a distribution phase in which aggregate queries are pushed down into the network, and a collective phase, where the aggregate values are continually routed up from children to parents. Aggregation functions are pre-defined in the system and users can query the network using one of these functions. Examples of aggregation functions are maximum, min-

imum, count, sum average, histogram (example of query: SELECT AVG(volume), room FROM sensors where floor = 6 GROUP BY room EPOCH DURATION 30s).

TAG operates as a database query service. It differs from our data fusion goal in many ways. Firstly, SECOAS regularly sends data back to the base station without user query. Secondly, data compression, or fusion in SECOAS is done autonomously and statistically without user intervention. The user merely defines the qualities of the data. This contrasts to TAG where the aggregation functions used are selected by the user. The objectives of the two services are different.

### 3.1.2 LEACH

LEACH stands for Low-Energy Adaptive Clustering Hierarchy [HCH00]. It is an adaptive clustering algorithm designed for the use in MEMS-based sensor technology. Its principle features are adaptive clusters and rotating cluster-heads allowing the energy requirements of the systems to be distributed. It works on the notion of comparing the power usages of transmission, amplifying and reception of sensor nodes in the network using different topologies, and concludes that their clustering approach elongate network lifetime and has better results in sensor distribution when some of the sensors start to die out. The clustering algorithm allows each node to have an equal opportunity to become a cluster-head by setting a threshold value controlling the probability of becoming a cluster head. Further versions of work will include an energy-based threshold to account for non-uniform energy nodes.

LEACH is a very simple clustering algorithm; nonetheless it brings out the importance of clustering in energy conservation in sensor network. It differs from our approach in that clustering in LEACH is energy based while in SECOAS, clustering should be done based on the environmental characteristics. Moreover, in LEACH the cluster size is fixed and the cluster group is static, which contrasts to the dynamic clustering approach in SECOAS.

### 3.1.3 DIMENSIONS

DIMENSIONS [GEH02] is a general purpose query and mining scheme for in-network storage sensor network. Its specialities are: progressive aging of summaries in support of long-term querying in storage and communication-constrained network, tempo-spatial multi-resolution summary, and drill-down queries. Users can request for an overall summary with low-resolution first and then query for a detailed summary for the particular area and time of interest. It uses wavelet for both temporal and spatial summarization.

DIMENSIONS works on a database type sensor network with in-network storage, where users trigger the resolution of data delivery from the network by sending out a query. This is different from SECOAS in that data are sent back to the user at regular intervals and in-network storage is minimum. However, the principle in DIMENSIONS that user queries determine the resolution of the summary sent is similar to the SECOAS project where users define the qualities of data to be delivered. Moreover, both projects have the requirement for temporal and spatial compression. In DIMENSIONS, spatial compression is achieved by using wavelet, this however, would not be suitable for the SECOAS project since sensor nodes are irregularly placed and the routing of data cannot be guaranteed. The irregularity of temporal and spatial sampling issues in DIMENSIONS is addressed in [GGP$^+$03].

### 3.1.4 TARGET tracking

Several papers have suggested the use of Kalman filter and transferable belief model (TBM) for data fusion for the purpose of object recognition or target tracking [PMMM04][SR03]. These works have the common goal to fuse the data or decision coming from different sensors to obtain a final decision. Kalman filter is an optimal recursive data processing algorithm that takes into account the reliability of the measurements and predictions in producing the final results [May79].

Target Tracking have different objective to SECOAS in the way that processed data are required by the user. Sensor raw data are used to process some decisions such as object type, target identification and these decisions from different sensors are fused. In SECOAS, data are interpreted by the sensor network to reduce redundancy and irrelevancy statistically. The "decision" or ultimate analysis on the data are made by the scientists on-shore. Kalman filter or other decision functions may be used for policy or decision making in the network within the nodes in a cluster.

### 3.1.5 TinyDiffusion

TinyDiffusion is an attribute naming scheme [HSI+01] that replaces rather than augments the underlying networking routing layers. Its aims are to eliminate the overhead of communication required for resolving name binding and to enable activation of application-specific processing inside the network. The architecture of TinyDiffusion is based on three components: directed diffusion for disseminate information in the distributed system and data are managed as a list of attribute-value-operation tuples, matching rules for identifying when data has arrived at its destination, and intermediate filters for data processing. The user expresses an interest, which is an attribute-value pair and this interest disseminates into the network by direct diffusion, sensor nodes identify if they have the information by using matching rules. They may perform some in-network processes such as aggregation before relaying the information back to the user.

Attribute-based naming to address the data by its attribute, which is the user interest in this case, is an efficient concept to reduce network complexity and overhead of communication. Its use may be explored in the SECOAS project. The attributes of interest of a datum in SECOAS are location, time and metric. The efficiency and benefit of such a scheme instead of conventional IP type addressing can be investigated.

## 3.2 Literature Review

### 3.2.1 Self-similarity and Long-Range Dependency

Self-similarity is one of the characteristics that result from a Long-Range Dependency (LRD) or $1/f$ random process. A stationary LRD process has the characteristic of:-

$$\lim_{f \to 0} \frac{S_x(f)}{C_s \left| f \right|^{\alpha}} = 1, \quad -1 < \alpha < 0 \tag{3.2.1}$$

where $S_x$ denotes the Spectral Density Function (SDF), $f$ is the frequency, and $C_s > 0$ is a constant. $S_x(f) \approx C_s \left| (f) \right|^{\alpha}$ with $f$ approaches zero. An alternative definition stated in terms of Autocorrelation Function (ACF) is such that:-

$$\lim_{\tau \to \infty} \frac{S_{X,\tau}}{C_s \left| \tau \right|^{\beta}} = 1, \quad -1 < \beta < 0 \tag{3.2.2}$$

where $S_{X,\tau}$ denotes the Autocorrelation Function (ACF) and $\tau$ denotes the time lag. A stationary process has $S_{X,\tau} \approx C_s \phi^{\tau}$ where a LRD process has $S_{X,\tau} \approx C_s \tau^{\beta}$ for large $\tau$. $\phi$ and $\beta$ are some arbitrary constants. In both case $S_{X,\tau} \to 0$ as $\tau \to \infty$, but the rate of decay toward zero is much slower for a long memory process, implying that observations that are widely separated in time can still have a covariance that cannot be neglected [PW00] [Ber92], thus, long memory because the current observations retain some "memory" of the distant past. Another characteristic of LRD processes is that the sample variance decays slower than $1/n$. For a process with normal distribution, the sample variance $var(X_t)$ is related to the true variance $\sigma$ by

$$var(X_t) = \sigma^2 n^{-1} \tag{3.2.3}$$

The sample variance of LRD process decay much slower and is approximated as

$$var(X_t) \approx \sigma^2 c(S_X) n^{-\alpha}, \tag{3.2.4}$$

where $-1 < \alpha < 0$ and $c(S_X)$ is a constant derived from the autocorrelation function $S_X$. We explore some common stationary and non-stationary self-similarity models as follow:

**Fractional Gaussian Noise (fGn)** By definition, if $X_t$ is a fGn, then it is a stationary process whose Autocovariance Sequence (ACVS) is given by

$$S_{X,\tau} = \frac{\sigma_X^2}{2}(|\tau+1|^{2H} - 2|\tau|^{2H} + |\tau-1|^{2H}). \ \tau = ..., -1, 0, 1, ... \tag{3.2.5}$$

$\sigma_X^2 = var(X_t)$ is the sample variance and is a positive number. $H$ denotes the Hurst (or self-similar) parameter satisfying $0 < H < 1$. fGn can also be regarded as increments of Fractional Brownian Motion (FBM) $B_H(t) : 0 \leq t \leq \infty$ with parameter $H$, and

$$X_t = B_H(t+1) - B_H(t), \quad t = 1, 2... \tag{3.2.6}$$

$B_H(0) \equiv 0$; $B_H(t)$ for $t > 0$ is a zero mean Gaussian random variable with variance $\sigma_X^2 t^{2H}$; and the $covB_H(t), B_H(s) - B_H(t) = 0$ where $s \geq t \geq 0$. Discrete Fractional Brownian Motion (DFBM) can be created by cumulatively summing $X_t$:.

$$B_t \equiv B_H(t) = \sum_{u=0}^{t-1} X_u, \qquad t = 1, 2, ... \tag{3.2.7}$$

Details derivatives can be found in [PW00]. For small frequency,

$$S_X(f) \propto |f|^{1-2H} \tag{3.2.8}$$

Processes with $1/2 < H < 1$ have increasingly prominent low frequency components as $H$ increases, and we have $log(S_X(f)) \propto (1-2H)log(f)$.

**Pure Power Law process (PPL)** The discrete parameter process $X_t$ is a Pure Power Law (PPL) process if its SDF has the form

$$S_x(f) = C_s \left|f\right|^{\alpha} \tag{3.2.9}$$

where $C_s > 0$ is a constant. Note that in some literature, $S_x(f) \propto \left|f\right|^{-\alpha}$ is used

These processes can be stationary $-1 < \alpha < 0$ and non-stationary with $-1 \leq \alpha$ with stationary backward difference. For $\alpha = 0$, a PPL process becomes a white noise process with variance $C_s$. All PPL processes with $\alpha > 0$ a deficiency in low frequency components, which is reflected in the fact that $S_x(0) = 0$ [PW00].

**Fractionally Differenced process (FD)** A Fraction Differenced (FD) process is a stationary long memory process $X_t$ related to a white noise $\epsilon_t$ with mean zero and variance $\sigma_\epsilon^2$ through $(1-B)^\delta X_t = X_t - X_{t-1}$; $-\frac{1}{2} \leq \alpha \leq \frac{1}{2}$; $B$ is the backward shift operator and $(1-B)^\delta$ is

$$(1-B)^\delta = \sum_{k=0}^{\infty} \left( \begin{array}{c} \delta \\ k \end{array} \right) (-1)^k B^k \tag{3.2.10}$$

SDF is defined as [PW00]:

$$S_X(f) = \frac{\sigma_\varepsilon^2}{4\sin^2(\pi f)^\delta}, \qquad -\frac{1}{2} \leq f \leq \frac{1}{2} \tag{3.2.11}$$

For small $f$, $sin(\pi f) \sim \pi f$ and we have $S_X(f) \propto \left|f\right|^{-2\delta}$ approximately. FD is a stationary long memory process when $0 < \delta < 1/2$.

**Relation of parameters** fGn, PPl and FD processes are similar in that each depends upon just two parameters, one controlling the exponent of the approximating power law as $f \to 0$ ; and the other, the variance of the process (or the level of the SDF).

The *Hurst parameter* $H$ is particularly used to characterise fGn where $0 < H < 1$ indicates a stationary LRD fGn process. Although fGn is always stationary while

| Non-Stationary LRD process | Stationary LRD process | White noise | Stationary not LRD process |
|---|---|---|---|
| $H \geq 1$ | $1/2 \geq H \geq 1$ | $H = 1/2$ | $0 < H \leq 1/2$ |
| $\alpha \leq -1$ | $-1 < \alpha < 0$ | $\alpha = 0$ | $\alpha > 0$ |
| $\delta \geq 1/2$ | $0 < \delta < 1/2$ | $\delta = 0$ | $\delta \leq 0$ |

Table 3.2: Relation of $H$, $\alpha$ and type of random process

LRD processes can be stationary or non-stationary, $H$ and $\alpha$ are particularly used interchangeably in literature for any LRD process with the relation of $\alpha = 1 - 2H$ (Table 3.2). $H$ is also related to the *fractal dimension* $D$ by $D = 2 - H$.

### 3.2.2 Wavelets and Applications

Wavelets, meaning small waves, are mathematical tools for analysing time series or images. They cut up data into different frequency components and study each component with a resolution matched to its scale [PW00] [Gra95], while preserving the temporal characteristics. This is possible because wavelets are finite, or local in both time and frequency domain and allow us to analyse the signal in a specific time and frequency resolution as shown in Figure 3.1.

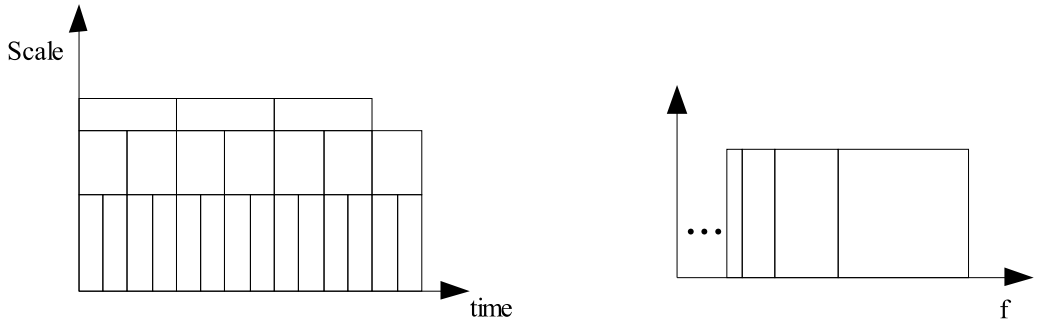



Figure 3.1: Partitioning of time, frequency and scale by wavelet analysis

Wavelets are functions that satisfy certain mathematical requirements and are used in representing the data in another form. *Scale* is an important concept in wavelet analysis similar to frequency in Fourier Transform. It represents the analysis resolutions for data processing. An *analysing wavelet* or *mother wavelet* is a

wavelet prototype function which can be dilated and shifted for temporal frequency analysis. This creates a wavelet series or a wavelet transform. Wavelet analysis has the advantages over transitional methods such as Fourier transform in that it preserves the location (time) information. For example, any discontinuity which exists in a time series can be located using wavelet analysis. The mother wavelet $\psi(\cdot)$ has the special property based on $\int_{-\infty}^{\infty} \psi(u)du = 0$ and $\int_{-\infty}^{\infty} \psi^2(u)du = 1$. This special characteristic means that the wavelet transform generated by dilation and shifting of $\psi(\cdot)$ is analysing the average of the difference of data within the wavelet intervals.

Wavelets are often characterized by *the number of vanishing moments*, which is defined as the number of moments $N_m = \int_{-\infty}^{\infty} t^m \psi(t)dt$ for $m = 0, 1, ...r-1$ equal to 0. The significance of vanishing moments stems from the fact that if $\psi(\cdot)$ and $r-1$ of its derivatives are continuous everywhere and satisfy certain bounded conditions, then $\psi(\cdot)$ has $r$ vanishing moments. Continuity of $\psi(\cdot)$ and enough of its derivatives helps eliminate artefacts in the analysis of a signal due to the wavelet function itself - loosely speaking, the more vanishing moments that $\psi(\cdot)$ has, the smoother is the transition from one approximation space to the next [PW00].

As shown in Figure 3.1, wavelet analysis partitions frequencies from the high frequency end, which is determined by the sampling frequency, towards the low frequency end. In each scale, the frequency width of analysis is halved. To avoid using an infinite number of scales, a low pass filter is used to cap the low frequency analysis. The iteration of the low pass filter generates the *scaling function*.

Some common wavelets and their definitions are given in Table 3.2[Mat04] (Wavelet toolbox user guide, [Res]):

**Discrete Wavelet Transform**

Similar to Fourier transform that analyses data in the frequency domain, Discrete Wavelet Transform (DWT) is used to analyse data in both time and scale domains.

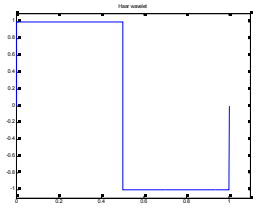
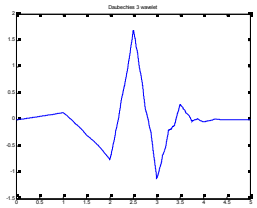
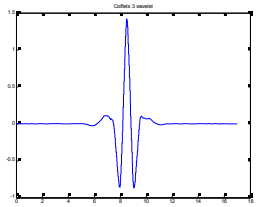
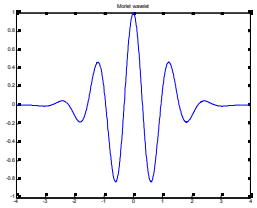
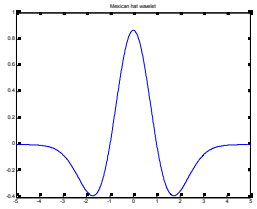
| **Wavelets** | **Definition and characteristics** | **Plot** |
|---|---|---|
| Haar function | - $\psi(x) \equiv \begin{cases} 1 & 0 \le x \le 1/2 \\ -1 & 1/2 < x \le 1 \\ 0 & Otherwise \end{cases}$<br>- $\psi_{jk}(x) = \psi(2^j x - k)$<br>- number of vanishing moment = 1 |  |
| Daubechies (dbN) | - Defined by its transfer function.<br>- Asymmetric<br>- Number of vanishing moments is N<br>- db1 is Haar function |  |
| Coiflets (coifN) | - Has *2N* moments equal to 0 for wavelet functions and *2N-1* moments equal to 0 for scaling functions.<br>- More symmetric than dbN |  |
| Morlet | - $\psi(x) = Ce^{-x^2/2}\cos(5x)$<br>- has no scaling function |  |
| Mexican Hat | - $\psi(x) = \int \frac{2}{\sqrt{3}}\pi^{-1/4}(1-x^2)e^{-x^2/2}$<br>- is proportional to the second derivative function of the Gaussian probability density function.<br>- has no scaling function |  |

Figure 3.2: Common Wavelets and their characteristics

DWT transforms the data in dyadic scale [Gra95], where wavelet scales are only defined for $2^{j-1}$ where $j = 1, 2, 3, \ldots$ denotes the wavelet octave [VA99]; and the samples in time are picked at locations that are multiples of $2^j$ . We usually associate the scale $2^j$ to frequency of $1/2^{j+1}$ to $1/2^j$ approximately [PW00]. There are $N_j \equiv N/2^j$ wavelet coefficients and the time associated with these coefficients are taken to

be $(2n+1)2^{j+1} - 1/2$, $n = 0, 1, ..., N_j - 1$, where $N$ is the total number of samples. For each scale, the product of the number of coefficients and their spacing in time is always $N$, which is the length of the original data. In DWT, the mother wavelet is dilated and shifted according to the equation:

$$\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k) \tag{3.2.12}$$

$k$ is the translation factor and the scale factor is only defined at $2^j$ interval, for $j = 0, 1, 2, ....$ What makes wavelet functions especially interesting is the *self-similarity* caused by the scaling and dilation [Gra95]. This is briefly discussed in section 3.2.2.

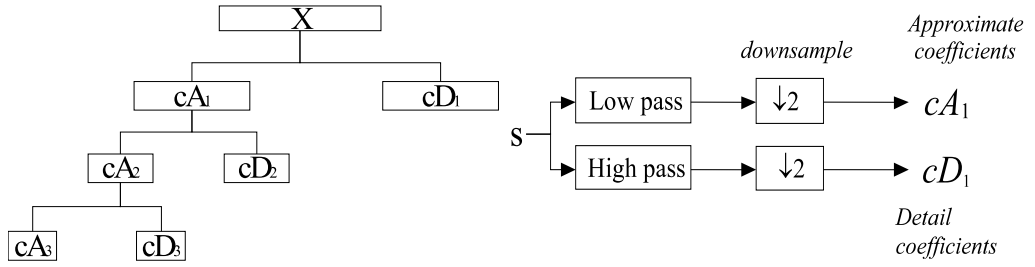
Each level of DWT decomposition gives two sets of coefficients, the *wavelet coefficient* and *scaling coefficients*. Wavelet coefficients represent the difference of averages across scales. Scaling coefficients serve as a low-pass filter and are proportional to the averages of the original data.

When the scale is chosen appropriately, scaling coefficients show the trend of data. Wavelet and scaling coefficients from DWT are calculated by the equations:-

$$\begin{cases} C_{j,k} = \sum\limits_t x(t)\psi_{j,k}(t) \\ C^s_{j,k} = \sum\limits_t x(t)\varphi_{j,k}(t) \end{cases} \tag{3.2.13}$$

where $\varphi_{x,\tau}(t)$ is a scaling function for calculating the scaling coefficients $C^s_j$, and $\psi_{j,k}(t)$ is a wavelet function for calculating the wavelet coefficients $C_j$. $x(t)$ is the time series for analysis.

The multi-resolution analysis (MRA) enabled by wavelets is advantageous because features such as discontinuities and extreme values may be localised in both time and frequency domains [PW00] [Gra95]. Wavelet compression can also be applied to different types of signals. Hence, the technique developed can be re-used in other sensor network projects.




(a) (b)

Figure 3.3: Wavelet decomposition of multiresolution analysis. (a) hierarchy of MRA. (b) retrieval of approximate and detail coefficients.

### Multiresolution Analysis

Multiresolution Analysis (MRA) uses wavelets to decompose a time series in time domain to different scales. In MRA the wavelet function is related to the impulse response of a high-pass filter and a low-pass filter is also defined. Figure 3.3 illustrates MRA, in each scale, the signal, or lower-scale approximations go through a low pass and a high pass filter to obtain approximate and detail coefficients. The coefficients are down-sampled so each higher scales would have half the number of coefficients of the lower scale [PW00][Mat04]. Overall, DWT includes the detail coefficients ($cD_j$) of each scale and the approximate coefficients ($cA_j$) from the largest scale.

MRA is used to analyse any special features across different scales of the time series. It is useful for extracting anomalies, discontinuity in the time series and identifying the scales where interesting features occur. For details of MRA and examples applications, please refer to [PW00][Mat04].

### Detection of Long Memory Processes

Self-similarity can be detected in the time domain or frequency domain in most stationary data sets. An example of time domain estimators is the rescaled range analysis, or $R/S$ statistics, where $R$ denotes the range given by the difference between

the maximum and minimum value of $Y_m = \sum_{i=1}^{m} |(x(i) - \bar{X_m})|$ over the first $m$ time steps, where $\bar{X_m}$ is the moving average over $m$ steps, and $S$ is the standard deviation of $X_m$. The relation of $R/S$ and $m$ is given by $R/S \propto m^H$ [Ber92][Spr03]. Detection of self-similarity in the frequency domain usually means least squares regression in the spectral domain. These methods work well with stationary data but not with data that have linear or periodic trend.

For this reason, wavelet transform has been explored for detection of self-similarity. The "analysis by scales" concept as well as the intrinsic self-similarity characteristic of wavelets mean that it can be a good tool for detecting the scale invariant self-similar processes. Abry has proved that wavelet estimator for LRD works well with linear trend and low-frequency periodic signal [AV98][APP93]. We look into the use of wavelet estimator to detection LRD in Section 3.3.

**Wavelet Compression**

The JPEG 2000 standard[1] has certainly brought a lot of attention to the research in wavelet compression. The attraction of using wavelet transform is that it decorrelates the data, allows multiresolution decomposition of the data and uses sub-band coding. Thus, sub-bands containing important information can be given more bits representing the data. Wavelet compression can be used in spatiotemporal compression in sensor networks. Example can be found in the DIMENSION project [GEH02].

Wavelet based compression can be achieved in different ways as listed below.

- *Compression by resolution.* Coefficients in small wavelet scales (e.g. $j = 1, 2$), which are the details of the signal, are only included in the compressed file with high resolution settings.

- *Compression by accuracy.* Coefficients with small amplitude below certain thresholds in each scale will be represented by zeros. Usually, it is accomplished

[1]JPEG 2000 official site: http://www.jpeg.org/jpeg2000/

by entropy coding such as Huffman coding or run-length zeros to achieve data reduction.

- *Identify regions of interest.* More interesting areas of images, or regions of time series are encoded with more bits compared to the background or less important areas or regions.

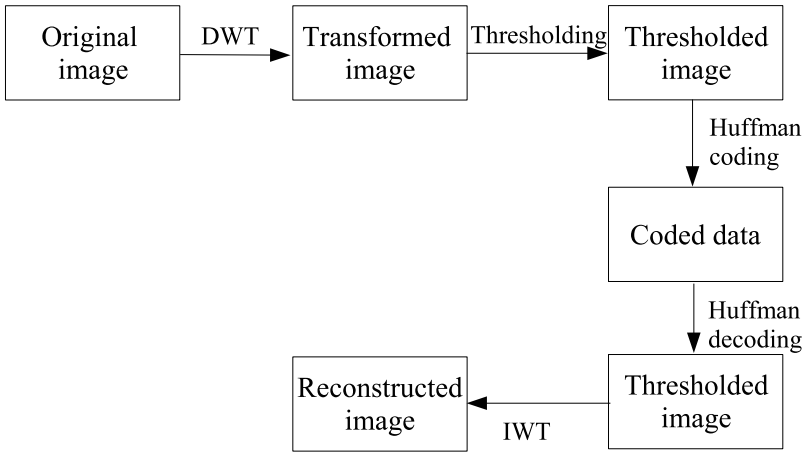



Figure 3.4: Procedures for image compression using thresholding. IWT stands for Inverse Wavelet Transform

An example of image coding procedure using wavelet thresholding technique can be found in Figure 3.4. We can explore wavelet based spatiotemporal compression technique for the use in the SECOAS project. JPEG2000 has given us an insight into how spatial compression of images can be achieved by using wavelet. The practicality and implementation issues will need to be investigated if a similar technique is to be developed for the embedded and distributed environment.

### 3.2.3 Time Series Analytical Tools

This section gives a review of the analytical tools used for time series analysis. We have especially evaluated the capability and limitations of various tools for determining self-similarity in the presence of trends and periods. This gives us an idea of the accuracy of the results.

### Trends and periods

The time series are plotted to give us a first insight on any trends, periodicity, non-stationary and anomalies present in the data. The data would be analysed in at least 2 time scales, a sample series from one of the hour data bursts (bins, see Figure 3.7) and the hourly mean of the bins. The hourly mean is calculated from the expected values in a bin (1024 data points) and it is plotted against hours. Short and medium-term behaviours are both observed.

### QQ-plot/ Probability Plot

In some cases histograms are plotted for the observation of the distribution of a random valuable. QQ-plot is used to aid the identification of the distribution. QQ-plot stands for Quantile-Quantile plot and is a graphical technique for determining if two data sets have a common probability distribution.

A QQ-plot is a plot of the quantiles of the first data set against the quantiles of the second data set. By a quantile, we mean the fraction (or percent) of points below the given value. For example, the 0.3 (or 30%) quantile is the point at which 30% percent of the data fall below and 70% fall above that value. A 45-degree reference line is also plotted. If the two sets come from a population with the same distribution, the points should fall approximately along this reference line. The greater the departure from this reference line, the greater the evidence for the conclusion that the two data sets have come from populations with different distributions [STA04]. For the verification of distributions the second set of data is taken from a theoretical random variable of the distributions to be tested, e.g. normal or lognormal. This is sometimes called a probability plot.

QQ-plot is used in this thesis in the wavelet compression section (Section 3.5) as one of the techniques to determine the compression thresholds. It is used to distinguish signal from noise, which is assumed to be Gaussian in the case.

**Spectral Analysis**

Frequency domain analysis is useful to analyse a signal for its periodicities and assist identification of the type of random process present in the data. Fourier transform is used for extracting harmonics of different frequencies. When applied to stationary processes, the frequency domain representation is called a periodogram and the spectral density is analysed. The definitions of periodogram, DFT and spectral density are [BD96][Mat04]:

DFT $X$:

$$X(k) = \sum_{t=1}^{N} x(t) e^{-2\pi(k-1)\frac{(t-1)}{N}} \tag{3.2.14}$$

Periodogram $I$:

$$I_t(\omega) = \frac{1}{N} \left| \sum_{t=1}^{N} x(t) e^{-jt\omega} \right|^2 \tag{3.2.15}$$

Spectral density $S$:

$$S(\omega) = I_t(\omega)/2\pi \tag{3.2.16}$$

$x(t)$ denotes the discrete time series, $N$ denotes the total number of samples, $\omega$ denotes the frequency in radians and it relates to the physical frequency $f$ (Hz) with $\omega = 2\pi f/f_s$, where $f_s$ is the sampling frequency.

Frequency spectrums are obtained by first applying Fast Fourier Transform (FFT) to the time series and then squaring the absolute values of the coefficients obtained from FFT. These values are then plotted against frequency to obtain a frequency spectrum. Spectral analysis on two different time scales is carried out because of the discontinuous nature of the data and the large frequency scales that are observed. Analysis on the hourly data in a bin (see Section 3.4 for the description of data) gives us an idea on short term frequency performance from seconds to minutes. The mean of hourly data form a series with the time scale in hours allowing investigation on the longer-term periodic behaviour.

**Wavelet Multiresolution Analysis**

The data analysis in this project utilises wavelet Multiresolution Analysis (MRA) for discovery of discontinuity and anomalies in the time series. Moreover, MRA is used for analysing the turbidity data to obtain the boundaries of values where the data follow a normal distribution. Description of Wavelets and Wavelet MRA can be found in Section 3.2.2.

## 3.3 Evaluation on Hurst Estimators for Self Similarity Process

In this thesis, we use three self-similarity estimators for estimating the Hurst parameters (see section 3.2.1) in our analysis, which are the periodogram, wavelet Least Squares fit Estimator (LSE) and wavelet Weighted Least Square fit Estimator (WLSE) as proposed in [VA99]. Since our data exhibit both linear trends and periodicity, the simpler form of time domain estimators including the R/S (Rescaled Range) statistics, correlogram and variance plot [Ber98][Ber92] have poor statistical performance, notably high bias and sub-optimal variance [AV98][Ber98]. This effect has been studied in many other research papers [AV98] [Haw99] [TT97]. We re-visit some non-stationary effects on the three estimators specific to our data characteristics in preparation for the data analysis section.

### 3.3.1 Periodogram

Periodogram (also see Section 3.2.3) is a tool for estimation of the Spectral Density Function (SDF), which can be used to characterise LRD. The raw estimator $\hat{S}_x$ is given by

$$\hat{S}_x(f) = \frac{\Delta t}{N} \left| \sum_{t=0}^{N-1} x(t) e^{-i2\pi f t \Delta t} \right|^2 , \tag{3.3.1}$$

where $N$ is the total number of data, $\Delta t$ is the time interval between samples and $x(t)$ is the discrete time series data. The raw estimator $\hat{S}_x(f)$ is a function of frequency and as $N \to \infty$,

$$\hat{S}_x(f) \stackrel{d}{=} \begin{cases} S_x(f)\chi_2^2/2, & for\ 0 < f < f_N \\ S_x(f)\chi_1^2, & for\ f = 0\ or\ f_N \end{cases} \tag{3.3.2}$$

where $\chi_\eta^2$ denotes a random variable with a chi-square distribution with $\eta$ degrees of freedom and the expected value $E(\hat{S}_x(f)) = S_x(f)$ [PW00][SM97].

Periodogram is an inconsistent estimator of SDF that its variance does not decrease with sample size. Smoothing technique is available to improve the performance given by

$$\bar{S}_x(f) = \frac{1}{2M+1} \sum_{l=-M}^{M} \hat{S}_x(f_{l-1}), \tag{3.3.3}$$

where $M > 0$, or modified estimator such as multitaper SDF and Burg method can be used [PW00][SM97].

Periodogram decomposes the process variance into different frequency components. This is similar to the concept of wavelet variance (see Section 3.3.2), which decomposes the process variance into different scales. We use the simplest periodogram without smoothing in this project as it is used as a comparison tool to the other two estimators.

LRD is defined such that $\lim_{f \to 0} \frac{S_x(f)}{C_s|f|^\alpha} = 1$ where $C_s > 0$ (see Section 3.2.1 Equation 3.2.1). A stationary long memory process has an SDF $S_x(f)$ such that $S_x(f) \approx C_S\,|f|^\alpha$ and $-1 < \alpha < 0$. A non-stationary LRD process would have $\alpha < -1$ (see Section 3.2.1). Therefore, we can estimate $\alpha$, or $H = (1-\alpha)/2$ by regressing $\log(\hat{S}_x(f))$ over $log(f)$ using least square estimate [Ber98][PW00].

### 3.3.2 Wavelet Least Square Fit Estimator (LSE)

This estimator is based on Discrete Wavelet Transform described in [PW00] (Section 3.2.2) for estimating a quantity called wavelet variance $\hat{v_x^2}$ for time series $X_t$, which

is defined as follow:

$$\hat{v}_x^2(\tau_j) = \frac{1}{N_j} \sum_{t=0}^{N_j-1} W_{j,t}^2 \tag{3.3.4}$$

where $W_{j,t}^2$ denotes the wavelet coefficients at octave $j$ or scale $\tau_j$, $N_j$ is the number of wavelet coefficients at scale $\tau_j$ .

When exploring the relation of wavelet variance and $\alpha$, we found two different expressions in literature which need to be clarified. In [PW00] p.297 it is stated that $\hat{v}_x^2(\tau_j) \propto \tau_j^{-\alpha-1}$ where $S_x(f) \propto |f|^\alpha$. This is derived from the assumption of $v_x^2(\tau_j) \approx \int_{1/2^{j+1}}^{1/2^j} S_Y(f)df$, which states that the wavelet variances are equal to the integral of the SDF across the wavelet scale. This however, contradicts the results in [VA99] that $\hat{v_x^2}(\tau_j) \propto \tau_j^{-\alpha}$ based on $E(\hat{v_x^2}(\tau_j)) = \int_{1/2^{j+1}}^{1/2^j} S_x(f) 2^j \left|\Psi_j(f)\right|^2 df$, where $\Psi_j(f)$ denotes the frequency transform of analysing wavelet at scale $\tau_j$. The variance is obtained by integrating both SDF and the frequency domain representation of the wavelet. There are two major differences in the definitions:

- The approximation in [PW00] is based on a perfect wavelet filter with cutting frequencies of $1/2^{j+1}$ and $1/2^j$, while [VA99] takes into the account of the frequency response of the wavelet filter.

- [VA99]'s estimate has a multiplication factor of $2j$. This is derived in [TT97].

These differences contribute to the discrepancy in the estimators. We have taken out several experiments to test the estimators with fractional Gaussian noise (fGn) with different Hurst parameters and it was found that $v_x^2(\tau_j) \propto \tau_j^{-\alpha}$ gives accurate results. Thus, the estimators, both LSE and Wavelet Weighted Least Square Fit Estimator (WLSE, see the next section) used in this project will be based on $v_x^2(\tau_j) \propto \tau_j^{-\alpha}$ and that $\alpha$, or $H = (1-\alpha)/2$ can be estimated by linearly regressing $log(v_x^2(\tau_j))$ on $log(\tau_j)$ using simple least square fit.

### 3.3.3 Wavelet Weighted Least Square Fit Estimator (WLSE)

This estimator is based on the same theory of DWT and wavelet variance as the LSE, except that weighted least square fit is used for regression instead of simple least square fit. The weight $S_j = (n \ln^2 2)/2^{2+1}$ is the inverse of the theoretical asymptotic variance of $\eta_j$ [VA99], which depends on the number of data available in each octave for calculating the wavelet variance and

$$H(j_1, j_2) = \frac{1}{2}\left[\frac{\sum_{j=j_1}^{j_2} S_j j\eta_j - \sum_{j=j_1}^{j_2} S_j j \sum_{j=j_1}^{j_2} S_j \eta_j}{\sum_{j=j_1}^{j_2} S_j \sum_{j=j_1}^{j_2} S_j j^2 - (\sum_{j=j_1}^{j_2} S_j j^2)} + 1\right] \tag{3.3.5}$$

### 3.3.4 Robustness of the Estimators

Experiments are performed to test the sensitivity and asymptotic behaviour of the estimators in the presence of a linear trend and periodic signal. Amplitude and frequency of the periodic signals in relation to fractional Gaussian noise (see Section 3.2.1 for definition of fGn, LRD and hurst parameters) are added to the test signal in the experiment.

Figure 3.5 shows the performance of the three hurst-estimators. (a) shows that in the presence of linear trend WLSE has the best performance overall. Both periodogram and LSE over-estimated the Hurst parameter where linear trend exists. While the periodogram shows consistent results, although incorrect, that are consistent with the presence of the trend, the LSE estimates increase in proportion to the size of the linear trend.

Figure 3.5 (b) and (d) show that the performance of the estimators in the presence of a periodic signal varies with amplitude. The diagram shows all the estimators fail to estimate the theoretical value when the amplitude of the signal is over 10 times (b) and 100 times (d) of the amplitude of fGn. At low amplitudes of the periodic signal

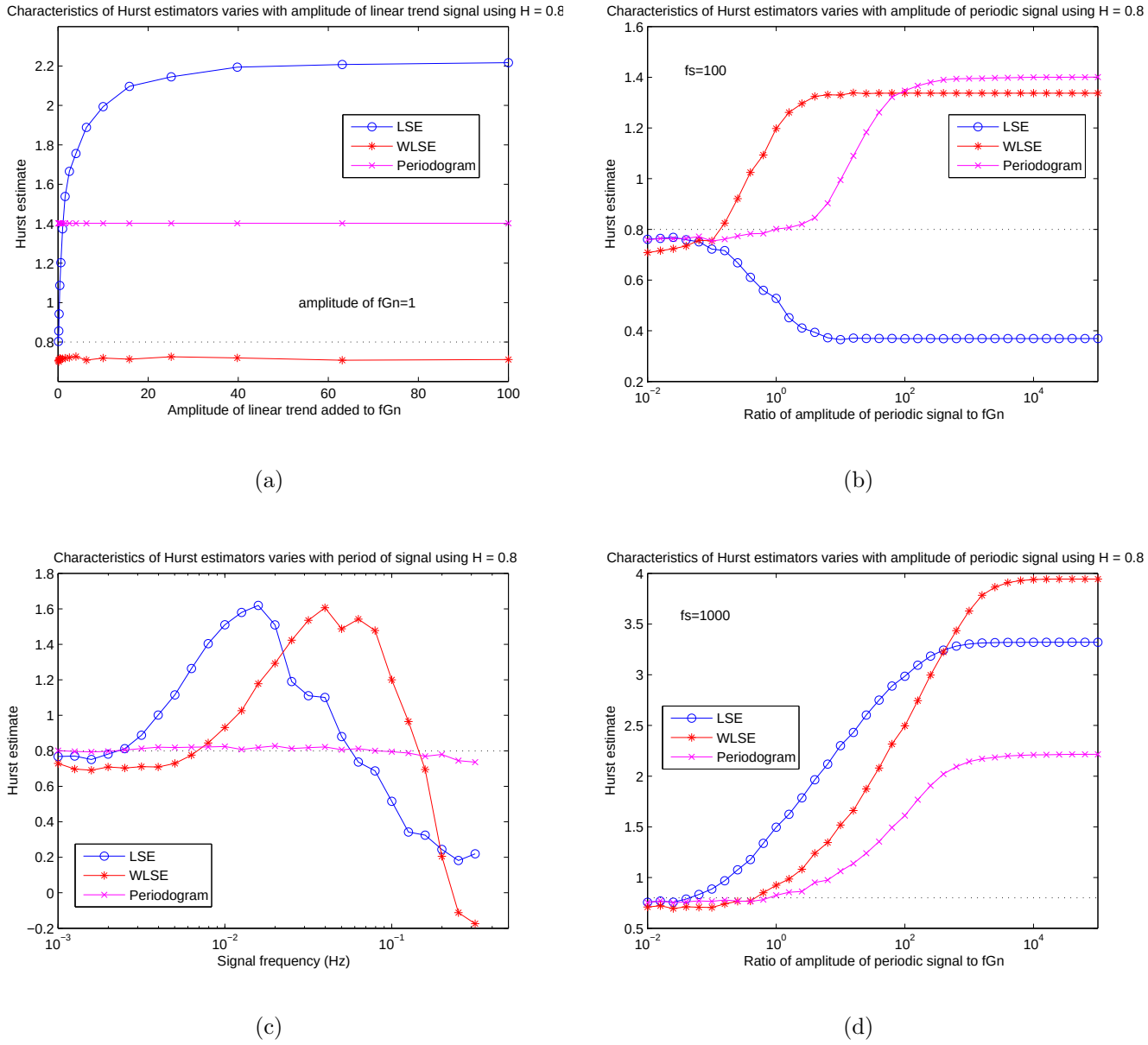



Figure 3.5: Experiment on hurst estimators on fGn noise, $H = 0.8$ and amplitude of fGn=1(unit). Testing of sensitivity to (a) amplitude of linear trend, (b) amplitude of periodic signal ($f_s = 100$), (c) frequency of periodic signal (amplitude of signal:fGn = 1:1), (d) amplitude of periodic signal ($f_s = 1000$)

in (b), the periodogram gives the best performance whilst in (d) WLSE shows better performance. It is observed that whether the estimators over or under-estimate the theoretical value depends on the frequency of the signal, which is explained in Figure

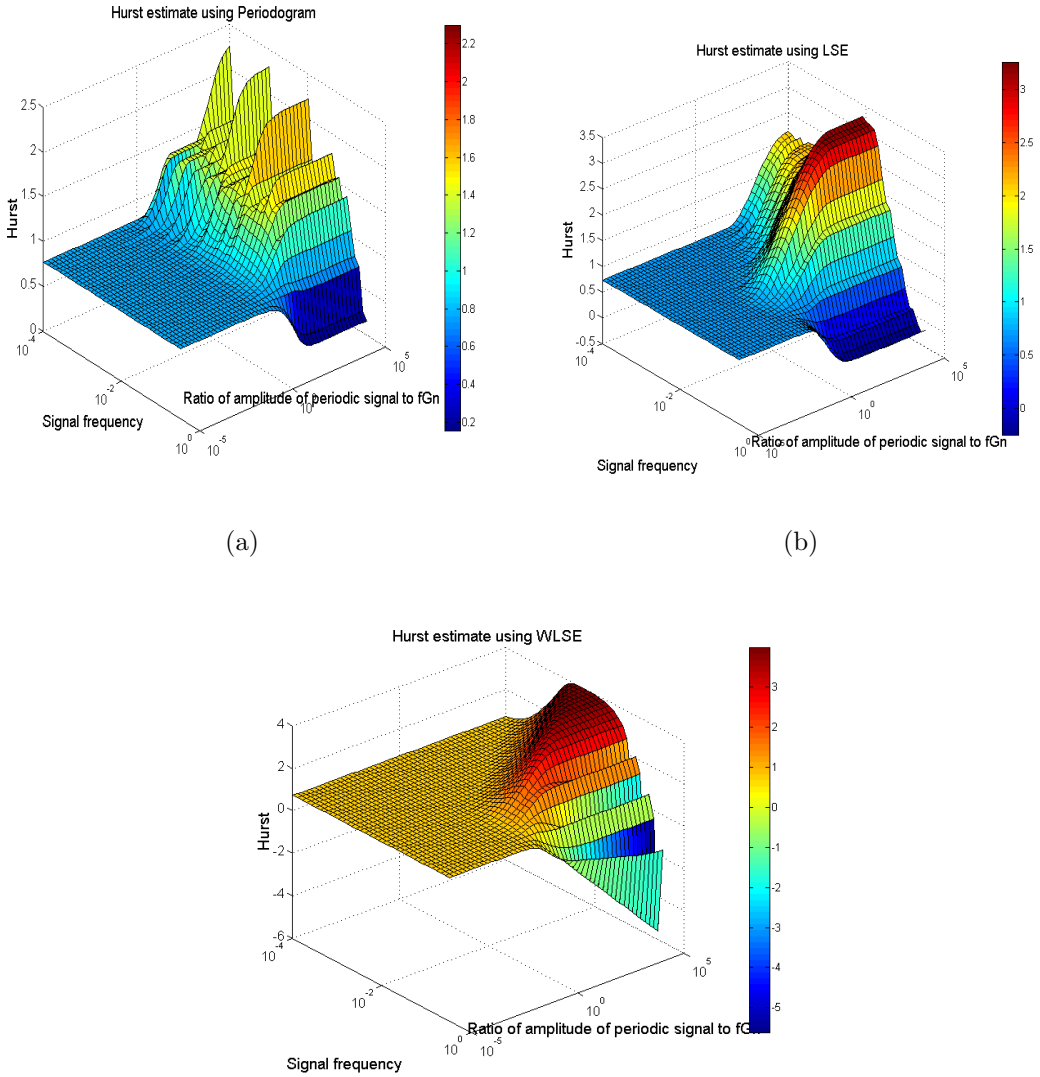
3.5 (c).

Figure 3.5 (c) shows the effects of the frequencies of the periodic signal on the Hurst estimation. In this experiment the ratio of the amplitude of the signal to fGn is set to 1:1. It is observed that if the signal frequency is a fraction of a 1000 times of the sampling frequency, both LSE and WLSE show better performance and are not affected by the presence of the signal.

So far we have evaluated the effect of the three hurst estimators based on one particular frequency (subfigure (c)), or a particular amplitude (subfigure (b) and (d)) of periodic signals. In figure 3.6, we evaluate the three estimators over a wide range of frequencies and amplitude periodic signals to check the asymptotic behaviour of the estimators. We can observe when each estimator fails to give accurate estimates.

In Figure 3.6 the 'safe regions', where estimators are robust, are investigated versus signal period and amplitude. In the graph this region is flat and is shown with the same colour throughout the region. In Figure 3.6(a), it is observed that the 'safe region' for periodogram estimator is actually independent on the frequency, but only depends on the amplitude of the periodic signal. The estimator is robust when the signal amplitude is equal to or smaller than the fGn amplitude.

As shown in Figure 3.6(b), LSE depends both on the frequency and amplitude of the periodic signal. The estimator is reliable when the amplitude of the signal is small and the frequency of the signal is low. In Figure 3.6(c) the case is similar that the WLSE estimator is more reliable for small amplitudes and low frequencies of the periodic signal. However, at low frequencies WLSE is more robust than the LSE that the estimator gives satisfactory results independent on the amplitude of the signal. WLSE also have a larger 'safe region' than LSE in this experiment. Hence, if we estimate that the amplitude of the periodic signal in the data is small (signal to noise ratio $< 1:1$), then Periodogram should be used as it gives good estimate independent of signal frequency. However, if we know that the signal frequency is




(a) (b)

(c)

Figure 3.6: Performance of Hurst estimates for fGn with H=0.8 with periodic signal amplitude and frequency using (a) Periodogram, (b) Wavelet least square estimate, (c) Wavelet weighted lease square estimate.

low, then WLSE should be used instead.

## 3.4 Analytical Results of Oceanographic Data

The objective for data analysis prior to algorithm development is to understand the characteristics of the data in the application such that appropriate strategies can be

applied. For example, statistics such as mean and variance can assist in setting the thresholds for event detection. In this section, we extract the important features of each parameter from our oceanographic data set in the available time scale. This prepares us for the development of the compression strategy in latter sections.

The test data for data analysis are obtained from the Wavenet project[2] at 3 locations around the Scroby Sands area (Appendix A). Location 204 is reported for the more complete analysis. Data were obtained from 23 April (13:00) to 9 June 2003 (09:00) which consists of the following measurements and their corresponding units.

Although there are 3 different locations of data available, the locations are too far apart for spatial analysis for the scale of SECOAS WSN, which has node separation between 20-100m. The parameters measured are,

- temperature (in $^{o}$C),
- conductivity/ salinity (in mmho/cm),
- pressure (in bar),
- turbidity/ Sediment level (in FTU / Formazine Turbidity Unit).

The data were gathered in bursts of 1024 points every hour at 1Hz frequency (a bin) for the 4 parameters measured, which is about 17 minutes of measurements. The system then slept for the rest of the hour (2576 seconds) and woke up afterwards for the measurement in the next hour. There are total 1126 hours of data. Figure 3.7 illustrates the time series of data collected.

The data are trimmed from bins 30 to 1110 due to the anomalies appear at the beginning and the end of data. A total of 971 bins is available for data analysis after trimming.

[2]Centre for Environment, Fisheries & Aquaculture Science (Cefas) Wavenet project http:\\ www.cefas.co.uk

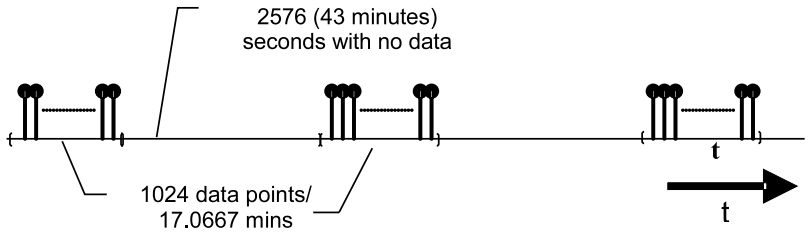



Figure 3.7: Characteristics of Wavenet data

### 3.4.1 Temperature and Salinity

Changes in temperature and salinity can increase or decrease the density of water at the surface, which can lead to convection; If water from the surface sinks into the deeper ocean, it retains a distinctive relation between temperature and salinity which helps oceanographers track the movement of deep water; Moreover, temperature, salinity and pressure are used to calculate density. The distribution of density inside the ocean is directly related to the distribution of horizontal pressure gradients and ocean currents. The distribution of temperature and salinity at the ocean's surface is influenced by heat fluxes, evaporation, rain, river inflow and freezing and melting of sea ice [Ste03a].

Salinity measures the total amount of dissolved material in relation to water. Its variability is very small. The salinity of most ocean water is from 34.6 to 34.8 parts per thousand, and thus, the definition of salinity needs to be accurate and practical.

Temperature is defined in Kelvin (K). However, Celsius ($^oC$) is also used and it relates to K by $K =^o C + 273.15$. The commonly used device found in the ocean is a platinum-resistance thermometer. Usually it is calibrated with an accuracy of a milli-degree (0.001°C) and the temperature scale itself has uncertainties of a few milli-degrees. The distribution of temperature at the sea surface tends to be zonal, which is independent of longitude. The deviations from a long-term average for temperature are small, less than 1.5°C except in the equatorial Pacific where the deviations can be 3°C [Ste03b].

We analyse the temperature and Salinity data by analysing the time series, periods and trends in the data. We also investigate if cross-correlation exists for the parameters.

**Time Series**

Figure 3.8 illustrates the time series of temperature and conductivity. We observe from 3.8 (a) and (c) that the two parameters are slow varying signals. The fluctuation between the two values is due to the precision of the measuring instruments. In our data, the rate of change for temperature in an hour is between 0 to 0.0007°C per 1024 seconds; the rate of change for conductivity in an hour is between 0 to 0.0006 mmho/cm per 1024 seconds. (b) and (d) show the hourly mean of temperature and pressure data of three months. Both graphs show an upward linear trend as well as periodicity. The shape of the hourly mean data of temperature and conductivity are alike.

**Frequency Domain**

In Figure 3.9 the periods of temperature and conductivity are captured. Fast Fourier transform (FFT) is used to produce the spectrums and the peaks are captured and translated automatically in Matlab. Periods of around 12 and 24 hours are present. The periods captured in the spectrums are slightly different than the actual ones because the spectrum is not smoothed. The downward shapes of both spectrums are due to the obvious linear trend observed in the time series.

**Parameter Correlation**

The similar shapes of temperature and conductivity in the time series may suggest that they are cross-correlated. Figure 3.10 shows the scatter plot for temperature and conductivity. The result of the linear regression is,

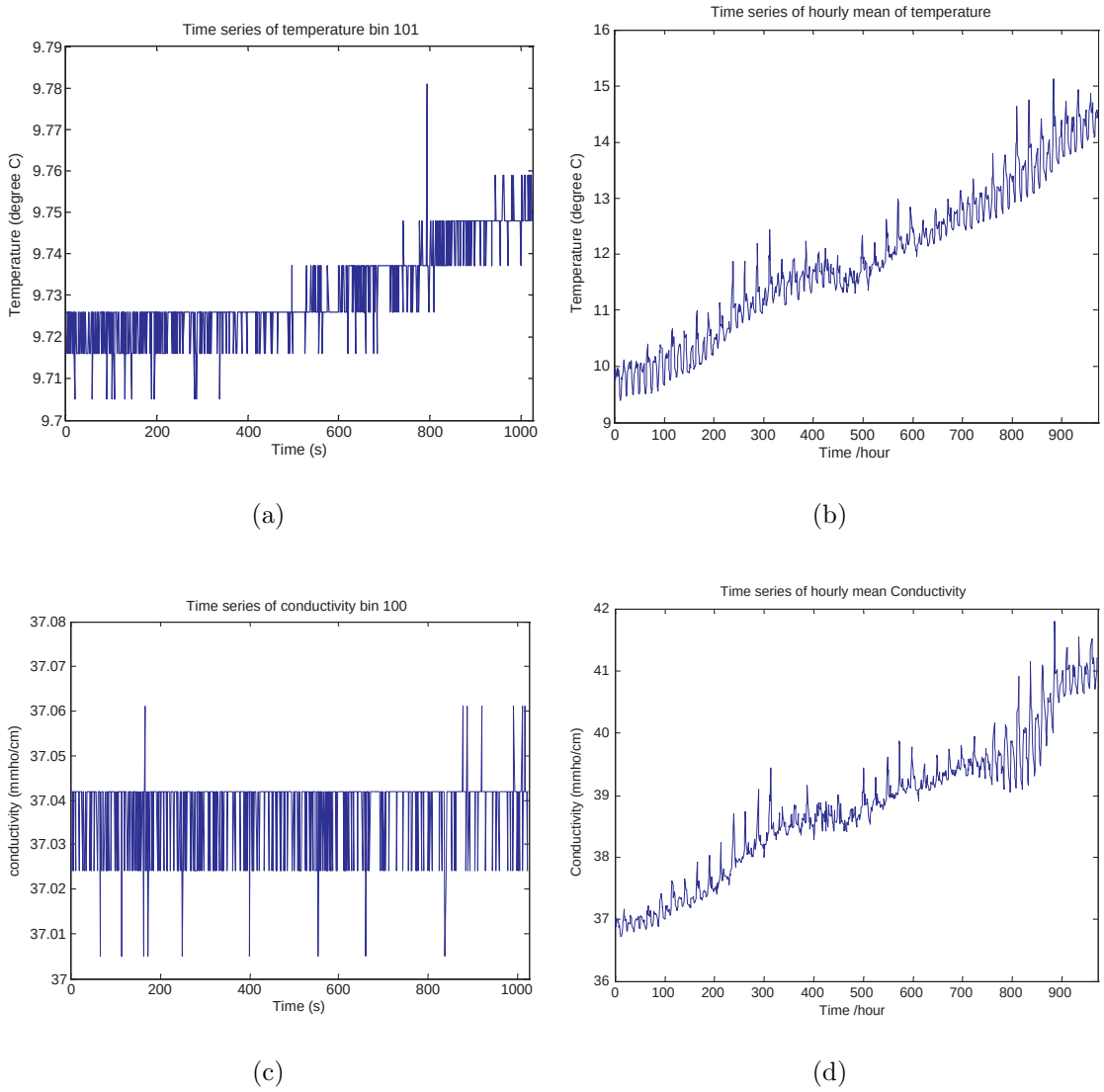



Figure 3.8: Time series of (a) temperature of bin 101, (b) hourly mean of temperature, (c) conductivity of bin 101, (c) hourly mean of conductivity

$$Conductivity = 0.84 * Temperature + 28.86 \tag{3.4.1}$$

Parameter correlation of the temperature and conductivity data may be used in data compression. For example, we can send only one set of the parameters together with the coefficients of linear regression instead of both sets of data.

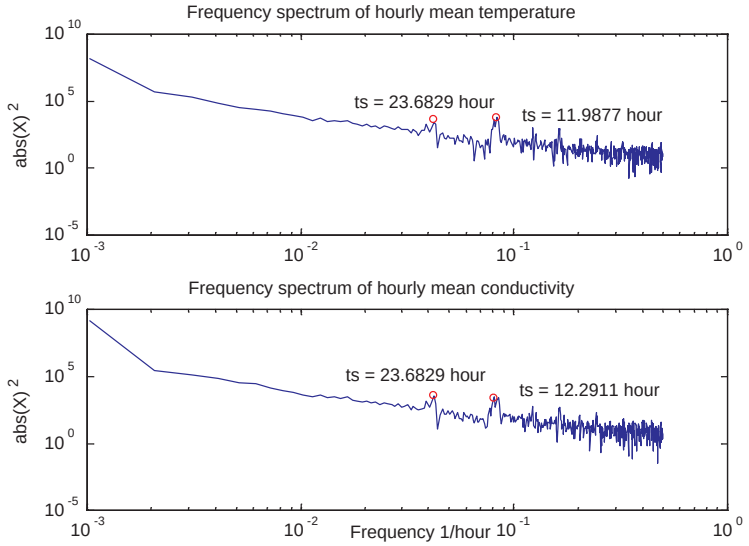



Figure 3.9: Frequency spectrum of temperature and conductivity

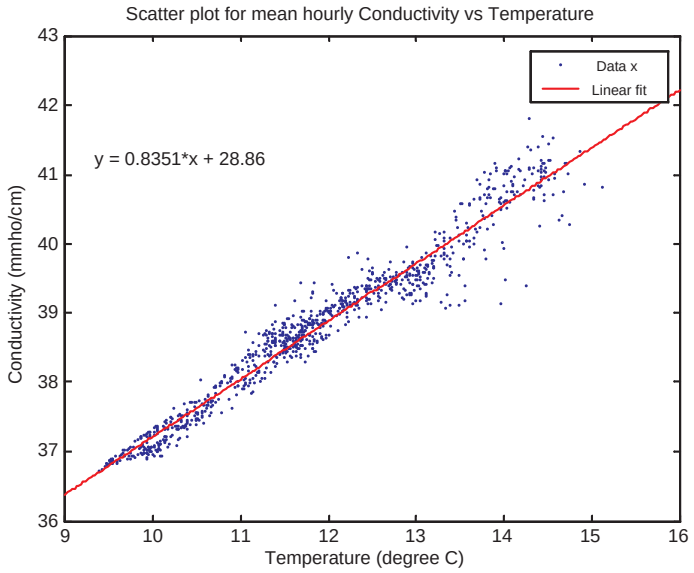



Figure 3.10: Correlation between Temperature and Conductivity

### 3.4.2 Pressure

Pressure is a measurement of sea-level and its variations are contributed by ([MH03]):

- Short-term temporal fluctuations in the height of the sea surface from waves and tides. Additional influences include changes of atmospheric pressure, wind-induced current along the coast, river runoff and ocean circulation caused by fluctuations in the oceanic wind field.

- Long-term temporal changes resulting from changes in the mass of the ocean,

and heating of cooling of the ocean

- Long-term Coastal residence
- Large-scale crustal movement.

Long-term variations refer to period over 50 years. SECOAS mainly focuses on the short-term variations, which are more related to the engineering aspect of the design.

The short term variation mainly consists of tides and waves. For tidal movement, the principle semidiurnal and diurnal tidal constituents with periods of 12.42 and 23.93 hours contribute to one of the regular periodicities of sea-level variations, and is dependent on the location in relation to the equator.

In most coastlines, waves represent the dominant source of energy in the nearshore zone and can be distinguished by regular and irregular waves. The motion of regular waves is periodic. Regular waves can be described in terms of wave height $H^w$, wave length $L^w$ and wave period $T^w$. Wave height is the difference in elevation between the crest and the trough of the wave, wave length is the distance between successive crests and wave period is the time it takes for the wave to travel a distance equal to its wave length. Wave-by-wave analysis is used for irregular wave and can be described using $H_s^w$(significant wave height), $H_{rms}^w$ (rms wave height), $T_z^w$,(mean wave period) and $T_s^w$(significant wave period). Wave spectrum can also be used to describe irregular wave, for which peak spectral period $T_p^w$ is used for description. The normal wind waves are irregular waves. Typical significant wave heights and periods along swell-dominated coastlines are 1-2m and 10s, respectively. Along locally-generated wind waves dominated areas, wave heights and periods are generally 0.5-1m and 3s respectively. During storms, wave heights can be substantially larger and wave exceeding heights of 10m can be found. Readers can refer to [ET01] for more details of tides and waves in relation to pressure measurement.

We analyse the pressure data by firstly looking into the time series. Analysing the frequency spectrum can help us extract the tidal periods in the data. We then introduce mean spectrum periods and spectrum width, which can help us to extract the wave feature in the data. We also investigate if self-similarity exist in the raw data itself and the standard deviation of the hourly data.

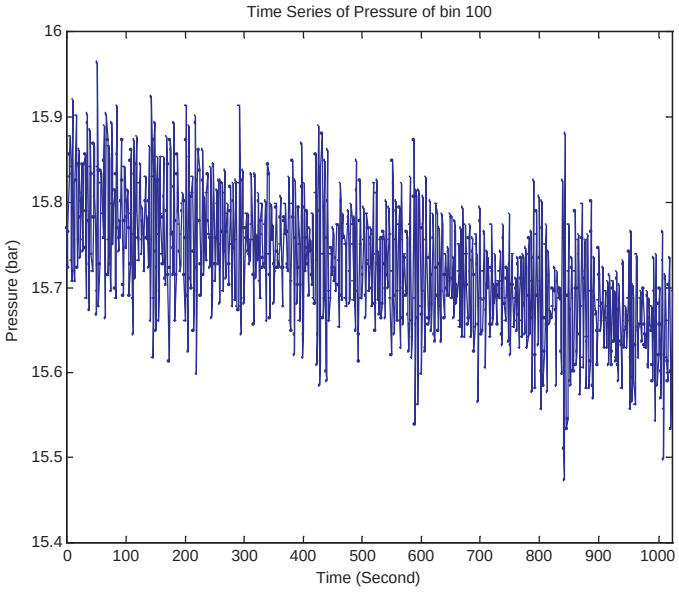
**Time Series**

Figure 3.11 shows the time series of pressure data across difference time scales. From (a), we can see that pressure data exhibit both changing mean and periodicity in the scale of seconds. The changing mean seems to be driven by a longer-term trend. The mean and standard deviation of the first half of bin 100 is 15.7640 and 0.0664 bar. The mean and standard deviation of the second half of bin 100 is 15.6909 and 0.0640 bar. The non-stationarity of the first moment is very small and the variance seems to be stable across this bin. In Figure 3.11 (c), we can observes the longer term periodicity due to tidal movement. (c) also shows a drifting upward mean for mean pressure data.

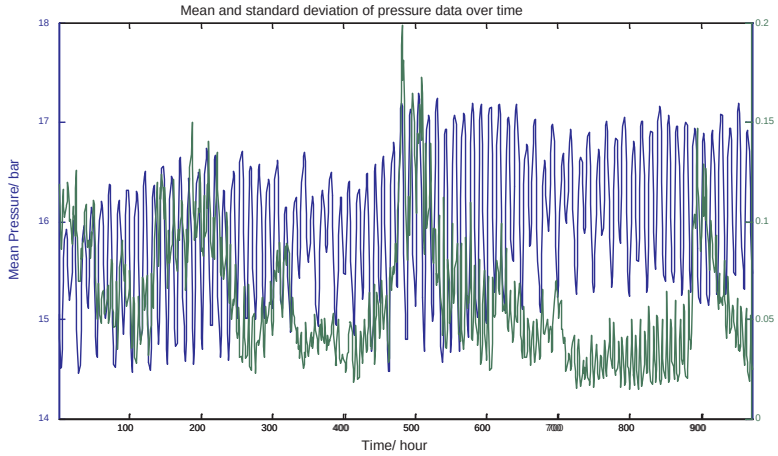
The short-term pressure variations are contributed by tides and waves. We can associate the mean pressure of a bin with sea level and tides, and the *variance or standard deviation* of a bin with wave height. The mean and standard deviation of pressure over time is shown in Figure 3.11 (c). The standard deviation of pressure shows periodic behaviours. However, it is not in synchronisation with the mean pressure data and it seems that there is no correlation between the two statistics.

The standard deviation of pressure data in a bin can be used to characterise the wave. For example, the variance around hour 500 is particularly high, which may be due to stormy weather. While the low variance period from hours 700-900 may represent calm weather.

The shape of the graph in Figure 3.11 (c) resembles a semi-diurnal tidal cycle.




(a)




(b)

Figure 3.11: Time series analysis of pressure data. (a) 1024 data in bin 100.(b) mean and standard deviation of hourly data of all bins.

For the definition of the tidal cycles, please refer to [MH03].

### Frequency Domain

The two time scales we investigate in the frequency domain of pressure shows very different characteristics. Fast Fourier transform (FFT) is used to produce the spec-

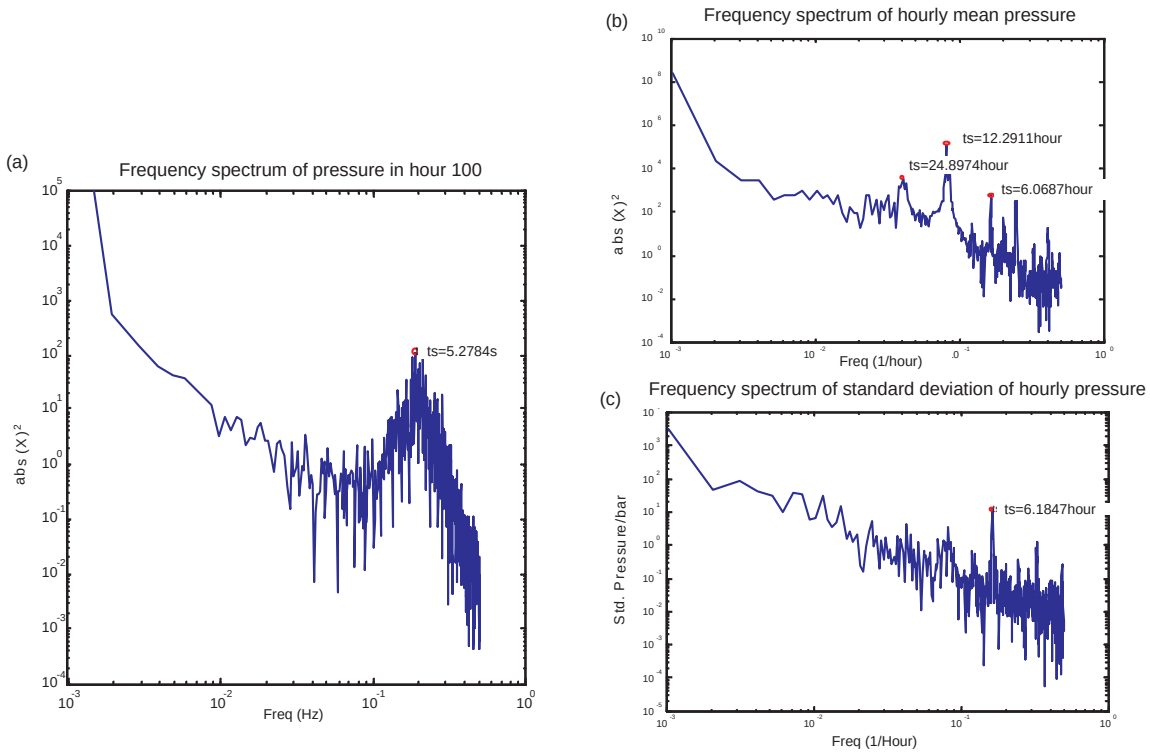



Figure 3.12: Frequency spectrum of pressure data for (a) hour 100. (b) mean of hourly pressure. (c) standard deviation of hourly pressure.

trums and the peaks are captured and translated automatically in Matlab. Figure 3.12 (a) shows the frequency spectrum for pressure data in hour 100. It is observed that rather than having a sharp peak, the dominant frequencies spread over a range of values. Figure 3.12 (b) shows the spectrum for mean hourly pressure data. A dominant period of 12.3 hours is observed while another two less dominant periods at 24.9 and 6 hours are visible. Figure 3.12 (c) shows the spectrum for the standard deviation of hourly data. The dominant period here is 6.2 hour. This is double the frequency of the period of mean pressure data.

It is also observed that, excluding the periodic characteristics, all three graphs show a downward linear trend of a $1/f$ (LRD) process. It is most clear in Figure 3.12 (c) as the periodic peak is of relatively weaker power than the $1/f$ trend. We will investigate this behaviour in the later section.

As wave spectrum is not simple sinusoidal, it is necessary to derive additional metrics for describing the short term frequency spectrum in order to understand the characteristics of wave. The NIWA Instrument System[3] has derived these quantities, mean spectral period and spectral width to characterise wave.

**Mean spectral period** describes the main period of the wave, and is calculated by the equation,

$$\bar{T} = 2\pi m_0/m_1 \tag{3.4.2}$$

where $m_n$ is the $n^{th}$ moment of the spectrum of $h(t)$, which is the time series of "hydrostatic water depth" related to the raw pressure time series $p(t)$ by,

$$h(t) = \frac{p(t) * F}{\rho g} + z_p - D \tag{3.4.3}$$

where $g$ is acceleration due to gravity, $\rho$ is water density and $F$ is a factor used to convert pressure in pounds per square inch to pressure in $kg/m^2$. $z_p$ is elevation of the instrument above the seabed. $D$ is a correction factor for the particular instrument.

The moment $m_n$ given by: $m_n = \int_{1/T_2}^{1/T_1} f^n S_h(f) df$. $T_1$ is the minimum wave period and $T_2$ is the maximum period. In this project, it is found to be 3s (Nyquist frequency) and 20s respectively. $S_h(f)$ is the SDF of $h(t)$.

**Spectral Width** describes the range of the wave period, and is calculated using the equation:

$$v = \mu_2/\bar{w}^2 m_0, \tag{3.4.4}$$

where $\bar{w} = 2\pi/\bar{T}$ is the angular frequency and $\mu_2 = (m_2 m_0 - m_1^2)/m_0$.

We analysed these quantities and the results are shown in Figure 3.13. It is observed that mean spectral period ranges from 4 to 12 seconds and mean spectral

[3]National Institute of Water & Atmospheric Research, http://niwa.cri.nz, technical note No. 98/5

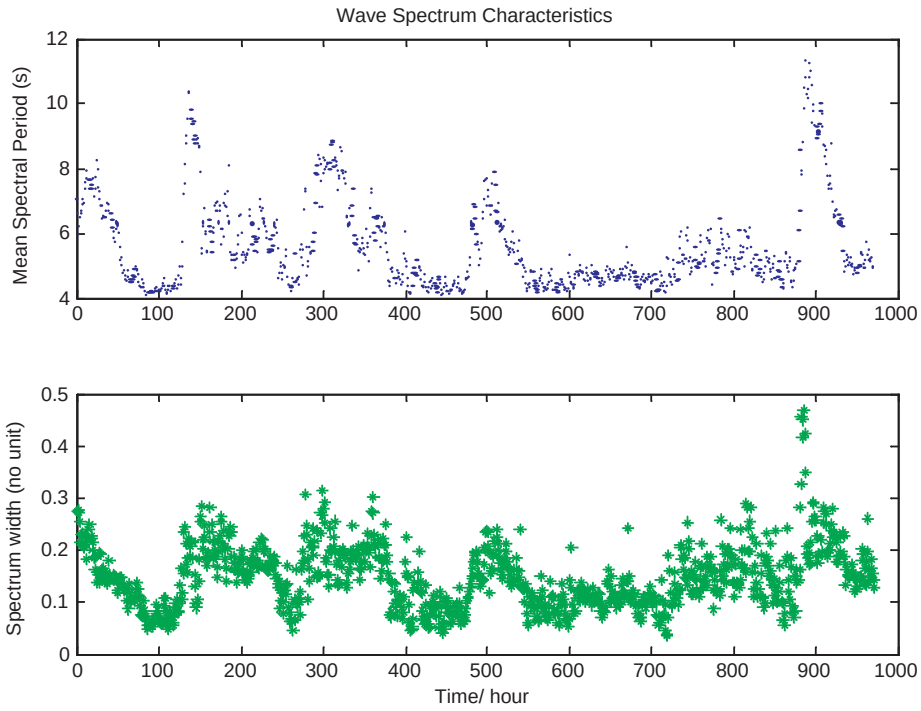



Figure 3.13: Mean spectral period and spectral width for pressure data

width is between 0 to 0.5. Note that there is no unit for spectral width. It is observed that mean spectrum period generally increases with spectrum width.

### Self-Similarity

**Estimates within an hour** Determining self-similarity in the pressure data is difficult as short-term and long-term periods are the dominant signals in the data. In Figure 3.14 all three estimators show inconsistent results. Since the results in section 3.3 show that we do not know the accuracy of the estimators unless we can acquire some background information regarding the amplitude and frequency of the signal in relation to the degree of self-similarity, we cannot conclude whether self-similarity is present in the short term pressure data.

**Standard deviation of hourly data** Previously, we saw that in Figure 3.14 (c), the standard deviation of pressure may display self-similarity characteristics. We further verify the Hurst estimates with periodogram, LSE and WLSE as shown

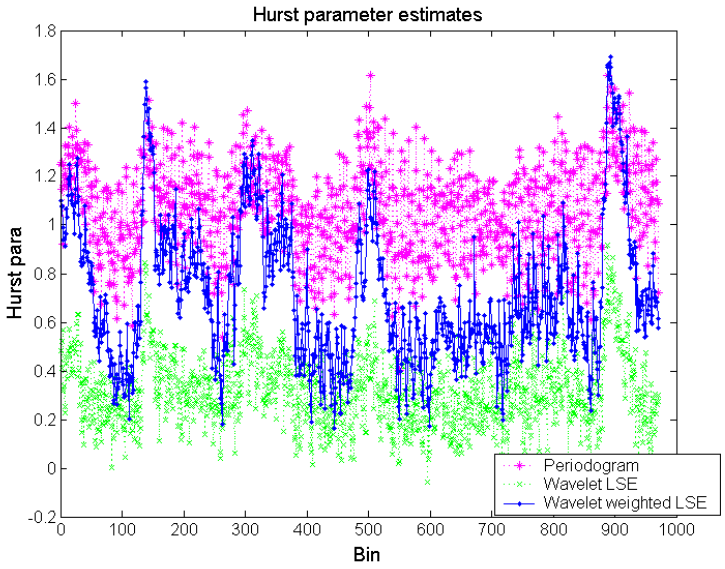


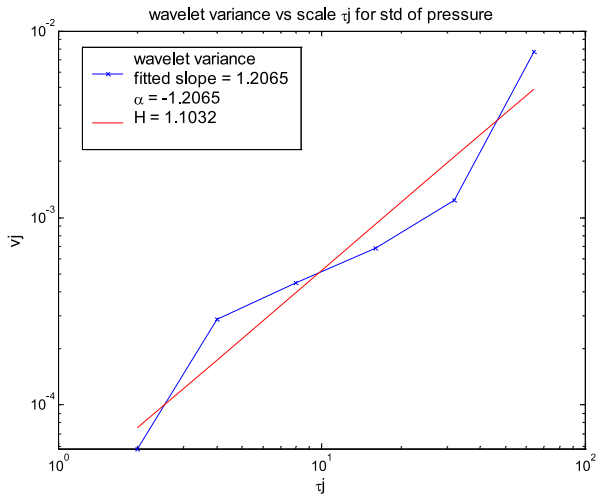

Figure 3.14: Hurst estimate using periodogram, Wavelet least sq estimate and Wavelet weighted least sq. estimate

in Figure 3.15 (a), (b) and (c) respectively. Although the three tools give slightly different results, they all indicate the result of non-stationary LRD data ($H > 1$). We can conclude that LRD is very likely to be present in the std. of hourly data.

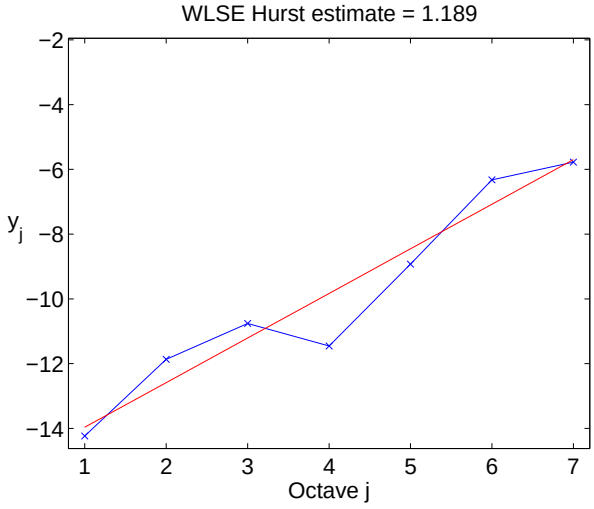
### 3.4.3 Sediment Concentration

Sediment, or precipitation refers to both organic and inorganic loose material that is moved from time to time by physical agents including wind, waves, currents and gravity. The sediments found in coastal environments can be either imported from external environments or locally produced [ET01].

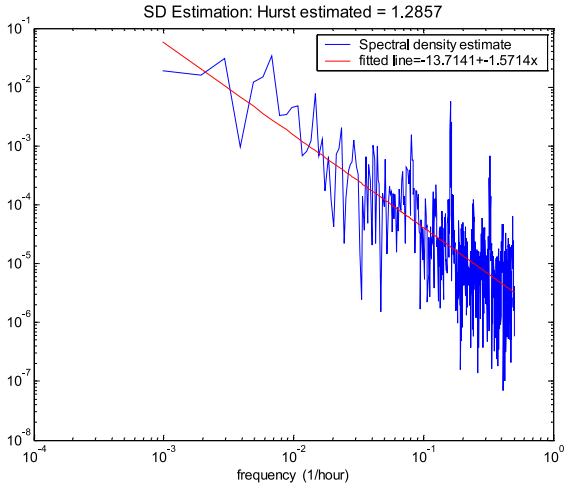
"Turbidity" or cloudiness of water is a relative term describing sediment level. It is an optical property depending on characteristics of the scattering particles, external lighting conditions and the instrument used. Turbidity is measured in Nephelometer units (NTUs) referenced to a turbidity standard, or in Formazin Turbidity Units (FTUs) derived from diluted concentrations of 4000-FTU formazin, a murky white suspension that can be purchased commercially. Since turbidity is a relative unit,




(a)




(b)




(c)

Figure 3.15: Hurst estimates for standard deviation of all bins (a) by LSE, (b) by WLSE, (c) by Periodogram.

calibration is a particularly important subject for the measurement.

We analyse sediment concentration by firstly looking at the time series, trend and frequency of the data. We then apply multi-resolution analysis on the data to break down the features into wavelet scales and extract important features in each

scale. Finally we look to see if self-similarity is present in the data.

### Time Series

It is observed that in the smaller scale in seconds within a bin in Figure 3.16 (b), there is no obvious periodicity. The mean of the time series is drifting and the variance is not constant and therefore, the series is non-stationary, which may be caused by either a longer-term trend or the series may have non-stationary LRD which has $\alpha < -1$. The mean and standard deviation of the first half of bin 100 is 30.1182 and 1.5518. The mean and standard deviation of the second half of bin 100 is 33.3931 and 2.2133, which prove the non-stationarity of both $1^{st}$ and $2^{nd}$ moments.

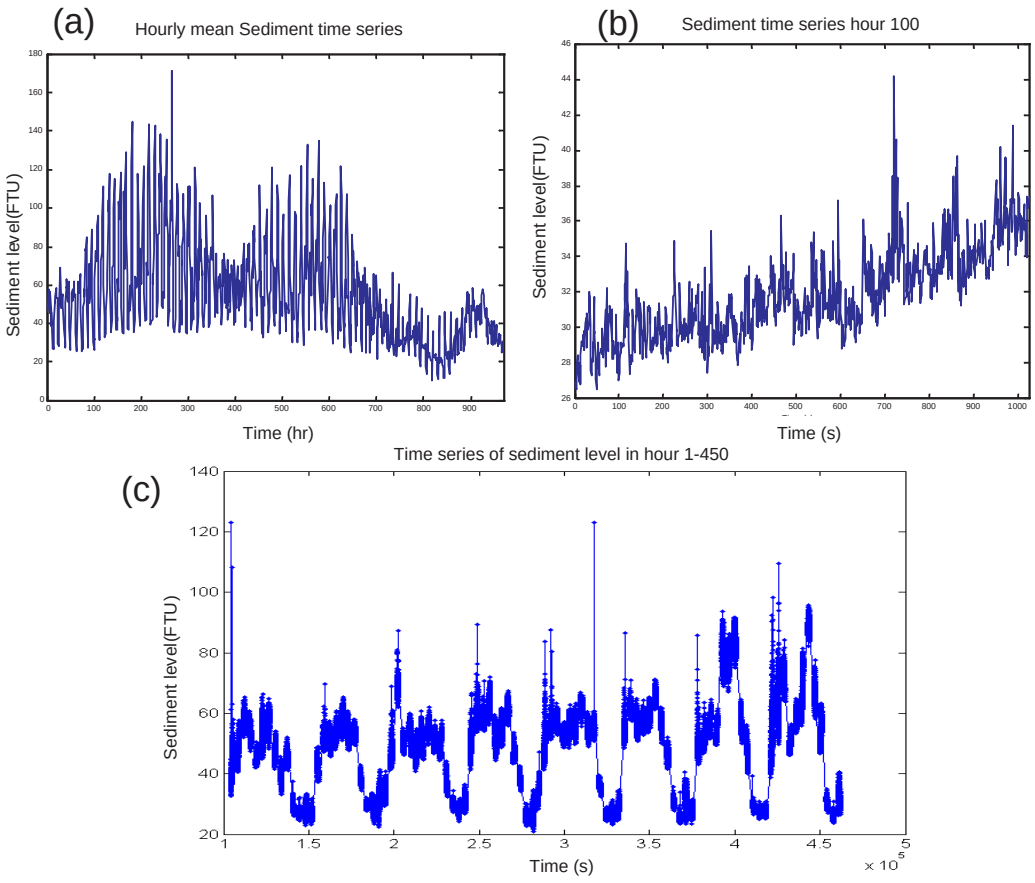



Figure 3.16: Figure 31: Time series analysis of sediment level. (a) mean hourly data of all bins. (b) 1024 data in bin 100. (c) bin 1-450.

In Figure 3.16(c), however, periodicity is observed. The periods exhibit an asym-

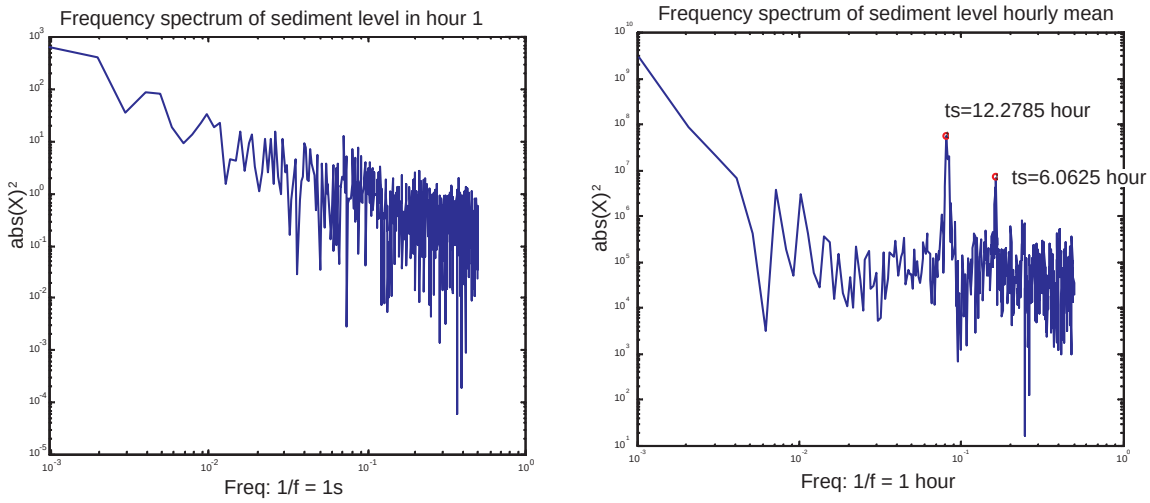



Figure 3.17: Frequency spectrum of (a) Hour 1. (b) hourly mean sediment level.

metric look in that a cycle stays longer at the top than the bottom values. In this scale the series appears to have a rather stable mean with varying variance. A few outliners are observed. Note the discontinuity of the data where there is a gap of 2576 data after every 1024 data points, which appears as a thin straight line in the graph. Figure 3.16 (a) plots the hourly mean of the 1024 data points in each bin. It further confirms the periodic behaviour observed before. The time series shows a drifting mean and varying variance overall. However, there is no obvious long-term linear trend to this scale of data.

**Frequency domain**

Figure 3.17 (b) confirms the periodicity observed in the sediment level. There is a sharp peak to the spectrum at 12.28 hours. A smaller frequency component appears at 6.06 hour. The strong power observed at the very low frequencies is due to the non-zero mean of the data.

There is no frequency peak in Figure 3.17 (a), which shows the frequency spectrum in bin/hour 1. However, it can be observed that the power decays with the frequency with a $1/f$ manner. This may be an evidence of the presence of long-range

dependency (LRD) in the data.

**Multi-Resolution Analysis**

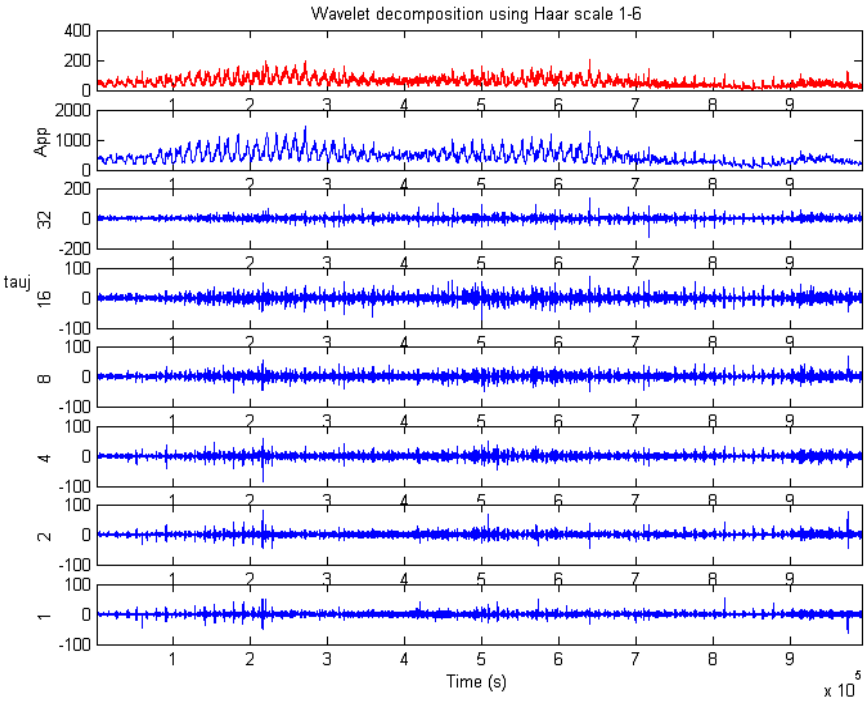



Figure 3.18: Wavelet decomposition octave 1-6.

Wavelet transform is used to decompose and analyse all data in different scales. Each octave $j$ represents a time scale $\tau_j$ of resolution $2^{j-1}$. Figure 3.18 shows detail decomposition of octave 1-6. Figure 3.19 shows detail decomposition of octave 10-14 of all the sediment concentration data.

In lower scale wavelet picks up the high frequency changes. In $\tau = 1$ *to* 32 (octave 1-6), we expect to see the discontinuity of the data within each bin at every 1024 data points. However, the discontinuities are not very obvious indicating the transition from bin to bin is not particularly obvious. Figure **??** (b) shows the larger scale wavelet decomposition from scale $\tau = 512$ *to* 8192 (octave 10-14). Note that octave 11 equals scale 1024, which is the analysis across the bins. Hence, from this scale on we can see the bigger picture of hourly change. And the shape and components of the

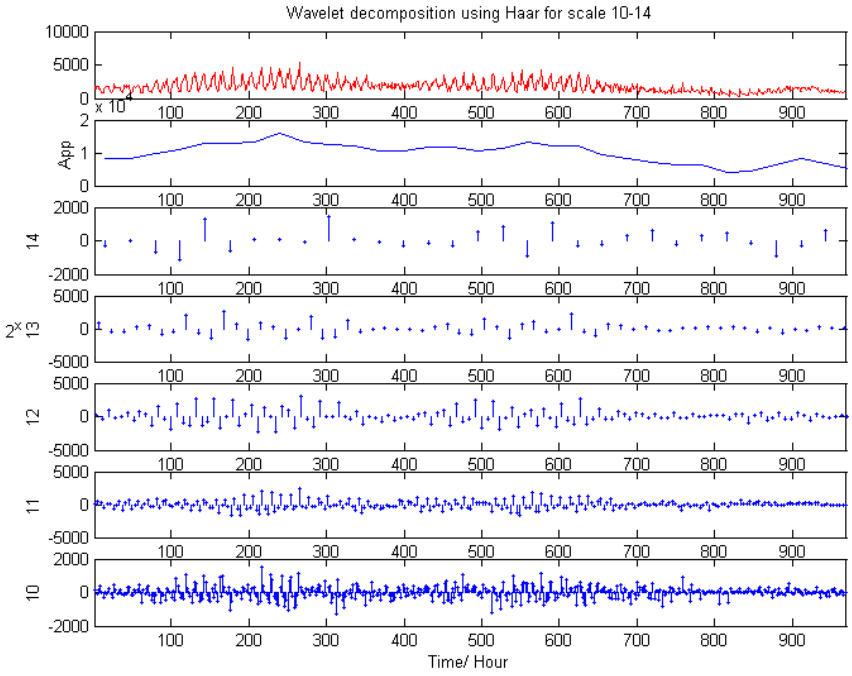



Figure 3.19: Wavelet decomposition octave 10-14.

original time series start to appear. We can see that wavelet coefficients are larger, indicating more changes, at bin 100 to 300 and around bin 500 to 600. Coefficients are smaller, indicating less changes, at bin 400 and 800.

By breaking down the time series into scale we can observe the components in details. For example, octave 1-6 has little resemblance to the original curves. However, they are useful in highlighting the discontinuities, outliers and sudden changes in the environment, while octave 10-14 contains the information of the shape of the curve. Wavelet MRA is a very useful tool for analysing data in a different domain and would help us to determine what is important in the signal, which is used in our compression strategy in the next section.

**Self-Similarity**

There is no obvious periodicity within the sediment data in an hour, however, a slight linear trend may be present. Therefore, we have chosen to use WLSE as the

Hurst estimator for this series.

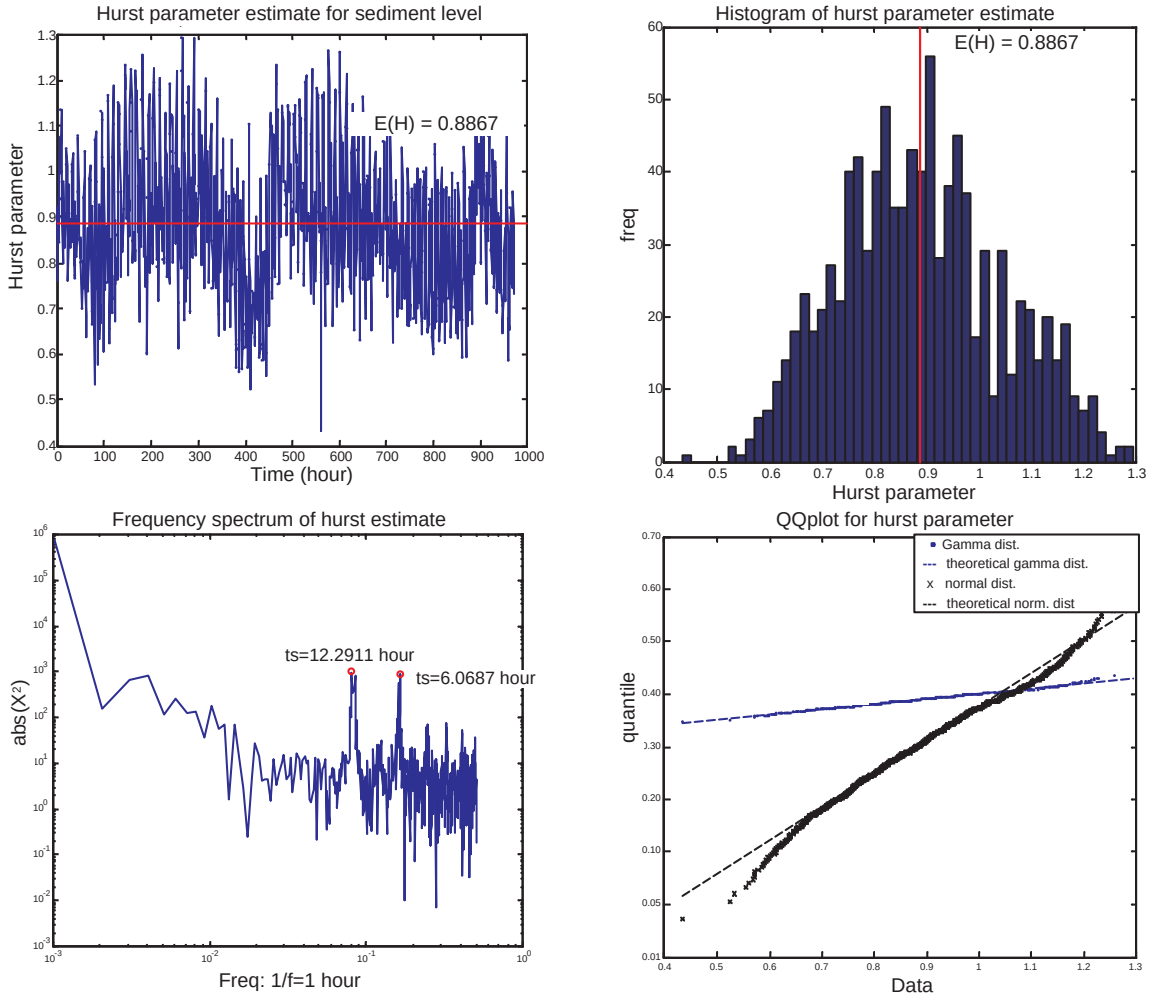



Figure 3.20: Hurst parameter estimation using Wavelet. (a) Hurst estimate for each bin across time using least square estimator. (b) Frequency spectrum for the Hurst estimate in (a). (c) Histogram for Hurst estimate. (d) QQ-plot for Hurst estimate.

Figure 3.20 (a) shows the Hurst parameter estimates for data in each bin/hour. We can observe a clear periodic behaviour similar to the one in the time series. In (b) the frequency spectrum confirms the two frequency peaks at around 12.3 and 6.1 hour, which agree with the result from Figure 3.17 (b). H varies from 0.4 to 1.3 with an average value of 0.8867. This is the evidence of the presence of long-range dependency in the data. The histogram in 3.20(c) shows that the hurst estimate appears to be a random variable. Note that the WLSE Hurst estimate is a random

variable itself which is chi-square distributed [VA99] [PW00]. The result from the QQ-plot in 3.20 (d) shows that the distribution is closer to a Gamma distribution than a normal distribution.

The likelihood of the presence of LRD in the short term sediment data is important in the development of a sampling scheduler in Chapter 4. As our comparison algorithm *eSense* assumes the data to be normally distributed (random walk model). This assumption is hence invalid for the sediment concentration.

### 3.4.4 Summary on Data Analysis

| | **Mean of all data** | **Std. of all data** | **Range of mean of data in a bin** | **Range of std. of data in a bin** |
|---|---|---|---|---|
| **Temp. ($^o$C)** | 11.8975 | 1.3451 | 9.3940-15.1328 | 0.0045-0.2116 |
| **Cond. (mmho/cm)** | 38.7955 | 1.1507 | 36.7237-41.8000 | 0.0064-0.2007 |
| **Pressure (bar)** | 15.9486 | 0.7099 | 14.4538-17.2985 | 0.0147-0.1991 |
| **Turbidity (FTU)** | 55.3299 | 25.8577 | 9.9911-170.8694 | 0.4148-19.4791 |

Table 3.3: Basic statistics of data

In this section, the principal observations and values, which may be useful for making engineering decisions are summarised in Table 3.3 and 3.5. Table 3.4 lists out the important features observed in the time scale of hours, days and months. These features can be used to characterise the data set in that time scale and for the use of the other algorithms.

| | **Hourly feature** | **Daily feature** | **Monthly feature** |
|---|---|---|---|
| **Temperature ($^o$C)** | Mean, standard Deviation and linear trend | Periodicity | Periodicity and Linear trend |
| **Conductivity (mmho/cm)** | Mean, standard deviation and linear trend | Periodicity | Periodicity and linear trend |
| **Pressure (bar)** | Spectrum mean and width, range and standard deviation | Periodicity | Periodicity |
| **Turbidity (FTU)** | Mean, standard deviation and hurst parameter | Periodicity | Periodicity |

Table 3.4: Important features in each time scale

| | | Temperature | Conductivity | Pressure | Turbidity |
|---|---|---|---|---|---|
| Periodicity | Short term | Not observed | Not observed | Wave period 4-12s | Not observed |
| | Medium term | $P \sim 23.7$ and 12.0* hours | $P \sim 23.7$ and 12.3 hours | Mean of bin: $P \sim 24.9$ and 12.3* hours. Std. of bin: $P \sim 6.2$ hours | $P \sim 12.3$ and 6.1 hours |
| Stationarity | Short term | Very slowly varying. Max. rate of change in $1024s = 0.0007^oC$. Can be treated as stationary | Very slowly varying. Max rate of change in $1024s = 0.0006$ $mmho/cm$. Can be treated as stationary | non-stationary | Non-stationary |
| | Medium term | Non-stationary Linear fit= $0.0047^oC/971$ hours, periodic | Non-stationary Linear fit=0.0040 $mmho/cm/971$ hours, periodic | Non-stationary Linear fit=0.0009 $bar/971$ hours, periodic | Non-stationary Linear fit=$-0.0357^oC$ /971 hours, periodic |
| Self-similarity | Short term | Not observed | Not observed | May exist (unknown H) | May exist $H \sim 0.089$ |
| | Medium term | Not observed | Not observed | Mean of bin: not observed. Std. of bin: may exist. $H \sim 1.2$ | Not observed |
| Correlation | | Linear parameter correlation between temp. and cond. $cond. = 0.84 \times temp. + 28.82$ | | | |

**major component*

Table 3.5: Summary of the behaviour of all parameters

## 3.5 Temporal Compression of Sediment Concentration Data

Temporal compression can be used in the SECOAS project to conserve battery power by reducing the number of transmissions required to send all the raw data back. A robust algorithm is required to preserve the important features of the data. The major constraint of temporal compression in SECOAS is the processing capacities of the sensor nodes. The compression technique needs to be simple and capable of deployment in the SECOAS nodes.

As discussed in the previous section, statistic-based wavelet transform allows us to break down the time series into multi-resolution scales. We then perform compression on the transformed signal in each scale. Wavelet compression technique has the advantage that little prior knowledge of the data is required. This is a contrast to model-based algorithms where the compression technique can only be applied to data with the presumed characteristics. Wavelet compression allows us to preserve details in different scales. It is a technique which is also used in the JPEG2000 standards (Section 3.2.2).

This section explores the benefits and feasibility of a lossy data compression strategy and the tradeoffs between data quality and compression ratio. We have investigated the data characteristics of oceanic data and this has prepared us for finding a suitable compression algorithm for the project. This section introduces the basis of lossless and lossy compression theory and terminologies such as entropy. We then explore a wavelet compression technique and investigate the benefits and limitations of it.

### 3.5.1 Review on Data Compression

The key classification for data compression is whether the algorithm produces lossless or lossy results. With lossless compression, a decoder can recover the exact original

input message whilst with lossy compression, only an approximation of the input message is recovered. Text compression is usually required to be lossless whilst image and video can tolerate the use of lossy compression.

An important concept, entropy, is frequently used in coding technologies, especially lossless compression. Claude Shannon has taken the analogy of entropy in thermodynamic to describe how "spread out" the energy of a system is and applied it to information technology. Entropy ($H_E$) of a distribution is defined as:

$$H_E = \sum_i p_i \log \frac{1}{p_i} = -\sum_i p_i \log p_i \tag{3.5.1}$$

$p_i$ describes the possibility of an outcome and log is normally taken to be base 2 as the number of bits required to represent the information. A signal with uniform distribution will have maximum entropy while a system with distribution concentrated at a fix point will have zero entropy. Entropy can be seen as a measure of compressibility in that, with a channel with capacity $C$, it is not possible to transmit an encoded signal at an average symbol rate greater the $C/H_E$. Moreover, with a file containing $n$ samples, we cannot encode the file using fewer bits than $nH$ bits.

Often, applying lossy compression would require some sort of prior knowledge of the signals and user's perception. For example, Pulse Code Modulation (PCM) is used for telephony speech coding. Telephony speech is band-limited to 4kHz and therefore, it is sampled at 8kHz and the signal is quantised using 8-bit to produce 64kbit/s speech coding. A prior knowledge that speech is band-limited is required. In the case of JPEG coding, the image is divided into 8x8 pixel blocks. Discrete Cosine Transform (DCT) is applied to the block and a coefficient matrix is obtained. The matrix is quantized in such a way that high frequency components of the signal are suppressed to zeros according to the compression requirement. In both cases, high frequencies of the signal are less perceptible to human beings and hence, the compression has achieved acceptable or even imperceptible loss in quality and obtained

a high compression ratio.

Popular lossy compression techniques explore the redundancy and relevancy in the data. Techniques such as transform coding and fractal compression look at the correlation of the data to achieve compression. Model-based compression techniques characterise the source data in terms of a strong underlying model. In general, the more prior knowledge we have of the data, the better compression ratio can be achieved. However, it also means the compression technique has less flexibility on the type of applications it can deal with.

### 3.5.2 Temporal Compression

We explore the use of wavelet threshold compression in this section and use Discrete Wavelet Transform (DWT) to convert the original series to wavelet coefficients, which have the same length as the original time series. Coefficients that are smaller than a threshold would be suppressed to zeros. We then combine lossless encoding techniques such as run-length zeros (RLZ) or Huffman coding to shorten the series and achieve compression. RLZ is chosen because of its simplicity to implement. A series that is comprised of many sequential zeros would benefit most from RLZ coding.

In this section, we use two methods to determine the threshold of compression.

- Fix ratio of the largest magnitude coefficient (FRC) Taking a fix fraction of the largest coefficient in a particular wavelet scale, the magnitude of the threshold is determined by the equation,

$$threshold = max(abs(C_j)) \times threshold\ percentage, \tag{3.5.2}$$

  and $C_j = 0$ if $|C_j| < threshold$, where $C_j$ represents wavelet coefficients at scale $j$.

- Deviation from normal distribution (DND)

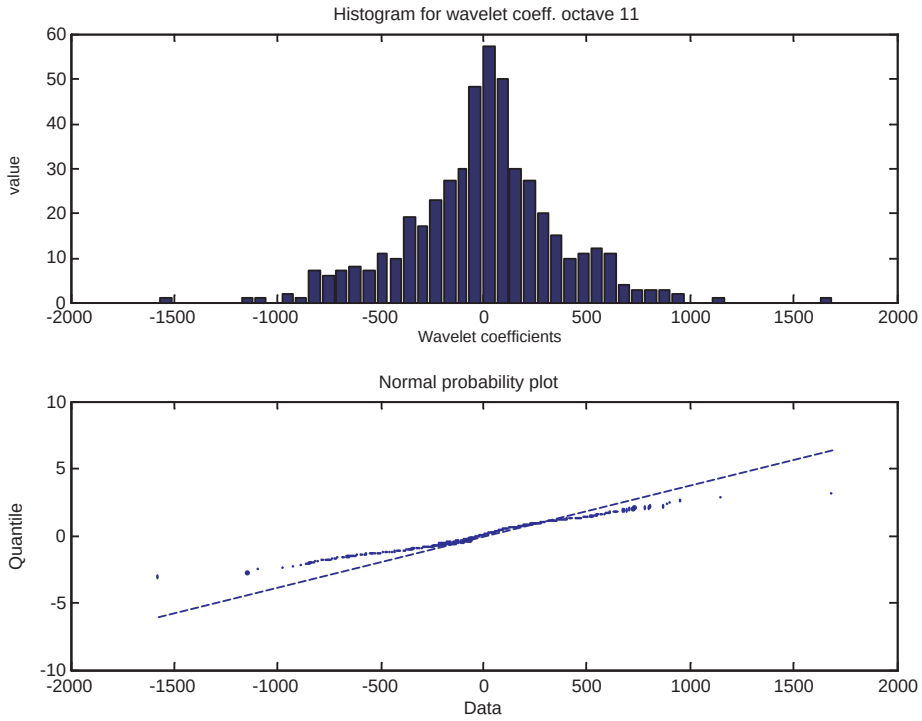



Figure 3.21: (a) Histogram of wavelet coefficient at octave 11 (b) QQ-plot of coefficient against normal distribution. The solid line is the theoretical result if the data is normally distributed.

The second method follow the analysis result in section 3.4.3 to choose the thresholds based on the deviation from a normal distribution using QQ-plot. From Figure 3.21 we can see that if we plot the wavelet coefficients against a normal distribution, the coefficients with small amplitude follow the distribution in a straight line, while the larger coefficients deviate from the line. Hence, we set the boundaries of the data to be where the data deviate from the normal distribution by a certain deviation ratio of the quantile, as shown in Figure 3.22, which is determined by the equation:

$$boundary = range\ of\ quantile \times deviation\ ratio \tag{3.5.3}$$

The lower threshold is where data crosses the boundary on the left hand side to zero. The upper threshold is where data crosses the boundary on the right hand

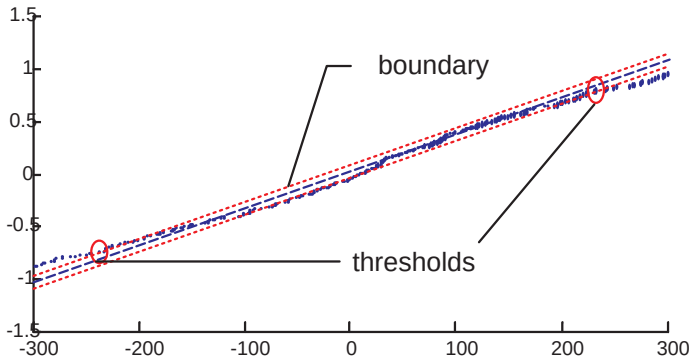



Figure 3.22: Boundary and threshold

side to zero, and $C_j = 0$ if $C_j \leq$ *lower threshold* or $C_j \geq$ *upper threshold*.

Both methods distinguish the important wavelet components (signal) from the irrelevant coefficients (noise).

*Haar wavelet* is chosen because of its simplicity. The wavelet and scaling functions in each scale are pre-calculated and put in a maths table in the MCU. The remaining operations at the time are multiplying the wavelet and scaling functions with the data and summation. A 6-level wavelet transform referring to $\tau_j = 1$ to 32 seconds (Octave 1-6) is used. The scaling coefficients at octave 6 approximates the trend of the data with $1024/2^6 = 16$ coefficients.

**Proposed encoding procedure**

1. 6-level Discrete Wavelet Transform (DWT) is performed using Haar wavelet.
2. For each wavelet scale, choose the threshold based on.
   - Method 1: fix ratio of the largest coefficient.
   - Method 2: deviation of the coefficients from the normal distribution.
3. If a coefficient is within the threshold, it is replaced by zero; if it is beyond the threshold, it will be kept.
4. Calculate the mean of the wavelet coefficients that are within the range of the thresholds and have been suppressed to zero.
5. Encode the wavelet coefficients using run-length zeros algorithm.
6. Suspend the mean at the end of the encoded coefficients.

**Proposed decoding procedure**

1. Decode the coefficients using run-length zeros.
2. Substitute zeros in the coefficients with the mean of the suppressed coefficients.
3. Reconstruct using Inverse Discrete Wavelet Transform (IDWT).

### 3.5.3 Results

Two quantities are computed in the compression, the compression ratio and the error incurred by the compression. They are defined as:

$$Compress\ ratio = \frac{(N - N_{comp})}{N} \cdot 100\% \tag{3.5.4}$$

$$Error = \frac{\sqrt{sum(X - X_{comp})^2}}{N}, \tag{3.5.5}$$

where $X$ is the original series and $N$ is the length of data, $X_{comp}$ is the reconstructed series using IDWT and $N_{comp}$ is the length of the compressed wavelet coefficient series.

**Method 1: Fix Ratio of the Largest Coefficient** Figure 3.23 shows the result using method 1 and a threshold ratio of 0.01. Compression is performed to each bin and observed over time. The magnitude of the error incurred by compression is in the order of $10^{-2}$ FTUs (same unit as the turbidity), which is small compare to the average values of turbidity of 55 FTUs. This is because the compression is only done on the wavelet coefficients but not the scaling coefficients. The majority of the energy in the data is in the scaling coefficients. The compression ratio over time is around 46%.

Figure 3.24 shows the effects on the result by varying the threshold percentage by compression bins 1 to 200 and the average of the compression ratios and errors are obtained. It is observed that the compression ratio remains close to 50% until

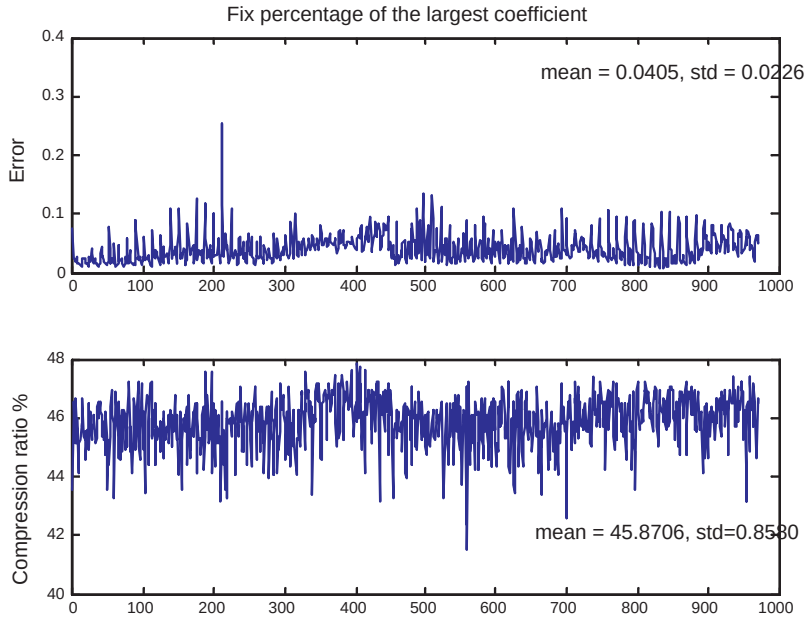


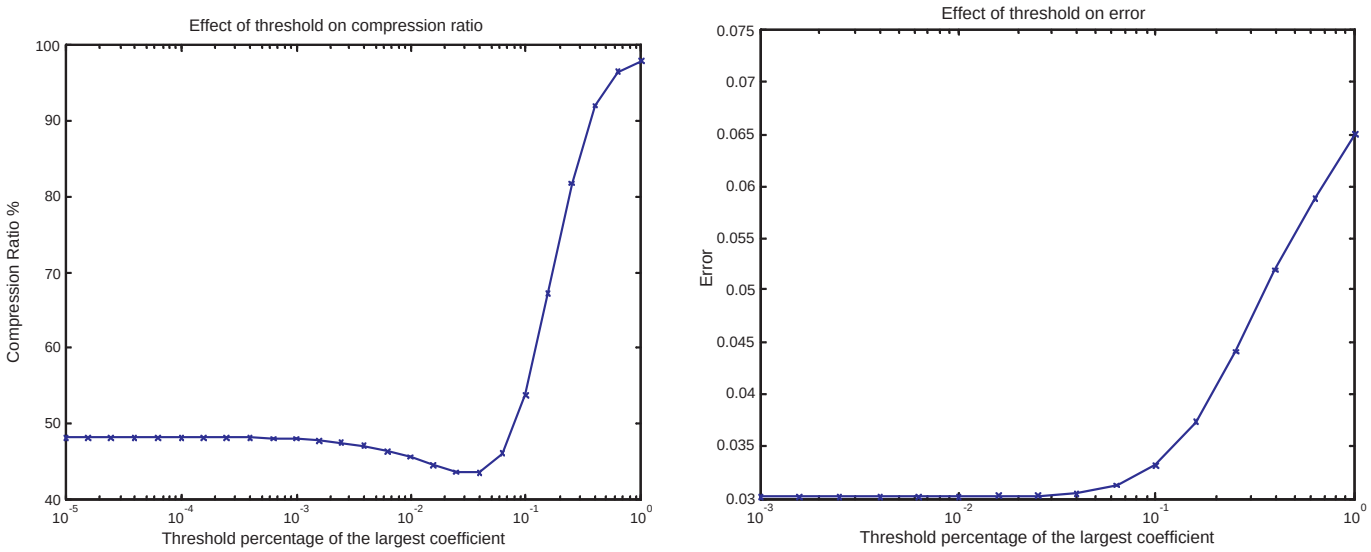

Figure 3.23: Result of compression using fix percentage over time

the threshold percentage is over 0.1 of the largest coefficient in that scale. At a ratio of 1 the threshold percentage equals the largest wavelet coefficient in each scale and only the scaling coefficients are preserves. The compression ratio achieved is 98% and the error is 0.0065, which is still very small.

**Method 2: Deviation of the Coefficients From Normal Distribution**

Figure 3.25 shows the result of the compression ratios and errors over time, using a deviation ratio of 0.1. The compression error is around 0.45 and the overall compression ratio of this setting is around 50%. One major difference of method 2 from method 1 is that the compression ratio varies between a few percentage to 80% in contrast to method one, which has the range of only a few percentages for different bins of data. This is reflected in the errors shown in figure 3.25. This may be due to the deviation ratio chosen falling in the linear region of the curve in Figure 3.26 (a).

Figure 3.26 shows the effect on the results by varying the deviation ratio and testing on bins 1 to 200. It is observed that both compression ratios and errors increase with deviation ratio. The negative compression ratio means the length of




(a) (b)

Figure 3.24: Effect of thresholds (a) on compression ratio, (b) on compression error

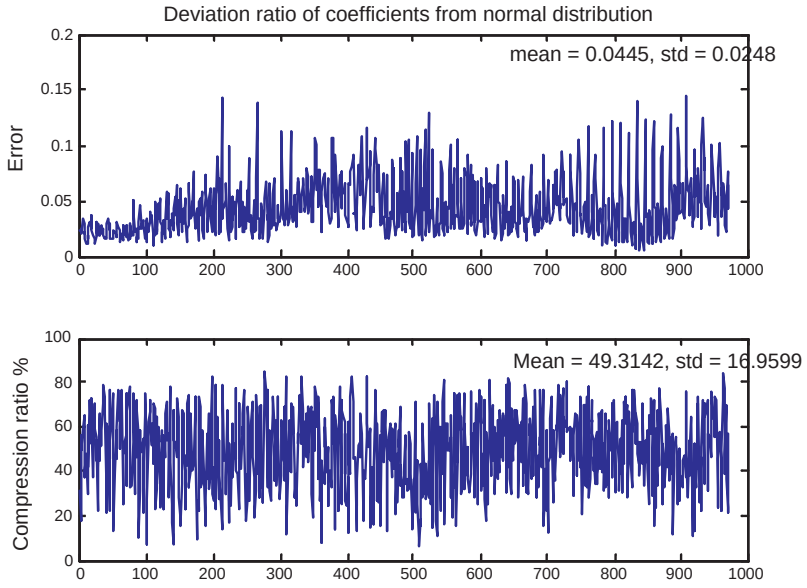
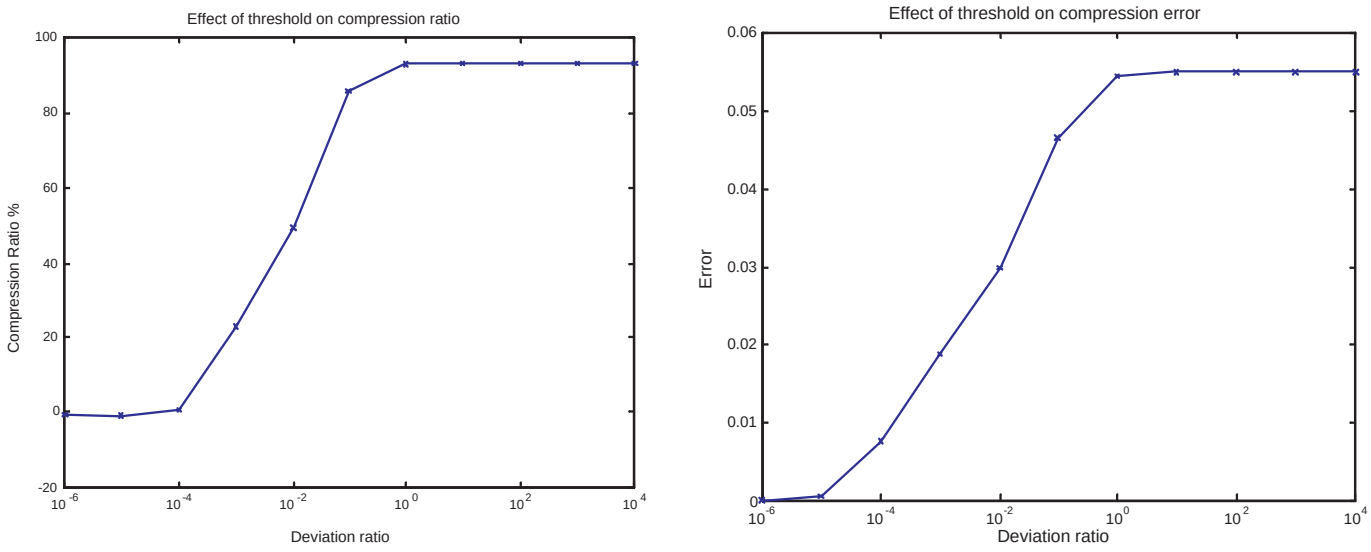



Figure 3.25: Result of compression using deviation threshold over time

the compressed vector is longer than the original series. In (b) the linear region of compression error is between 10-5 and 100. After that the curve flattens and reaches a maximum error of 0.05. It is also observed that the linear region of compression ratio in (a) has a larger range running from 1 to 96 than method 1, which runs from




(a) (b)

Figure 3.26: Effect of deviation ratio (a) on compression ratio, (b) on compression error

50 to 98.

### 3.5.4 Comparison and Conclusion

The difference in the reconstructed series resulting from method 1 and 2 are studied here. Bin 1 of sediment data is chosen for the study because of the presence of two anomalies of large amplitude.

Figure 3.27 is a comparison of the reconstructed series with different threshold percentages using method 1. Graph (a) is the original series, where two spikes are observed. The last series (d) is the result for a threshold percentage of 1, which means all the wavelet coefficients are suppressed. The reconstructed series shows the trend of the series only with all the high frequency changes missing. In (b) and (c), the two spikes are observed with smaller amplitudes than the original.

Figure 3.28 shows the results of different deviation ratio using method 2. Similar effects are observed where larger deviation ratios lead to more coefficients suppressed

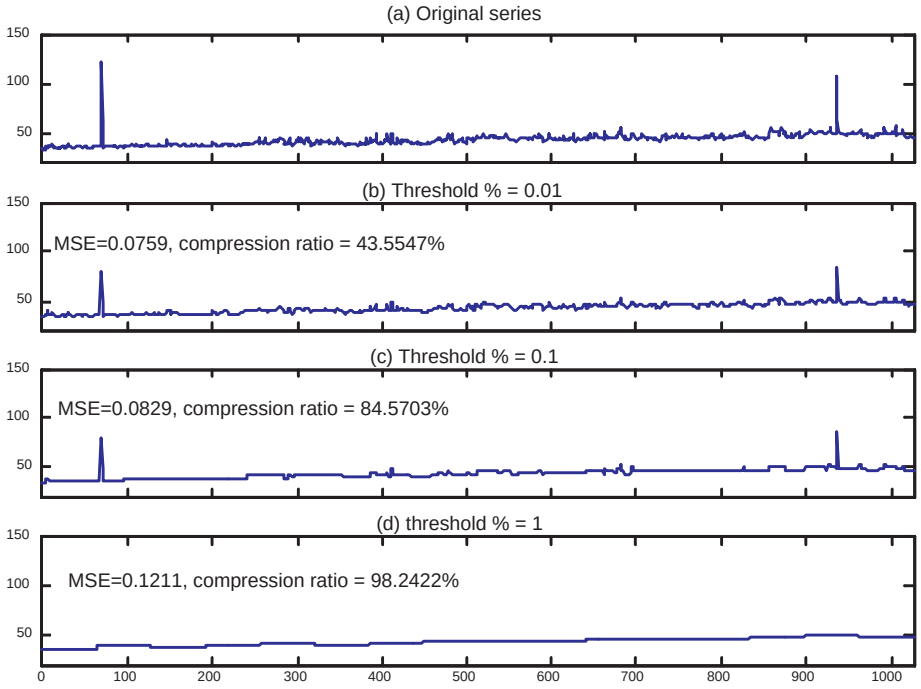



Figure 3.27: Compressed and reconstructed time series of sediment concentration using method 1: threshold percentage of the largest coefficients.

to zeros and hence, the high frequency components are not preserved. However, as the threshold is a ratio of the deviation from the theoretical values of normal distribution, coefficients with large deviation will always be maintained and the anomalies are kept. The two spikes are still observed with the ratio set to 5 and the compression ratio achieved is 96%, 2% less than if only scaling coefficients are transmitted. The following observations are obtained:

- From observing the time series in Figure 3.23 and 3.25, method 1 gives better control of the compression ratios over time than method 2. This is because the location of the linear region of method 2 (figure 3.26 makes the compression sensitive to the data values.

- Observing the results from Figure 3.27 and 3.28, method 2 is more useful for preserving anomalies.

- In general, method 2 is more complicated to compute and implement on an

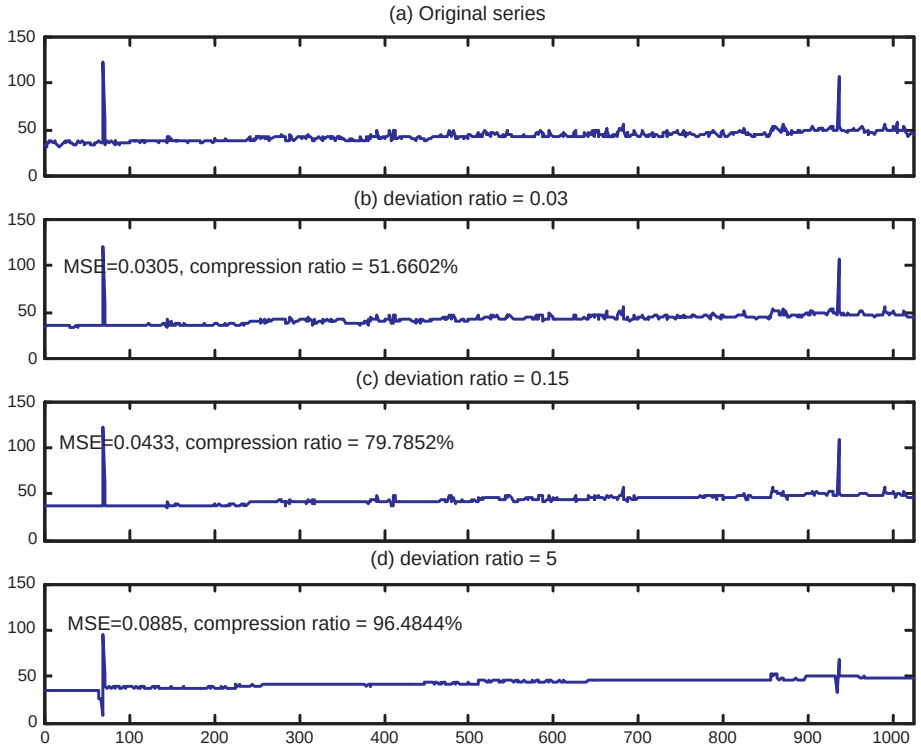



Figure 3.28: Compressed and reconstructed time series of sediment concentration using method 2: deviation ratio.

MCU board. Performing a QQ-plot involves:

1. Obtaining an order statistics of the data
2. Getting a theoretical series from the tested distribution. This series can be computed prior to implementation
3. Calculating the quantile/ percentile based on the data

Method 1 only involves a simple multiplication of the largest coefficients, hence, it is a lot easier to implement. Hence, wavelet compression using method 1 is more desirable in the SECOAS project. Moreover, it also gives a more consistent compression ratio over time.

## 3.6 Chapter Summary

In this Chapter, we reviewed different WSN data handling projects, data analysis tools, self-similarity and hurst parameters and wavelet transform and compres-

sion techniques. We then included a throughout data analysis on the oceanography data for our projects. All the parameters, temperature, conductivity, pressure and sediment concentration exhibited periodic features due to tides. Temperature and conductivity data showed strong linear relationship, which might be used for data compression purpose. We found that Sediment concentration showed long-range dependency characteristics with the Hurst estimate of 0.89.

Wavelet data compression was explored for the use on sediment concentration data using thresholding on wavelet coefficients followed by run-length zeros coding technique. Two methods were explored to suppress wavelet coefficients: deviation from normal distribution and fix percentage of the largest coefficient. It was found that both methods achieved good compression ratio 50-95%, while using deviation from normal distribution preserved the data abnormally better. However, the algorithm was more complicated and hence, was less desirable for the SECOAS types of environmental monitoring projects.

# Chapter 4

# Development of an Adaptive Sampling Scheduler

| symbol | description | equation | page no. |
|---|---|---|---|
| $\alpha$ | gain parameter for EWMA | 4.3.1 | 108 |
| $\hat{alpha}$ | normalised $\alpha_{long}$ and $\alpha_{short}$ for analysis | 4.4.1 | 113 |
| $\bar{d}_i$ | mean difference of reading between $i$ and neighbours | 4.3.4 | 110 |
| $E$ | averages for EWMA | 4.3.1 | 108 |
| $E_{long}$ | long term EWMA | 4.3.1 | 108 |
| $E_{short}$ | short term EWMA | 4.3.1 | 108 |
| $\gamma$ | miss ratio | 4.2.2 | 106 |
| $LB$ | lower bound regular sampling rate | – | 118 |
| $M$ | no. of time before re-broadcasting messages | – | 110 |
| $n$ | total number of sampling point | 4.2.2 | 106 |
| $n_f$ | number of missed hits | 4.2.2 | 106 |
| $n_p$ | number of false hits | 4.2.3 | 107 |
| $\hat{p_{idle}}$ | temporal probability of idling | 4.3.3 | 108 |
| $\hat{p_{sense}}$ | temporal probability of sensing | 4.3.3 | 108 |
| $p_t$ | normalised probability | 4.3.2 | 108 |
| $\rho$ | false hit ratio | 4.2.3 | 107 |
| $s$ | shift parameter | 4.4.2 | 115 |

| symbol | description | equation | page no. |
|---|---|---|---|
| $T$ | threshold for neighbour reporting | – | 110 |
| $TH$ | relative threshold | – | 117 |
| $TH_{abs}$ | absolute threshold | 4.4.3 | 118 |
| $UB$ | upper bound regular sampling rate | – | 118 |
| $x(\cdot)$ | sample at time | 4.3.1 | 108 |

A sampling scheme that can alter the sample rates according to the actual environmental changes can be very attractive in WSN as it reduces the amount of energy needed for taking samples and relaying the data back to the base station. This is particularly useful for applications where data changes at different rates over time and for rare-event detection.

In-network processing techniques have been developed to reduce communication overhead at the expense of more processing. Energy is conserved because as many as 1000 operations are required to consume the same power required to send one bit of data. Thus, wherever possible, data are processed within the sensor network.

Traditionally, temporal samples are taken at regular intervals [ADJ+02] [BBB04], governed by the Nyquist frequency if the data is periodic, to avoid aliasing. In the cases where more than one frequencies are of interest and are wide apart in temporal scale, such as wave and tidal periods in Oceanography analysis [RLW+04], burst sampling technique may be adopted as a mean of conserving power. However, regular or burst sampling techniques are not power efficient for rare-event detection. If the sampling frequency is set to be very low there is a possibility that important events may be missed. However, if the sampling frequency is set to a high value most sampling is redundant as it takes place during normal (no event) circumstances.

We propose a stochastic sensing schedule that adapts the sampling frequency to the chosen statistics collected from the environment. Compared to a regular sampling strategy, the frequency of the required sampling rate decreases and yet the scheduler

preserves and detects the phenomenon when it happens. The amount of information that needs to be sent back to the user decreases, and therefore, greatly reduces the power consumption in a node. Moreover, the sensor network will be able to predict and detect when a rare-event happens and trigger further actions such as alarms and alerts. Neighbour coordination can also be introduced to reduce event-detection delay due to sparse random sampling when there is no event.

Although adaptive sampling can be beneficial in the respect of energy saving and detection delay, temporal irregularity in data, which is a by-product of this type of sampling techniques, means that we cannot use most of the commonly available tool including Fourier and wavelet transform to analyse the collected data. Techniques such as interpolation and extrapolation may be used to construct a regular data set but it introduces uncertainty in the data series. Therefore, adaptive sampling is only used on WSNs that have clear objectives such as the detection of an event, that only the recent values are required by the user and data analysis is not required.

Previous works that have been done on adaptive sampling schedulers rely on a prediction model. This has limited the use of algorithm to the particular data type defined for the application. Moreover, the models require prior training and re-training during the deployment in order to keep the accuracy of the predictions, which impose additional processing and memory requirements in a sensor node. We propose a model that use a quantity called Exponentially Weighted Moving Average (EWMA) to calculate the probability of sampling, which adjusts to the rate of change of the latest samples. The method requires very little prior knowledge on the data and has low memory requirement. It is also applicable to most data types for event detection.

We start the discussion with a literature review on data sampling techniques in WSN. We then introduce the performance metrics that will be used in the evaluation of the algorithm developed. Our work on the design of the Basic Adaptive

Sampling Scheduler (BASS) is reported in the next session. After that, we evaluate the parameters in the algorithms and elaborate the adaptive component in BASS that has feedback from the measured data into the algorithm. In the final section, BASS is compared to eSense for the resulting sampling ratios and their ability to capture events.

## 4.1 Related Works

Research in sensing schedulers mostly focuses on exploiting redundancy in the network to minimise battery usage. Some consider radio and sensing coverage redundancy and optimise network life by putting the covered nodes to sleep [CAHS05] [HLTC06] [RLW+04]. With this approach only the spatial but not the temporal statistical redundancy is considered. However, the concept of using spatial redundancy to reduce sampling frequency is similar to the spatial neighbour coordination of BASS, the only difference is that the coordination in BASS is not organised and scheduled, but probabilistic.

The work of Tulone et. al. attempts to reduce the communication budget by using forecasting models. They develop a probability query adaptable system [TM06] which used a combination of autoregressive (AR) models located on each sensor to predict local readings. Model updates are only sent to a sink whenever measured readings are outside specified bounds. Jinboa et al proposed a similar scheme which dynamically adjusted the sampling frequency using a linear regression model [LL07]. These works have highlighted the importance of reducing the amount of communication in WSN by predicting whether the next reading is important to the user. However, both methods are limited by their model assumptions. Hence, the algorithms only apply to applications with defined data characteristics.

In [WAJR+05a], Werner et al developed an exponentially weighted moving average (EWMA) (see Section 4.3) detector to counteract sensitivity issues [WAJR+05a]

in the project. Detection is triggered when the ratio between a long and short term average is above some pre-specified threshold thus causing nodes to sample data continuously. The work has inspired us to develop our adaptive scheduler based on the simple and robust statistics of EWMA, which requires very little physical memory to implement on the sensor node. BASS uses similar concept of long term and short term averages calculated by EWMA, however, in our proposed algorithm these averages are used to drive a probabilistic state machine to whether a node samples or idles, while in [WAJR+05a], the difference of the EWMAs are used to turn on the detector.

In [LCS06], Liu et al proposed a stochastic sampling scheduler *eSense* achieving similar objective to our requirements. The scheduler uses a biased random walk method to predict the likelihood of an event happening in the $k$ steps time and determines whether a node should sample at that instance. The system also uses false hit ratio (the recent number of sampling instances which do not detect an event) to fine-tune the sampling probability. eSense has inspired our work in the adaptive feedback in BASS. However, eSense is largely based on the assumption that the environmental data follow a random walk model, while BASS based the sampling probability on EWMA statistic. This is the major difference between the two algorithms.

## 4.2 Performance Metrics

An adaptive sensing schedule is useful for two types of applications. The first application is simply relaying data to the user. The scheduler works such that the amount of data collected is proportional to the rate of change of the environment. The second application is rare-event detection. When a rare event occurs, the sensor network is required to respond to the event promptly and reliably and notify the users. Examples of such applications are bridge collision monitoring, flooding

detection, landslide warning systems and forest fire detection [AWSC02].

We define the following performance metrics to help us evaluate the sampling schedulers for both type of applications.

### 4.2.1 Root Mean Square

Environmental *Root mean squre* (RMS) intends to show the difference in environmental value and the last reading in a sensor node. It is defined as,

$$RMS = 1/N * \sqrt{\left( \sum_{all\ sensors} (environmental\ value - value\ last\ measured)^2 \right)}, \tag{4.2.1}$$

where $N$ denotes the number of sensors. $RMS$ is particularly useful in analysing the performance of an algorithm in time. It shows the delay of a state change detection and the effect of coordination of the nodes in the network.

### 4.2.2 Relative and Absolute threshold

*Relative threshold* is applicable to the first type of data relaying application. Relative threshold is defined as the difference between the last and the current measurements to trigger an event. When the rate of change of the environment is fast, the scheduler should sample more often to track measurement that have crossed the relative thresholds.

For event detection, an *absolute threshold* is more applicable. An absolute threshold defines the absolute reading that the measurement needs to cross for an event to happen. Detection of landslide for embankment monitoring is such an example. The tilt must be read above a certain threshold to indicate the event of a landslide.

In both cases, a *state change* indicates when a reading crosses the threshold (both relative and absolute), which triggers the reporting to the application.

### 4.2.3 Baseline sampling frequency and periodic sampling

The sensor nodes has a maximum baseline sampling frequency that is bounded by the sensor type and the ADC used. The baseline sampling rate determines the maximum resolution of the system. The adaptive sampling schedule can only give resolution equal to or lower than the baseline sampling frequency. In our simulated environment, the baseline frequency is the resolution of the data set collected, or the data set created for the simulation.

We used the oceanographic data sets for comparison and validation of our algorithm. However, we could not use a discontinued data set in our simulation as in reality the data does not jump from one state to another without intermediate steps and these intermediate steps are very important to us as it determines the rate of change of the environment. Hence, we could only use the coarse sets of the data and the baseline frequency is 1 per hour.

In our experiment we also compared the performance of the scheduling algorithms to periodic sampling. In theory, adaptive sampling algorithms adjust the sampling rate based on the rate of change of the data, hence, we expected to see lower missed ratio than using periodic sampling provided that the average sampling rate is the same.

### 4.2.4 Missed Ratio and False Hit Ratio

Unnecessary sampling should be avoided using an adaptive sampling schedule, therefore, a sensor node only samples when a state change is expected. With this approach, it is possible to miss certain state changes if data was not sampled at those particular time instants. This is referred to as the *miss ratio*, which is defined as [LCS06],

$$\gamma = \frac{n_f}{n}, \tag{4.2.2}$$

where $n_f$ denotes the number of missed hits and $n$ denotes the total number of

sampling points corresponding to the base sampling frequency.

Conversely, when a node takes a sample when no state change is happening, it is refer to as the *false hit ratio*, which is defined as,

$$\rho = \frac{n_p}{n}, \tag{4.2.3}$$

where $n_p$ denotes the number of false hits and $n$ denotes the total number of sampling points corresponding to the base sampling frequency.

Miss rate is also related to the delay of detecting a state change. For example, if the state change happens one second after the last sample, but the algorithm only schedules the next sampling event to happen after $T$ seconds, the delay of detection is $(T-1)s$ and the scheduler would have missed $(T-1)$ events compared to the baseline sampling rate $(1s)$, provided that the data remains at the same state after the state change.

## 4.3 Design of Basic Adaptive Scheduling Algorithm

### 4.3.1 Temporal Design

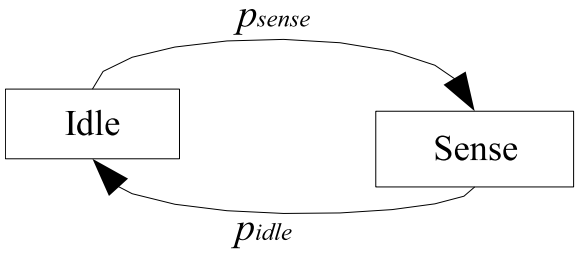



Figure 4.1: State diagram for adaptive sampling

Inspired by the research in [WAJR+05b], we have designed a Basic Adaptive Sampling Scheduler (BASS) to achieve our goal to track the environmental rate of change. The scheduler is based on a simple 2-states model as shown in Figure 4.1. $p_{sense}$ and $p_{idle}$ are the probability of the scheduler changing from one state to

the other. At sense-state, the scheduler takes a sample from the environment and updates $p_{sense}$ and $p_{idle}$ with the current sample. The state-transitions are random probabilities, which are based-on a statistical property called Exponential Weighted Moving Average (EWMA) denoted in 4.3.1. Probabilistic approach de-synchronises the regularity of sampling both temporally and spatially, hence, a node can detect a regular event in due time and node coordination can be applied to improve the system performance.

$$E(t) = \alpha \cdot x(t) + (1 - \alpha) \cdot E(t - 1), \tag{4.3.1}$$

where $E$ is short term ($E_{short}$) or long term ($E_{long}$) EWMA, $x(t)$ denotes the current sample, $\alpha$ is a gain parameter between 0 and 1.

EWMA is a way of preserving memory when calculating the expected values. $\alpha$ determines the weights of the current sample and the last average when calculating the new average. Current values are weighted more than historical values if $\alpha > 0.5$.

In BASS, we define a long-term average $E_{long}$ and a short-term average $E_{short}$ which uses the same EWMA equation 4.3.1 with $0 \leq \alpha_{long} < \alpha_{short} \leq 1$ and we defined normalised probability $p_t$ as a normalised ratio as below:

$$\begin{cases} p_t = \left| \left( \frac{E_{long} - E_{short}}{E_{long} + E_{short}} \right) \right|, & E_{long} + E_{short} \neq 0 \\ p_t = 0, \quad , & E_{long} + E_{short} = 0 \end{cases} \tag{4.3.2}$$

The temporal components of $\hat{p}_{sense}$ and $\hat{p}_{idle}$ are:

$$\begin{aligned} \hat{p}_{sense} &= k_1 + (1 - k_1) * p_t \\ \hat{p}_{idle} &= 1 - \hat{p_{sense}}, \end{aligned} \tag{4.3.3}$$

where $0 \leq k_1 \leq 1$ is the minimum sampling probability, $p_t$ is the normalised probability and $\hat{p_{sense}}$ and $\hat{p_{idle}}$ are the temporal element of probability of sensing and idling. $0 \leq p_t \leq 1$ and $p_t$ increases with the difference between $E_{long}$ and $E_{short}$.

Without $k_1$, $\hat{p}_{sense}$ is simply $p_t$ and $\hat{p}_{idle}$ is equal to $1 - p_t$, which means that

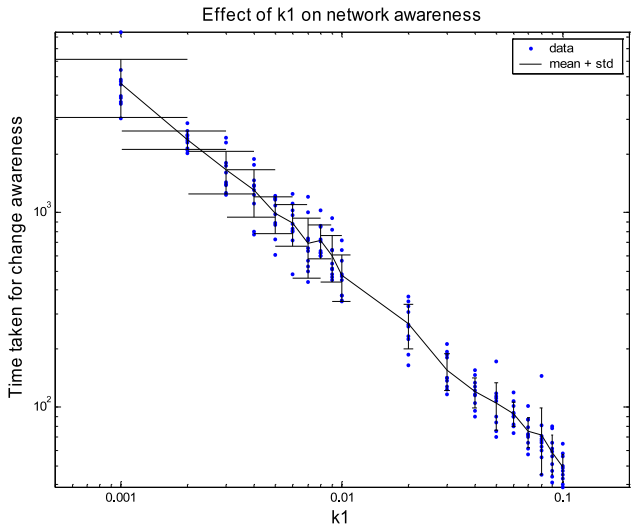



Figure 4.2: Relation of $k_1$ and network awareness in loglog scale. $\alpha_{long} = 0.01$, $\alpha_{short} = 0.05$. Environment changes from 0 to 50 at iteration 200.

sampling probability increases with the difference of long-term average and short-term average. $k_1$ is an integrity threshold that makes sure sampling probability does not fall to zero when the difference between $E_{long}$ and $E_{short}$ is very small. When the environment is stable, the sensor nodes are sampled asynchronously at a random rate close to $k_1$. One can visualise a problem when $k_1$ is very small, i.e., there would be a long delay for the detection of an event, or it is simply missed. The experiment set in Figure 4.2 demonstrates how $k_1$ affects the time for network to be aware of a change occurring to the whole network. The measured environment value changes from 0 to 50 (arbitrary units) at time instance 200. The time for awareness of the change in the network has an exponential relationship to $k_1$ as shown in Figure 4.2. A small $k_1$ sets a small sampling rate when no event happens in the environment, meaning a higher energy efficiency in long term.

### 4.3.2 Spatial Coordination

Setting $k_1$ to a small value can enhance energy saving in the network, however, the sensing rate is so slow after the settling period that it would take a long time for

the sensor nodes to pick up on any event happening in the environment. Therefore, neighbour coordination is introduced to improve detection delay, given that the neighbouring nodes are providing spatial sensing coverage for each other. When a single node detects some change in the network, it alerts the neighbours with its discovery. We introduce a spatial component in relation to the *reading difference* between the two nodes $\bar{d_u}$ to the equation in determining $p_{sense}$ and $p_{idle}$. The distance of node $u$ from its neighbours $v$ (if information is received from $v$) is defined in Eq. 4.3.4 and is the summation of the absolute differences of $p_t$ of all neighbours.

$$\bar{d_u} = \frac{1}{n} \sum_j |p_{t_u} - p_{t_v}| \tag{4.3.4}$$

The final equations (Eq. 4.3.5) for $p_{sense}$ and $p_{idle}$ are such that the probability of sampling is high when either $\bar{d_u}$ or $p_{t_u}$ is high; and the probability of sampling is low when both $\bar{d_u}$ and $p_{t_u}$ are low.

$$\begin{aligned} p_{sense} &= k_1 + (1 - k_1)(1 - (1 - \bar{d_u})(1 - p_t)) \\ p_{idle} &= (1 - p_t)(1 - \bar{d_u}) \end{aligned} \tag{4.3.5}$$

In the model developed, we use a push-approach to the information dissemination problem among neighbours. A sensor node sends a report to its neighbours when its $p_t$ is above a defined threshold ($p_t > T$), which indicates a significant change in the environment measured. Currently, $p_{sense}$ and $p_{idle}$ are updated at every time step, even when a node is at idle state, because the messages received from a node's neighbour may change the sensing schedule of a node. To minimise the communication expenses, a node which has broadcast its $p_t$ to its neighbour will wait for $M$ time steps before considering re-broadcasting its $p_t$ information. We set up an experiment to observe the effect of neighbour communication in the algorithm. $T$ is set to 0.6 and $M$ is set to 10 in the experiment. The network consists of 100 nodes in a $[100 \times 100]$ space. The environment is a step function that changes from 0 to 10 at

time instance 0. The result is presented in figure 4.3 and 4.4. We can see from figure 4.3 that where the number of sampling events is measured, when no communication exists between the neighbour, individual nodes rely on their probability of sampling to detect the change in environment. The sensor nodes in the network gradually realise there is a change to the environment and increase their sampling probability accordingly until their long-term average has adjusted to the new value. Hence, the detection peak is flattest among the comparison. With neighbour communication, the rate of detection for all nodes is much earlier and most nodes are aware of the changes at a similar time, hence, the peak of detect is much sharper. The delay is less for communication range = 20 (average degree = 10.18) than range = 10 (average degree = 3.22) and range = 5 (average degree = 0.68) since the nodes have more neighbours that are within the communication range.

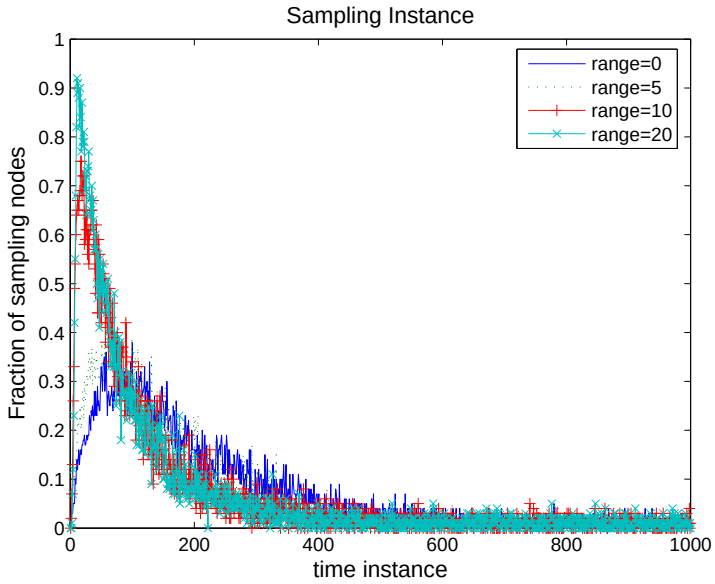



Figure 4.3: Number of sampling instances vs time. $n = 100$, $l^2 = [100, 100]$, $\alpha_{long} = 0.05$, $\alpha_{short} = 0.01$, $T = 0.6$, $M = 10$. The environment is a step function from abitrary values 0 to 10 at time 0. Number of iterations is 1000.

Figure 4.4 shows the results of the RMS over the network with time. At time instance 0 the environment changes from 0 to 10. From time 0 onwards, the sensor

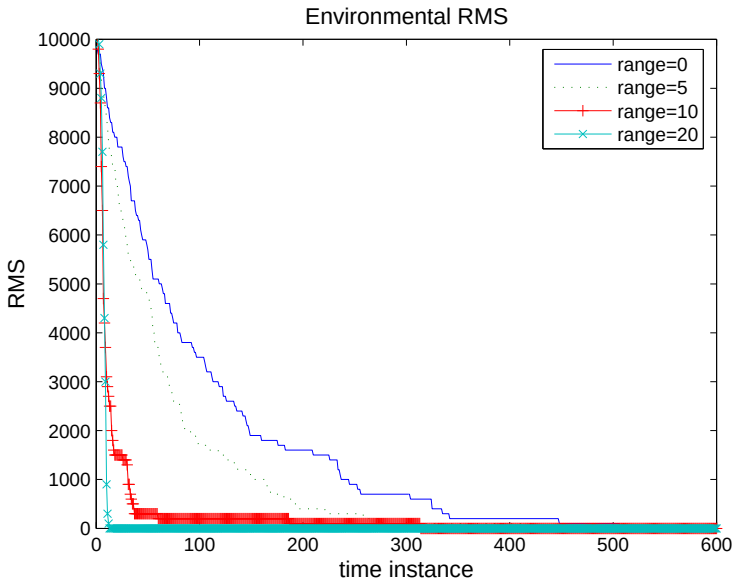



Figure 4.4: Environmental RMS over time

nodes in the network gradually realise there is a change to the environment and increase their sampling probability accordingly until their long-term average has adjusted to the new value. The percentage of node sampling in the network settles to a value close to $k_1$, which is set to 0.01 in this experiment, at time 500. Figure 4.4 shows the environmental RMS defined in section 4.2.

We can see from both figures that a larger communication range means a node can exchange their $p_t$ with more neighbours. The environmental RMS converges much quicker to zero when communication range is set to 20, meaning a much faster response of the network to the environmental changes and information disseminates more efficiently across the network. As a result, nodes which are monitoring the same area can share their responsibility of sensing and hence, $k_1$ can be set to a lower value in the group of nodes.

## 4.4 Setting the Parameters

### 4.4.1 Relation of $\alpha$ and $p_t$

We can see from equation 4.3.1, 4.3.2 and 4.3.5 that $p_t$, i.e. $\alpha_{long}$ and $\alpha_{short}$, defines the probability of sampling in relation to environmental changes. The value of $\alpha$ is between 0 and 1, the smaller the value of $\alpha$ is, the quicker the scheduler can adapt to the changes. However, the scheduler is also more exposed to noise and short term variations that we may not want to capture. We can exclude the short term variations and noise by choosing a larger $\alpha$ values, however, this may also mean our scheduler reacts more slowly to changes.

In order to investigate the combined and distinctive effect of the long and short term alphas and the averages, we have created a normalised parameter $\hat{\alpha}$ to relate the $\alpha_{long}$ and $\alpha_{short}$ in our analysis as in the following equation:

$$\hat{\alpha} = (\frac{1}{\alpha_{long}}) - (\frac{1}{\alpha_{short}}) \tag{4.4.1}$$

$\hat{\alpha}$ can be seen as the difference in the sampling instances $(1/\alpha)$ to create the long term and short term EWMA averages, and the convergence time of $E$ is approximately $5 * 1/\alpha$. $\hat{\alpha}$ has a direct relation in the calculation of $p_t$ as shown in figure 4.5 and figure 4.6. In this experiment, the experimental time series is a step function from 0 to 10 at time instance 0.

As an approximation, $\alpha_{short} \sim \hat{\alpha} \cdot \alpha_{long}$, as $\alpha_{short} \ll \hat{\alpha}$.

Figure 4.5 shows the maximum $p_t$ obtained in the experiment plotted against $\hat{\alpha}$. Even at a high value of $\hat{\alpha}$ where the difference of $E_{long}$ and $E_{short}$ is large, to get a good ratio between the two averages, the value of $\alpha_{short}$ is required to be greater than 0.05 to get a full range of $p_t$ given a reasonable value of $\alpha_{long} > 0.001$ to avoid slow convergence.

The sampling ratio is only dependent on $\hat{\alpha}$ for all the combination of $\alpha_{short}$ and

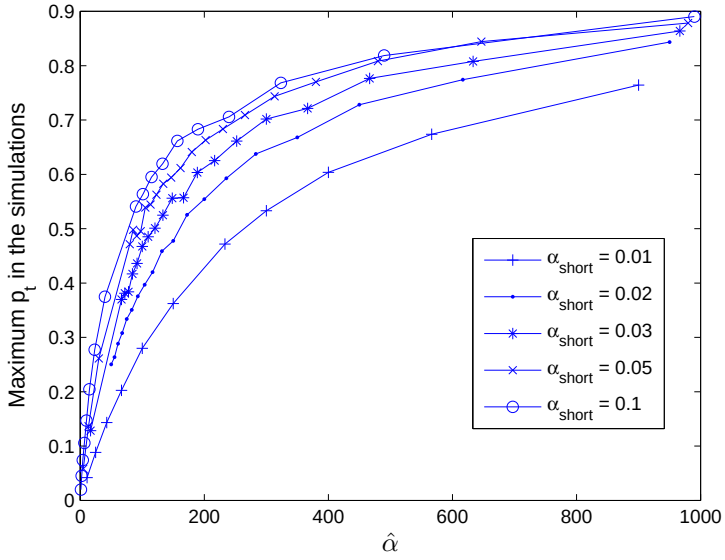



Figure 4.5: Maximum $p_t$ in the simulation

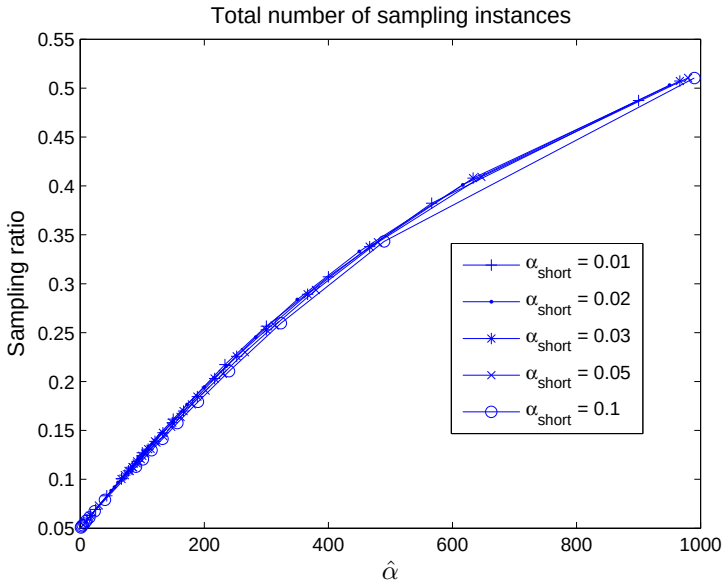



Figure 4.6: Sampling ratio is defined as the total sampling instances over the number of baseline samples

$\alpha_{long}$ that we experimented on, as illustrated in figure 4.6.

### 4.4.2 Curve Shifting Parameter $s$

The parameter $p_t$ should depend mainly on the statistical averages $E_{long}$ and $E_{short}$, provided that $\alpha_{long}$ and $\alpha_{short}$ are chosen appropriately. We have a look at the relation of $p_t$ and both $E$ in this section.

From figure 4.7, the linear section of $p_t$ is positioned at around $E' \sim 100\%$ ($p_t \sim 0.5$ when $E' \sim 100\%$) but this sensitivity setting of $p_t$ may not be suitable to all applications. Hence, we modify equation 4.3.2 such that we can shift the curve according to our requirement, and produce a new parameter $\hat{p}_t$ as shown in equation 4.4.2 with adjustable sensitivity. Note that equation 4.4.2 is equal to equation 4.3.2 when $s = 1$, $v = 1$ and $|E_{short}| >> |E_{long}|$ (or $|E_{short} - E_{long}| \sim |E_{short}|$), which means that there is a significant change in the readings. Note that the probability is maintained at $0 \leq \hat{p}_t \leq 1$ during the shifts.

$$\hat{p}_t = 1 - \frac{|E_{long}|}{|E_{long}| + s \cdot |E_{short} - E_{long}|^v} \tag{4.4.2}$$

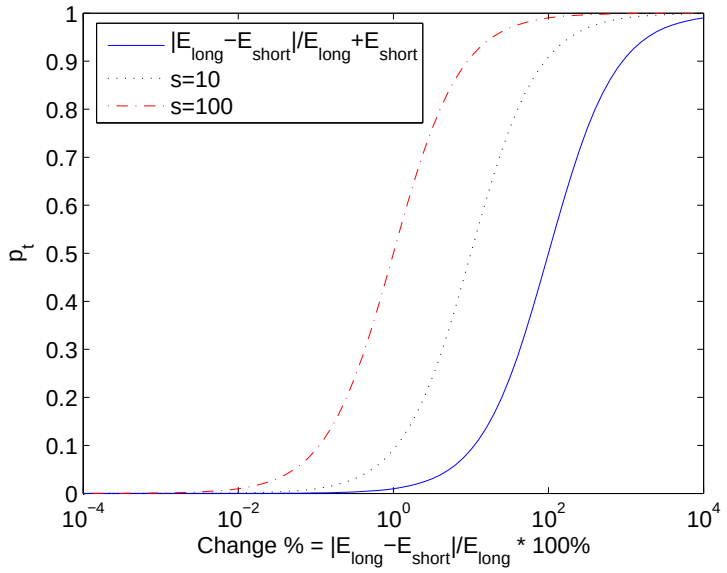



Figure 4.7: Curve Shifting using Equation 4.4.2

We can see from figure 4.7 that, the shapes of curves produced by the two equation

are similar, however, by setting $s > 1$ the linear region of the curves have been shifted to the left. The parameter $v$ controls the slope of the linear section. The larger the value of $v$, the steeper is the curve. We choose $v = 1$ for simplicity of implementation. The linear section of the curve can be shifted to the centre at 50% by setting $s = 10$.

We experimented on different values of $s$ on the salinity data as analysed in section 3.4.1, and plotted the result in terms of missed ratio and sampling rate as shown in Figure 4.8. We can see that as $s$ increases, the number of missed events decreases. The rate of decrease is the greatest between $s = 1$ to 10, whilst little improvement shows for $s > 80$. The number of samples taken has a much slower rate of increase than the missed ratio. Hence, for salinity data, $10 < s < 20$ gives the best missed event to sampling rate relation.

In figure 4.9 we experimented on the sediment data as analysed in section 3.4.3. The variation of sediment concentration is much higher than salinity. Hence, the improvement of missed ratio is the greatest between $1 < s < 20$ because the rate of increase of sampling is slower than the decrease of missed ratio between these values.

### 4.4.3 Adaptivity in BASS

In the last section, we have seen that $s$ is related to the amount of variation in the data, and that increasing $s$ would improve the missed events ratio but increase the sampling rate. Moreover, the optimal $s$ differs for each data set. In this section, we modified BASS so that it adapts the value of $s$ based on the measured feedback values of the environment.

Curve shifting parameter $s$ is adapted differently for capturing the relative threshold and the absolute threshold. For the relative threshold, $s$ is adapted iteratively based on the absolute difference of the last and current measured value.

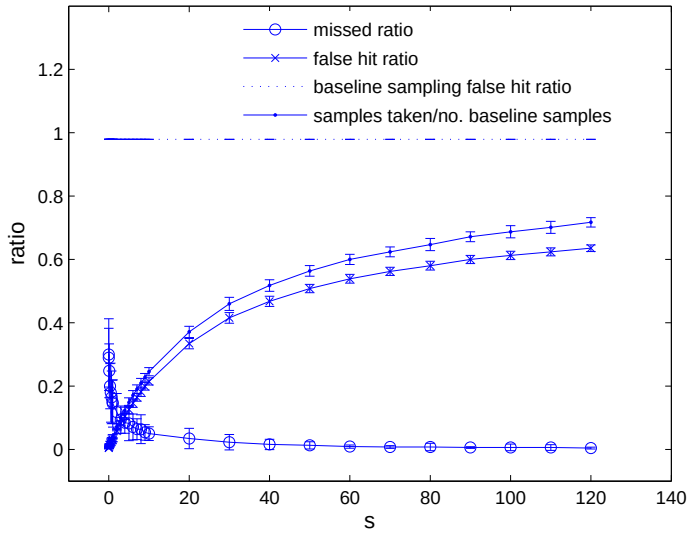



Figure 4.8: Relation of missed ratio, sampling rate and s using Salinity data. Results are averaged over 100 simulations. $k_1 = 0.01$, $\alpha_{long} = 0.001$, $\alpha_{short} = 0.1$, relative threshold= 0.5.

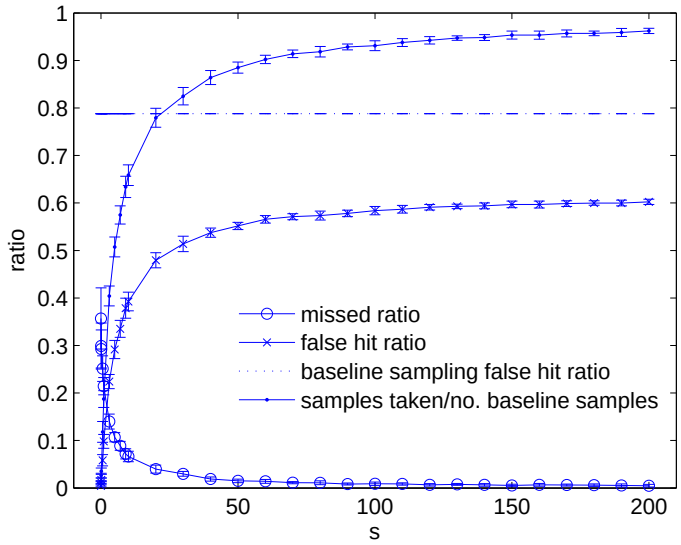



Figure 4.9: Relation of missed ratio, sampling rate and s using Sediment concentration data. Results are averaged over 100 simulations. $k_1 = 0.01$, $\alpha_{long} = 0.001$, $\alpha_{short} = 0.1$, relative threshold= 20.

Hence,

if $|x(t-1) - x(t)| > TH * \rho$

$$s(t+1) = s(t) \times 2$$

else

$$s(t+1) = s(t)/2$$

where $\rho$ is an adjustment parameter for the sensitivity of adaptation and is a percentage of the relative threshold. A lower $\rho$ means the algorithm is more sensitive to change in readings. It is set to between 0.4 to 0.6 in our experiments to obtain a comparable value to eSense. $s_{min} < s < s_{max}$ and $TH$ is the relative threshold.

The mechanism for capturing the absolute threshold requires a different mechanism than capturing relative thresholds. eSense uses the same algorithm for capturing absolute thresholds, which is based upon the calculated probability of the future data crossing the threshold line. This is similar to the relative case but is required to react more quickly. Hence, we tune the sensitivity parameter $s$ based on how close the latest measurements ($E_{short}$) are to the threshold. Hence,

$$s(t+1) = s(t) * \left| \frac{(TH_{abs} - E_{long})}{(TH_{abs} - E_{short}} \right|, \quad s_{min} \leq s \leq s_{max} \tag{4.4.3}$$

Therefore, $s$ increases when the data approach the threshold value and decreases when the trend moves away from the absolute threshold.

The final algorithm is presented in the pseudo-code in table 4.1.

## 4.5 Performance Analysis

In this section, we apply the adaptive sampling algorithm to the oceanography data set described in chapter 3 for the comparison of BASS with eSense adaptive algorithm. We only compare the temporal aspect of BASS with eSense as the eSense algorithm does not have spatial coordination.

In addition, we compare the results to periodic/regular sampling. We obtain the equivalent sample rates from both algorithms and translate it to the closest regular sampling rate. The translation may not be exact, for example, if the average sampling interval obtained from one of the algorithm is 2.5s [1], we can only achieve

[1]This is calculated by dividing the number of samples taken by total number of samples in baseline samples

**Psuedocode:**

*Initialisation:*

$\alpha_{long} \leftarrow 0.001$
$\alpha_{short} \leftarrow 0.1$
initialise $k_1$
initialise $s_{min}$ and $s_{max}$
$s = s_{init}$
$E_{long} = x_1$ /*$x_1$ is the first measurement*/
$E_{short} = x_1$

*Iteration*

FOR $i = 1$ to maximum round {
*evaluateNodeStatus*
IF nodeStatus = sample {
*take a sample*
*updateS* }
*updateProbs* }

*Proc evaluateNodeStatus:*

generate random probability $p$
IF nodestatus = idle
{IF $p < p_s$, nodestatus = sample }
ELSE IF nodestatus = sample
{IF $p < p_i$, nodestatus = idle }

*Proc updateProbs:*

$E_{long} = \alpha_{long} * sensorvalue + (1 - \alpha_{long}) * E_{long}$
$E_{short} = \alpha_{short} * sensorvalue + (1 - \alpha_{short}) * E_{short}$
$p_t = 1 - \frac{E_{long}}{E_{long} + s * |E_{long} - E_{short}|}$
$p_s = k_1 + p_t$
$p_i = 1 - p_t$

*Proc updateS:*

IF Relative Threshold {
IF $|sensorvalue - lastMeasuredValue| > TH * \rho$ {
s = s*2 }
ELSE {
s = s/2 }}
IF Absolute Threshold {
$s = s * abs((TH - E_{long})/(TH - E_{short}))$ }
IF $s > s_{max}$, $\{s = s_{max}$ }
IF $s < s_{min}$, $\{s = s_{min}$ }

Table 4.1: pseudo-code of Adaptive BASS

either 1/2s or 1/3s sampling rate for regular sampling. Therefore, we define,

- Upper bound (UB) regular/periodic sampling rate. It takes the lower integral number of baseline sampling intervals and converts it into the sampling rate. Therefore, UB sampling has a higher or equal sampling rate than our input sampling rate.

- Lower bound (LB) regular/periodic sampling rate. It takes the higher integral number of baseline sampling intervals and converts it into the sampling rate. Therefore, LB sampling has a lower or equal sampling rate than our input sampling rate.

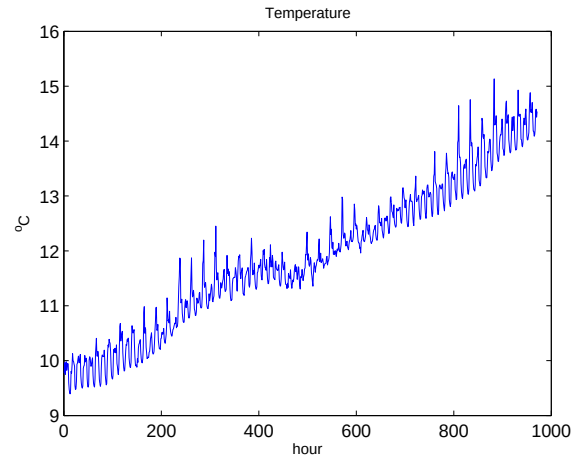
To minimise the error caused by randomness in eSense and BASS, we repeated the simulation 100 times to obtain each result. For regular sampling, the sampling moments and therefore, the starting instance can be a variable contributing to the miss ratio. Hence, we repeat the experiment with different starting instances between 0 and (sampling interval−1) and average the results.

### 4.5.1 Comparing with eSense Algorithm

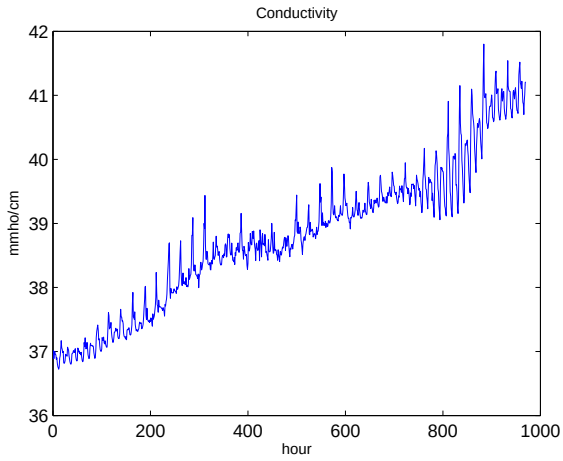
To adopt eSense algorithm for comparison in our experiment, a set of training data is required for the biased random walk model for data prediction [LCS06]. We used the same experimental data as the training data, which should give eSense the optimal results for the algorithm.

### 4.5.2 Relative Threshold

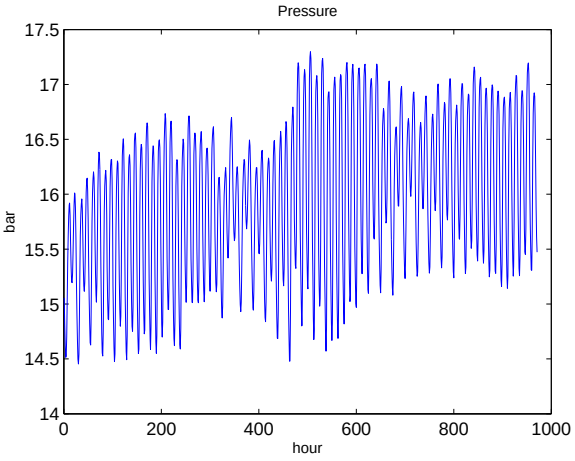
The hourly time series of the four oceanographic data sets are re-plotted in figure 4.10. The data sets have very different characteristics in terms of their periodic behaviours, the rate and level of change and statistical distributions and hence, we can observe the strengths and weaknesses of BASS and eSense in the presence of these different characteristics.



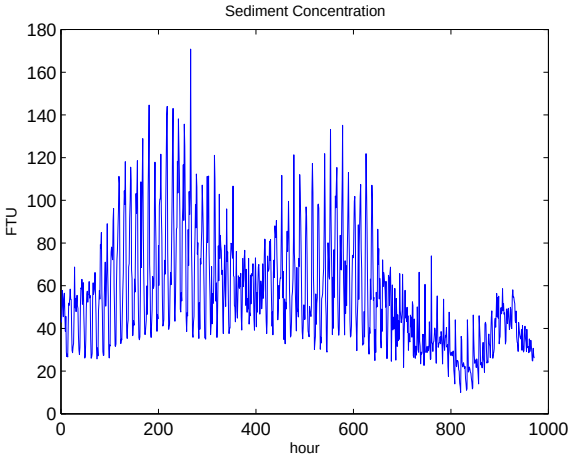

(a) Temperature




(b) Conductivity




(c) Pressure




(d) Sediment

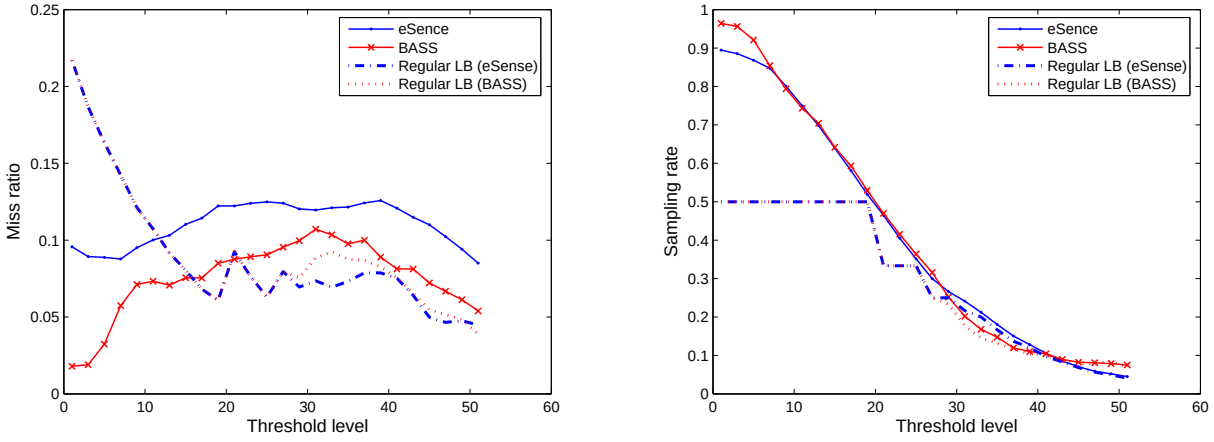
Figure 4.10: Raw hourly temperature data, conductivity data, Pressure data and Sediment concentration

We used the parameter $F_n = 0.1$, $w = 100$ and $\rho = 0.8$ for eSense and with BASS the parameters $\alpha_{long} = 0.001$, $\alpha_{short} = 0.1$ and $k1 = 0.05$ are adopted. The parameter $\rho$ in BASS varies slightly for different data sets so that the resulting sampling rates are comparable to the one of eSense and we can compare their miss ratios.

We re-capture the properties of the two algorithms to help us analyse the results of the experiments in this sections.

- *eSense* calculates the probability of the sample crossing the threshold level at certain time in the future based on a biased random walk model trained by some data. The data are assumed to be Gaussian and the accuracy of the model depends heavily on the training data.

- *BASS* algorithm bases its sampling rate on the rate of change of the data, which is a ratio of the short term and long term averages. The latest event is fed into the system to tune the sensitivity of the sampling probability. If the difference of the current and last events are close to the threshold level, the sensitivity is increased to capture the change.



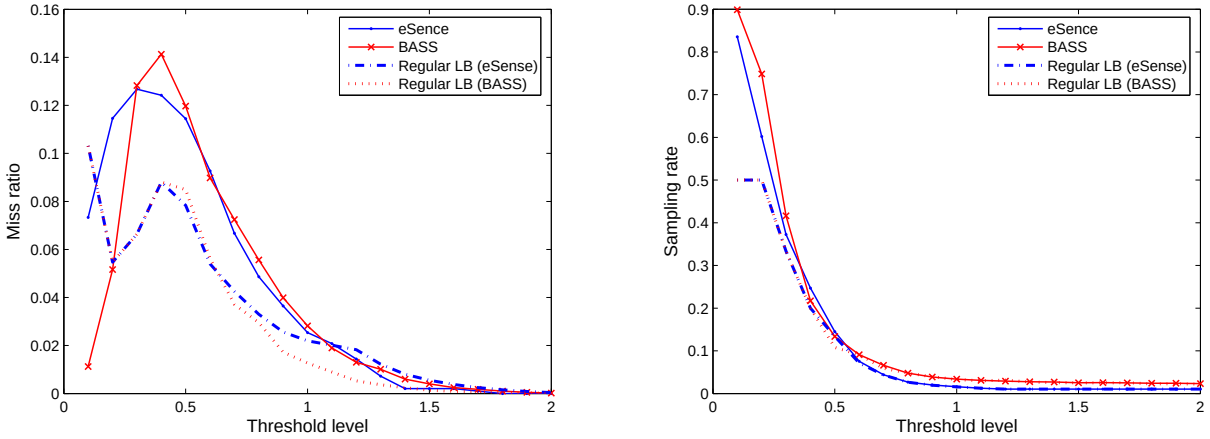

(a) miss ratio

(b) sampling rate

Figure 4.11: Comparison of BASS, eSense and lower bounded periodic sampling on sediment concentration data. $\rho = 0.6$, $0.01 < s < 100$

Figure 4.11, 4.12, 4.13 and 4.14 shows the comparison for BASS, eSense and lower bound (LB) regular sampling with different threshold level. The sampling rate for BASS and eSense is derived from taking the number of sampling instants from the algorithms divided by the total number of baseline sampling instance.

We used LB regular sampling rate as a comparison by taking the larger integral sampling intervals derived from BASS and eSense's sampling rates. Hence, the LB




(a) miss ratio

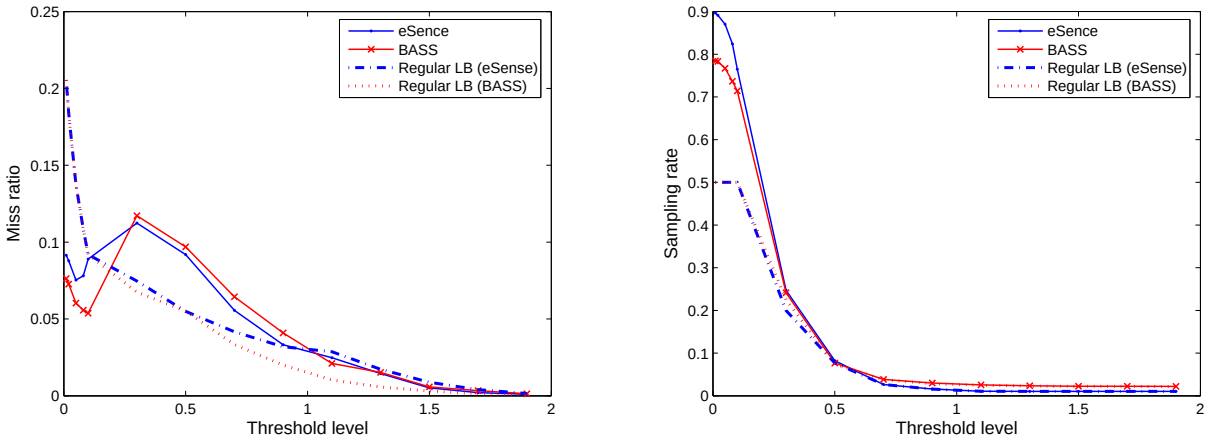
(b) sampling rate

Figure 4.12: Comparison of BASS, eSense and lower bounded periodic sampling on temperature data. $\rho = 0.45$, $0.01 < s < 100$

regular sampling always has a lower sampling rate than (or equal to) the algorithm itself. Note that there is no regular sampling rate between 0.5 to 1 as the sampling interval relative to the baseline sampling is either 1 or 2.

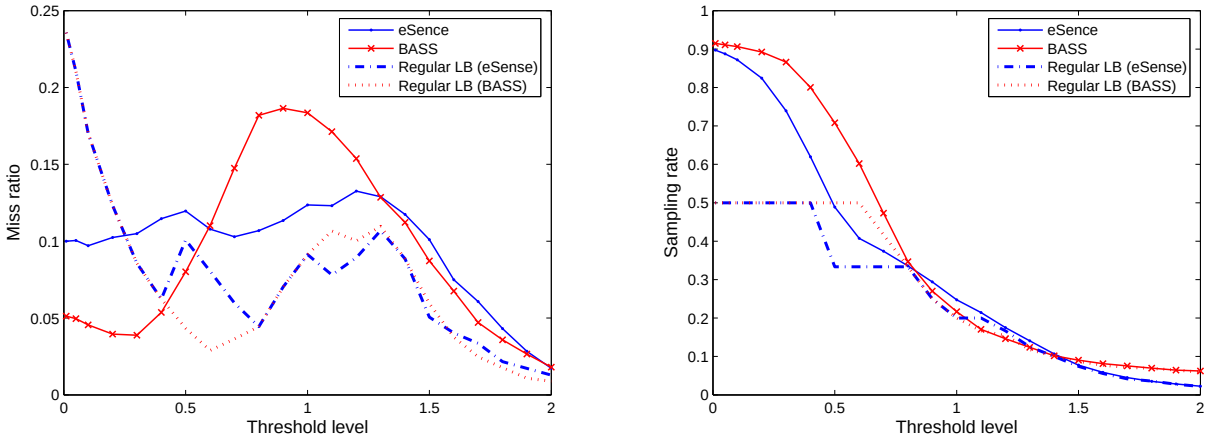
Among the four data sets, BASS performs the best on the sediment concentration data (figure 4.11) as compare to eSense. It gives a comparable sampling rate to the eSense algorithm between 10 to 40 FTUs for threshold level, but BASS has achieved a lower miss ratio below 0.1 for all the threshold levels. eSense has failed to maintain the miss ratio below 0.1 for threshold levels between 15 to 45 FTUs. The major reason for the failure of the eSense algorithm is that the sediment concentration data has the least Gaussian distribution out of the four data sets (discussion of self-similarity characteristic is found in section 3.4.3) and hence, the assumption for a normal distribution is not valid in this case. Moreover, the 2nd moment of the data varies largely over time and hence, the eSense algorithm is not effective using just one normal distribution for the whole series. On the other hand, BASS just measures the rate of change in the data and $s$ is adaptive based on the threshold levels. It has

succeeded in tracking the change of this data series.




(a) miss ratio (b) sampling rate

Figure 4.13: Comparison of BASS, eSense and lower bounded periodic sampling on conductivity data. $\rho = 0.4$, $0.01 < s < 200$




(a) miss ratio (b) sampling rate

Figure 4.14: Comparison of BASS, eSense and lower bounded periodic sampling on pressure data. $\rho = 0.5$, $0.01 < s < 200$

The four sets of results exhibit one common feature for both algorithms. The results appear to be worse when the threshold level is close to the peak to peak

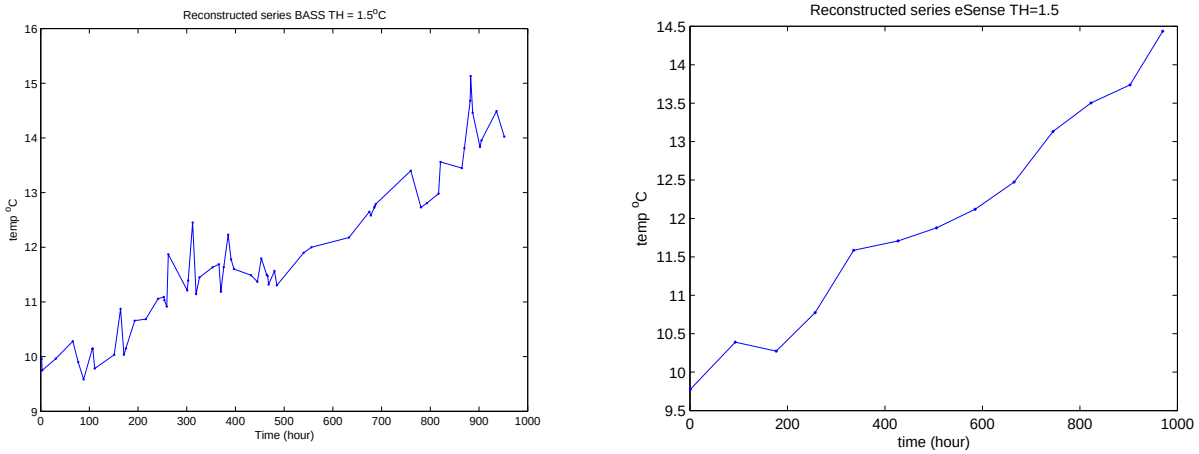
short-term periodic variation. This phenomenon is most obvious in the temperature and conductivity results as shown in figure 4.13 and 4.12. The temperature and conductivity data also have the strongest trends. eSense's sampling pattern throughout the reconstructed series is more even because the same model is used. The adaptivity (the proportion of false negatives) has limited effect on the sampling probability; while BASS's sampling rate varies based on the measured variation in the data. And because of this characteristic in BASS, when the sampler captures the data at non-peak moments during the periods, the difference in the measurements does not reflect the true variation in data and BASS is not as accurate in responding to the true variations in the data.

Both algorithms can capture the trend of the data effectively as shown in Figure 4.15. The difference between the two sampling rates at high threshold levels is largely based on the fact that in our adopted eSense model, the maximum interval between 2 samples is 100, while in BASS, the value $k_1$, which determine the lowest sampling probability, is set to 0.05 (around 1 in 20 samples).

eSense shows better performance than BASS for the pressure time series as shown in figure 4.14 especially for threshold levels between 0.7 and 1.3 bars. For BASS, the poorer performance is for the same reason as for the temperature and conductivity data at around the peak to peak periodic variation. The pressure data is heavily dominated by periods (rather than trends) and this variation is captured by the eSense model.

It is worth noting that when comparing the result of eSense and BASS to LB regular sampling, LB sampling always achieves a lower miss ratio with less sampling instances for these data sets, *except when the sampling rate is between 0.5 and 1.* That is one of the advantages of adaptive sampling as it can adjust the sampling rate between these values. However, if the sampling rate is lower than 0.5, and if we know the required sampling rate to use (which may not be easy), regular sampling

works just as well as, if not better than adaptive sampling for these oceanographic data sets.




(a) Reconstructed BASS series

(b) Reconstructed eSense series

Figure 4.15: Reconstructed BASS and eSense data using Relative $TH = 1.5$ for temperature time series

### 4.5.3 Absolute Threshold

We adopted $\alpha_{long} = 0.01$ and $\alpha_{short} = 0.3$ for a faster response in BASS to capture the absolute threshold. Conductivity data is used in this experiment because it has a strong trend and is similar to what we want to capture in the presence of an event.

The resulting sampling rate for eSense is 0.14 averaged over 100 simulations for conductivity data. The threshold is set at 39 mmho/cm. The missed ratio for BASS is slightly higher at 0.020 mmho/cm and eSense is 0.017 mmho/cm. Both algorithms demonstrate the ability to sample more often when the measured data is close to the absolute threshold level.

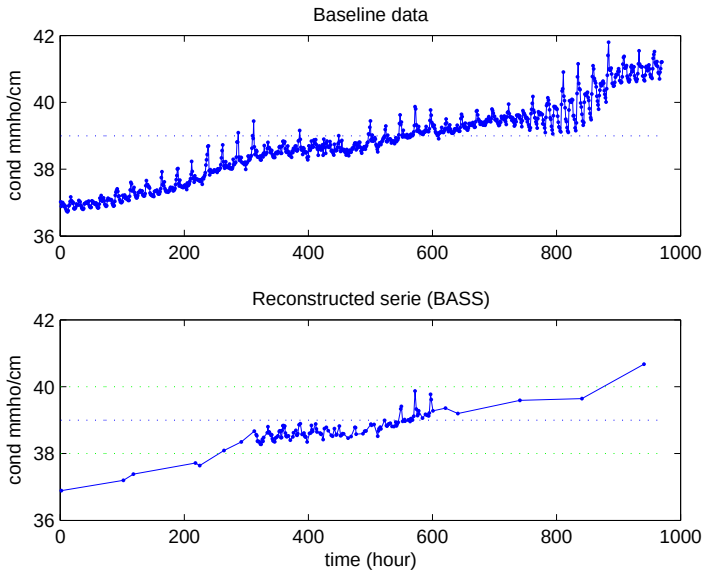



Figure 4.16: Reconstructed conductivity series using BASS with absolute threshold 39 mmho/cm

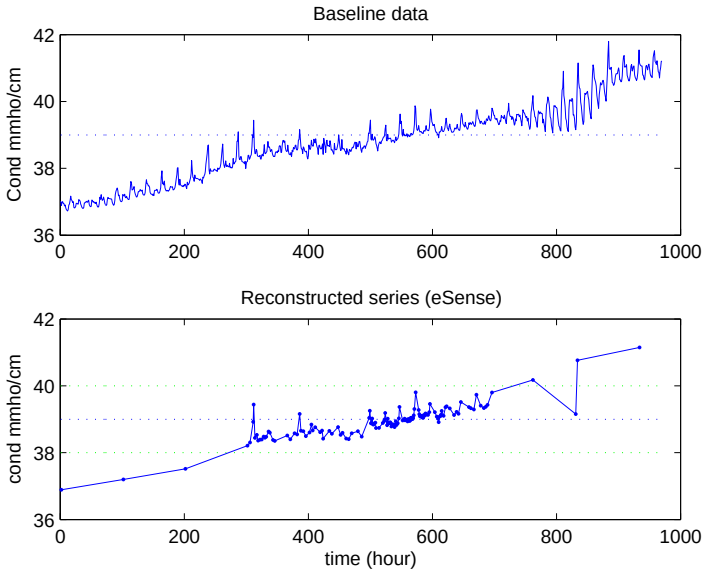



Figure 4.17: Reconstructed conductivity series using eSense with absolute threshold 39 mmho/cm

## 4.6 Conclusion

BASS and eSense are designed to achieve similar objectives via different measures. In eSense, the probability of sampling is directly related to the probability of the

future data crossing the threshold(s). The ratio of false positive events is used as a feedback to fine tune the sampling probability. In BASS, we intend to relate the sampling rate to the rate of change of the data. The sensitivity parameter in BASS is tuned based on the latest measured values and the pre-defined threshold levels.

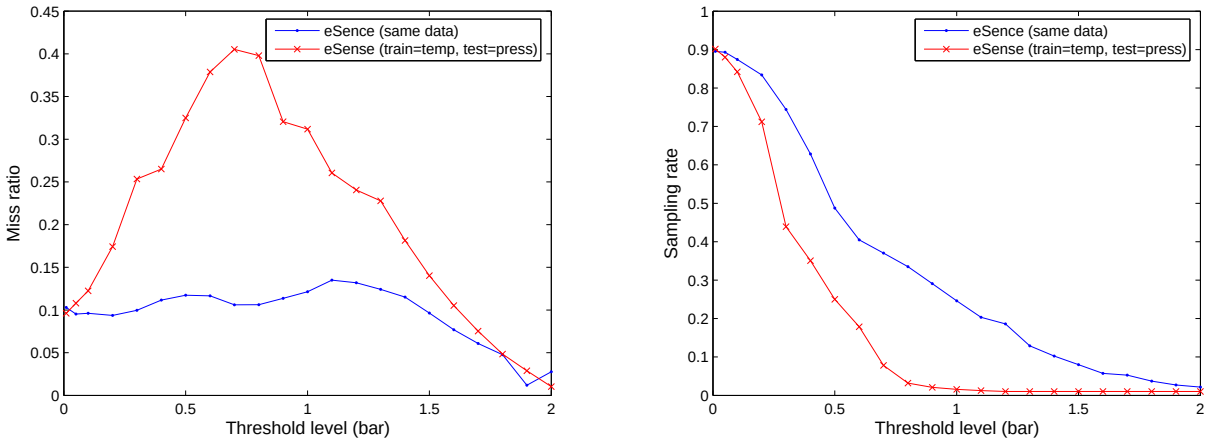
eSense is better at dealing with strongly varying periodic data with little trend as compared to BASS, i.e., the pressure data in our data sets. The biased random walk model can accurately calculate the probability of the future data crossing the thresholds in signals predominated by periodicity.

However, the eSense algorithm has the prerequisite of an accurate model. Hence, we would need to either obtain some training data beforehand, or guess the parameters and fine-tune it on the way. The results can be very poor if the model does not accurately describe the data characteristics. Figure 4.18 shows such an example, using temperature data to train the model and test it on pressure data. However, if the training data has similar features as the tested data, eSense gives very satisfactory results on all data types, providing eSense has the ability to re-train the model to keep track on the variation in the data.

Moreover, the re-training of eSense model can only be done on a node with adequate memory resource and processing power. The model may need to be trained for individual nodes for accuracy due to spatial variation. This means that this super-powerful node would require all the data from all the nodes in the network, which is not a scalable solution for a wireless sensor network. The re-training of the models would also incur significant delays in the network to detect changes in the data.

On the other hand, BASS uses EWMA statistics to capture the change of rate in the data, hence, it should work on most data sets without the requirement of model training. The objective of BASS is to track the changing rate in the data, and hence, BASS does not perform as well in tracking fast periodic changes with peak to peak values close to the threshold levels. It is possible to improve the miss ratio by tuning

the sensitivity thresholds $\rho$, but this would also significantly increase the sampling rate overall. In such applications where the strongly varying periodicity features is to be traced, it is recommended to use eSense, or if the period is known, periodic (or nyquist) sampling would give a better miss ratio when compare to both eSense and BASS.




(a) Miss ratio

(b) Sampling rate

Figure 4.18: The results of the eSense algorithm using the same and different training data from the testing data.

BASS shows especially good results in sediment concentration data in compare to eSense, which verifies the limitation of eSense's prediction model and data feature. On the whole, BASS gives satisfactory results on the four data sets on the detection of both relative and absolute thresholds. Moreover, it has the advantage that very little prior knowledge of the data is required and it works on data with different characteristics.

# Part III

# Topology Control in Wireless Sensor Network

# Chapter 5

# Topology Measurement and Control

Nomenclature used for Topology Control in Chapter 5, 6, 7 and 8.

| symbol | description | equation | page no. |
|---|---|---|---|
| $\eta$ or $\eta_4$ | subgraph number | | |
| $d$ | dimension of Region | – | 140 |
| $\delta$ | normalised density | | |
| $G$ | the Graph $G = (N, E)$ produced by all the nodes $N$ and edges $E$. | | |
| $G'$ | an arbitrary subgraph of $G$ | | |
| $G = (n, E)$ | Graph $G$ contains node set $n$ and edge set $E$ | | |
| $k$ | node degree | – | 139 |
| $l$ | length of the region | – | 140 |
| $n$ | number of nodes | | 140 |
| $r$ | transmission range | – | 139 |
| $r_c$ | critical transmission range | 5.4.1 | 140 |
| $R$ | Region | | |

Network topology is the study of the arrangement or mapping of the elements (links, nodes, etc.) of a network, especially the physical and logical interconnection between nodes. A node in a network will have one or more links to one or more nodes in the network. The mapping of these links onto a graph results in a geometrical

| Notation | Intuition | Definition |
|---|---|---|
| $f(n) = O(g(n))$ | f is asymptotically upper bounded by g (up to constant factor) | $\exists(C > 0), n_0 : \forall(n > n_0)\ \|f(n)\| \leq \|Cg(n)\|$ |
| $f(n) = \Omega(g(n))$ | f is asymptotically lower bounded by g (up to constant factor) | $\exists(C > 0), n_0 : \forall(n > n_0)\ \|Cg(n)\| \leq \|f(n)\|$ |
| $f(n) = \theta(g(n))$ | f is bounded tightly by g asymptotically | $\exists(C, C' > 0), n_0 : \forall(n > n_0)\ \|f(n)\| < \|Cg(n)\| < \|C'g(n)\|$ |
| $f(n) = o(g(n))$ | f is dominated by g asymptotically | $\forall(C > 0), \exists n_0 : \forall(n > n_0)\ \|f(n)\| < \|cG(n)\|$ |
| $f(n) = \omega(g(n))$ | f dominates g asymptotically | $\forall(C > 0), \exists n_0 : \forall(n > n_0)\ \|cG(n)\| < \|f(n)\|$ |
| $f(n) \sim g(n)$ | asymptotically equal | $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 1$ |

Table 5.1: Notation of Computational Complexity used in Part III

shape that determines the physical topology of the network. While the mapping of the *flow of data* between the nodes in the network determines the logical topology of the network.

In a WSN, a node is connected to the other nodes within its radio transmission range. In general, a WSN, or an ad hoc network, would have a meshed network topology with at least 2 multihop pathways to any node in the network. If one of the paths fails, the other is still available. However, a mesh network is not always achievable within a WSN. Other elements like star or linear topology may be found in part of the network because some of the factors contributing to the resulting topology, e.g. placement, mobility, geographic constraints, etc. may not be controllable.

Hence, WSNs are mostly irregular networks. Research on topology control (TC) is introduced to tackle the uncontrollability of the above factors to maintain good connectivity throughout the network at all time, together with the general requirement of power conservation in a wireless sensor network. It is achieved by either tuning the transmission power of the sensor node, or pruning the logical connection for data relaying purpose.

Whilst there are existing TC algorithms that can achieve connectivity in the network, some of them require special hardware to implement, for example, directional antennas, and position estimate. These equipment are not available in the SECOAS sensor nodes. We propose Subgraph Topology Control (STC) algorithm that uses neighbour tables for the calculation of the local connectivity metric, *subgraph number*. The metric has provided additional insight into the network compared to *node degree*, which is used by our compared k-Neighbours algorithm. STC provides better result in terms of the overall connectivity and minimum node degree as compared to the k-Neighbours algorithm, which indicates that the networks formed by STC has a more robust structure. STC is an iterative algorithm, which makes it a very scalable solution.

In this part of the thesis, we investigate TC and report on the design (Chapter 6) and evaluation (Chapter 7) of Subgraph Topology Control (STC) algorithm, and STC's adaptivity in dynamic environment(Chapter 8). This Chapter focuses on the background of topology control including literature research and discussion on the motivation and classification of TC.

## 5.1 Motivations

The major requirement of topology control in WSN is to maintain connectivity in the network. Once the connectivity is ensured, the second goal is usually to reduce the radio transmission power of individual nodes for 2 reasons. The first aim is to reduce the power used for transmitting packets. The second one is to reduce the node degree in the neighbourhood. A sparse network is desirable because it can enhance the performance of the MAC protocol. If a CSMA type scheme is used, low network degree means less probability of contention. If a TDMA scheme is used, slot assignment is easier with fewer nodes and there is less chance of congestion. Moreover, routing is simpler in a sparse network than a dense network because there

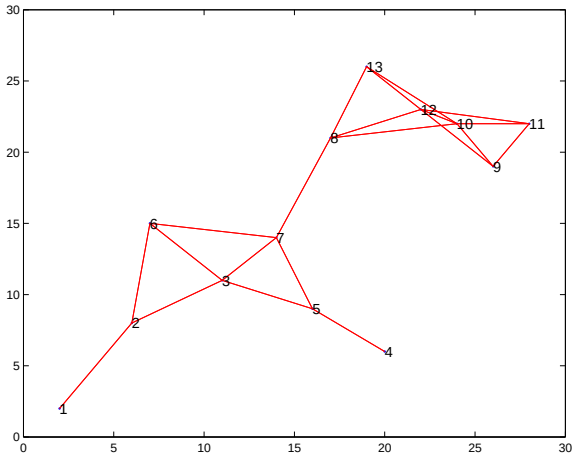



Figure 5.1: A generated example network for the discussion of network irregularity

are less routes to consider.

In addition to these TC specific goal, the common goal to WSN algorithms applies, which includes self-organisation, energy efficiency and adaptivity.

Most existing TC algorithms only have connectivity as their goal. However, we argue that connectivity on its own is not adequate for measuring the health of an ad hoc sensor network. This is obvious in a lot of other wired and wireless networks, as redundancy is always provided to combat node and link failure. Hence, in our work we aim to find a metric that can also represent local network redundancy.

One characteristic of WSN is that nodes are only connected to the geographical neighbours. All we can control, once the network is deployed, is the radio range of the nodes. In general the number of nodes we can communicate with increases with the transmission range, thus, more redundancy is provided because additional routes to the destination are available. Topology control is a research in optimising this radio range, so as to ensure connectivity and provide a certain amount of redundancy, while minimising power usage and resulting node degree.

In an example shown in Figure 5.1, although the network is connected, node 7 and 8, which are connected to 2 separate clusters, are linked by only 1 edge. The removal of this edge would separate the two clusters of nodes. Both node 7 and 8

has node degree of 4 and has redundancy in their own clusters and it is not easy to determine the weakness locally. When visualising the entire network it is easy to pick up weaker locations of the network, however, is it possible for a node to look around its neighbour and conclude the above observation in a distributed environment? This brings out 2 questions, how to measure redundancy in an ad hoc environment effectively and how to achieve it in a distributed manner. Our thesis aims to find out the solution to the above problems and design a topological control algorithm to generate a healthy wireless ad hoc sensor network.

In the SECOAS project the sensing modules are located in the bottom of the sea, and the radio transceivers are floated above the sea level. The radio signal is affected by sea conditions. In stormy weather the communication between nodes can be very unreliable. Moreover, the nodes can drift on the seabed. Hence, a topology control algorithm is required to maintain communications in the network

The SECOAS radio modules are equipped with 8-bit microprocessors, radio chips and sensor modules connecting the measuring sensors to the nodes. There is no physical means of knowing the location of the nodes nor the directional information of the neighbours. This has limited us to use some of the competitive TC algorithms including CBTC and LMST. Moreover, we have also limited a node to communicate with neighbours no more than 2-hop away for scalability purpose. If the algorithm requires the nodes to use information from neighbours that are many hops away, it will take a long time for the network to reach stability and the network would not be able to react to a fast changing environment.

There is a tradeoff between resources and performance and hence, we expect a localised TC would produce sub-optimal result compare to an algorithm that uses global knowledge of the network. We aim to improve our results over the existing algorithm with the same objective. The keys of our algorithm development is to find a representative metric to measure local redundancy and utilise this to generate our

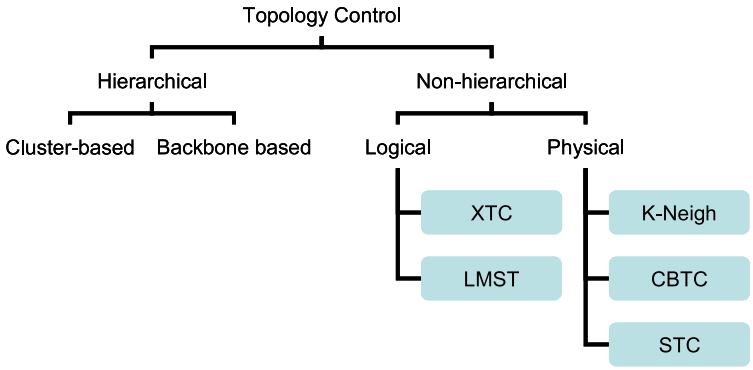



Figure 5.2: Classification and examples of topology control algorithms

global objective of connectivity and power conservation in the network.

## 5.2 Classification/ Taxonomy

Figure 5.2 is a summary of the classification of TC. Some TC base their research on the construction of a *backbone* such as connected dominant set (CDS) [BCar] for efficient communication. Some researches divide the whole network into communication clusters for management and energy conservation [PHC+03]. In our scenario we consider the network has no obvious hierarchy.

The hierarchical and non-hierarchical structures that form our upper categories of TC algorithms are shown in Figure 5.2. We further classify the non-hierarchical category into physical or logical topology controls based on their optimisation objective. In general *physical topology control* determines the optimal *transmission ranges* of nodes, while *logical topology control* determines the optimal *neighbour sets* of nodes. We can view logical TC as 'pruning' a communication graph to remove redundant edges on the graph. The major contribution of a physical TC is to generate a reliable underlying structure for connectivity, while a logical TC focuses on a generating a sparse graph, which can simplify the process of routes finding. Physical TC is also known as the *range assignment* problem, which will be discussed in Chapter 6.

Looking from the algorithmic approaches, a major difference in the various work

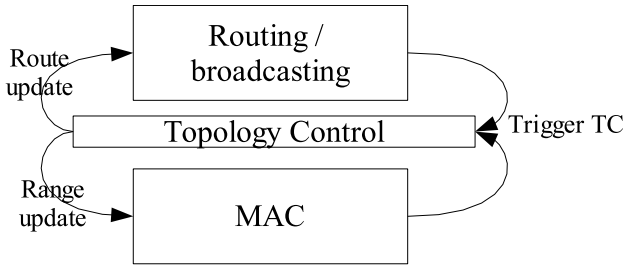



Figure 5.3: Topology control in relation to other communication layers

on TC is the assumption and objective of the algorithms. Some TC research assumes that the maximum graph[1] of the network is connected and that the algorithm optimises the maximum graphs while preserving connectivity. Another group of work does not assume connectivity, but the algorithm aims at generating a networks that is connected with high probability (w.h.p.). Both methodologies have their merits in different sensor network applications. We use the second scenario and do not assume maximum graph connectivity but work on generating a network that is connected w.h.p.

In the networking protocol, TC sits between the MAC layer and the upper level networking protocols as depicted in figure 5.3. A physical TC provides extra reliability in the link layer connectivity, while a logical TC prepares the network for routing and broadcasting optimisation in the networking layer.

## 5.3 Challenges

Although TC in an ad hoc sensor network would improve the overall performance of the network, there are a number of factors that make the design and implementation of the algorithm difficult.

- **The control parameter**

  The major challenge of TC is finding the right measurable parameter for the

[1] A *maximum graph* is formed by all nodes transmitting at maximum power.

algorithm. However, in the SECOAS project some parameters such as location, angle of arrival, etc. are not available in the sensor nodes. Moreover, we cannot use information that requires global knowledge of the network such as path length, hop count to the basestation, etc.

- **Hardware limitation of the radio module**

  This is a practical consideration as to how much we can control the radio module in a sensor node. In some radio modules the transmission power is not a tunable parameter, while in others there are only a couple of transmission level settings so it is difficult for a TC algorithm to make a significant difference on the connectivity and energy performance. However, in our simulation we adopt a radio module that has 31 levels of transmission power from -20dBm to 10dBm. This has given us a chance to develop our TC algorithm.

- **Overheads**

  The cost of a distributed TC algorithm includes the processing power required and the message exchange overheads between neighbours. Hence, we aim to develop a simple TC algorithm that gives good results rather than a very complicated one that gives the best results. Moreover, a simple algorithm can be easily implemented on an MCP board with less processing and memory requirement and this is a very important factor to a WSN.

- **Adaptability**

  TC is required to operate in different geographical terrains and environments. A lot of TC algorithms have only been tested in a 2D symmetric space, however, Real environments are most likely to be not symmetric and can be 1D, 2D or 3D.

  Moreover, TC needs to adapt to node mobility, environmental changes and even terrain changes. The deployment area can be the ocean, in which nodes

may be drifted or washed away by the waves and tides. TC is required to recognise these changes and react correspondingly to maintain connectivity.

## 5.4 Review on the Existing Topology Control Algorithms and Methodology

In the last Chapter the requirements for our TC algorithm was discussed. It needs to preserve network connectivity, be operable on a simple micro-controller and radio transceiver, and adaptive to a changing environment. In this chapter, we review the existing TC algorithms and discuss their suitability for the SECOAS project.

We start with some background literature related to the TC problem. Firstly we introduce the concept of critical transmission range, then move onto the theoretical background of $k$ (node degree) for connectivity. We then look at the generation of some interesting proximity graphs related to the CBTC algorithm.

In this section we evaluate the existing TC algorithms based on its control parameter and control logistic. The control parameter is the heart of a TC algorithm as it defines how accurately a node can measure the local connectivity, which in turn triggers the control activities. We also discuss the suitability of the algorithms in the SECOAS project environment.

Unless stated otherwise, this Chapter uses the following mathematical notations

### 5.4.1 Critical Transmitting Range

A simplified case of physical TC is a study of Critical Transmitting Range (CTR). CTR is a type of homogenous range assignment where every node in the network transmits with the same range $r$ to achieve connectivity. CTR also represents the longest edge in a minimum spanning tree. Given a connected, undirected graph, a spanning tree of that graph is a subgraph which is a tree and connects all the vertices together. A minimum spanning tree (MST) or minimum weight spanning

tree is then a spanning tree with weight less than or equal to the weight of every other spanning tree. In a WSN node density varies in places due to irregular node placement and it is necessary to find the minimum transmission range that generates a connected network taken this randomness into consideration. We have taken the theory of CTR and introduce the concept of *normalised density*, which takes care of the relationship between CTR, density of nodes and the size of the world. This will help us to understand the local redundancy problem in the later part of this thesis.

The problem of CTR for connectivity as a study based on probability is stated as: Suppose n nodes are placed in a certain region $R = [0, l]^d$, with $d$ = 1,2, or 3. Which is the minimum value of r such that the r-homogeneous range assignment is connecting? [San].

CTR has also been studied in other more complicated contexts including mobility [San], k-connectivity [WY04] and connectivity with Bernoulli Nodes [San05a]. However, we only study the fundamental theory, CTR in Sparse and Dense Network, in the SECOAS context and use it as a step to understand the baseline of TC algorithm.

**Dense Networks**

This was the first CTR problem studied. CTR is deduced in a define space $[0, l]^2$ with $n \to \infty$ and has led to the following result by Penrose.

**Theorem 5.4.1.** *[Pen97]*

*Assume n points are distributed randomly with uniform distribution in the unit square* $[0, 1]^2$*, and let* $M_n$ *be the random variable denoting the length of the longest MST edge built on the n nodes. Then*

$$\lim_{n\to\infty} P[n\pi(M_n)^2 - \log n \leq \beta] = \frac{1}{\exp(e^{-\beta})} \tag{5.4.1}$$

*for any* $\beta \in \Re$

Based on this theorem and its derivatives, we can deduce the theoretical values of $r_c$ in different dimensions.

In 1-D space, $r_c = \frac{logn}{n}$

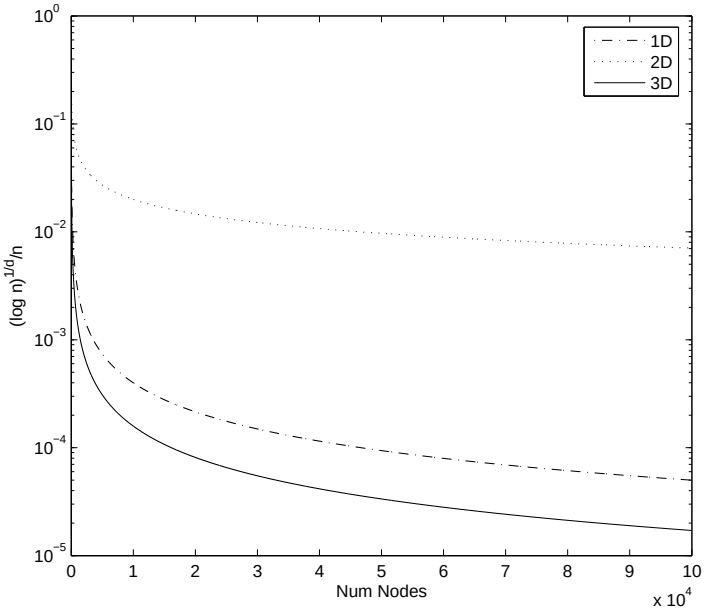



Figure 5.4: A plot of the theoretical critical transmission range against number of node in a sparse network based on the derivative value of Theorem 5.4.1

In 2-D space, $r_c = \sqrt{\frac{logn+f(n)}{n\pi}}$

In 3-D space, $r_c = \sqrt[3]{\frac{logn-loglogn}{n\pi} + \frac{3}{2}\frac{1.41+g(n)}{n\pi}}$

We plotted the above relation in Matlab in figure 5.4 for different $n$. It is observed that CTR is $\Theta(\sqrt[d]{\frac{\log n}{n}})$, (refer to Table 5 for computational complexity) which means that CTR generally decreases with the number of nodes $n$ but with a much slower rate as $n$ increases. We can also see that the CTR for 1D network is much greater than 2D and 3D network for any value of $n$. This is due to the giant component phenomenon.[2]

**Sparse Networks**

The above theorem only works on a dense network on $n \to \infty$, however, this is not the case for most ad hoc wireless sensor network. Hence, Santi et al. proposed

[2]Giant component is a network theory term referring to a connected subgraph that contains a majority of the entire graph's nodes. Percolation theory is based on adding nodes and connections to an empty graph until a giant component surfaces. [Pen99]

to investigate the CTR problem in a sparse network by using $l$ as an independent variable and looked for the asymptotic values of $r$ and $n$ for connectivity w.h.p. as $l \to \infty$.

**Proposition 5.4.2.** *[San05a]*
*If $R = [0, l]^d$, with d = 2,3, and n nodes are distributed uniformly at random in R, the CTR for connectivity is*

$$r_c = \sqrt[d]{k\frac{l^d logl}{n}} \tag{5.4.2}$$

*where k is a constant with $0 \leq k \leq 2^d d^{\frac{d}{2}+1}$*

Details of the proof can be found in [San05a]. Similar to the result obtained for dense network, $r_c$ grows in the order of $\sqrt[d]{l^d \log l}$ as $l \to \infty$.

**Normalised Length and Critical Density**

To help us explain k-Neighbours and the density-based metric Subgraph Number used in our TC algorithm, we have introduce the concept of *normalised length* and *critical density* in this section. The equation of critical density is derived from proposition 5.4.2.

Theorem 5.4.1 shows us the CTR for a dense network as $n \to \infty$ and proposition 5.4.2 shows us the relation of $r_c$, $l$ and $n$ as $l \to \infty$ in a sparse network. These two equations are not dissimilar. As $n \to \infty$, or $l \to \infty$, the boundary effect on the network becomes minimal. Another observation is that in both cases, the size of the world $[0, l]^d$ in relation to the transmission range $r$ becomes infinity, which is also necessary for the derivatives of asymptotic values when boundary effect can be ignored.

Hence, we say that the size of the world $[0, l]^d$ is a relative concept and should always be considered in conjunction with the transmission range of the radio. Hence, we introduce *normalised length* $l_n = l/r$ and modify proposition 5.4.2 to obtain the following asymptotic results for a sparse network.

**Proposition 5.4.3.** *Assume $n$ nodes, each with transmitting range $r$, are placed uniformly at random in $[0, l]^d$, with $d = 2, 3$ and assume that*

$$r^d n = kl^d log(l/r), \tag{5.4.3}$$

*for some constant $k > 0$. If $k > d2^d d^{d/2}$, the resulting communication graph is asymptotically almost surely (a.a.s.) connected.*

The proof is similar to the one given in [San05b]. Rearranging the equation 5.4.3, we get

*If $R = [0, l]^d$, with $d = 2, 3$, and $n$ nodes are distributed uniformly at random in $R$, critical density for connectivity is,*

$$\delta_c = \frac{n}{{l_n}^d} = klogl_n \tag{5.4.4}$$

*where $k$ is a constant with $k < d2^d d^{d/2}$ and $l_n = l/r$, is the normalised length.*

We see that the critical density for connectivity $\delta_c = n/(l_n)^d$ is in the order of $\log l_n$ based on the theoretical results. Since $l$ can be represented in the order of $r$ and $l = \hat{k} * r$, $klogl_n$ is reduced to $\hat{\delta_c} > d2^d d^{d/2} * log\hat{k} >> 1$, which means that it would take a lot more than one node per unit coverage to achieve connectivity. This is of course due to the probabilistic nature of the calculation. In reality if we have some control over the position of the sensor nodes, the required density for connectivity should be a lot less.

### 5.4.2 $k$ for connectivity

Node degree $k$, the number of edges incident to the vertex (a graph theory reference), is used in a number of topology control literature [BLRS03][BLRS06] due to its simplicity. The metric can be easily obtained by each node recording the number of 1-hop neighbours within its receiving range in a fully distributed manner. As $k$ is a density-based metric and its information requirement is similar to our proposed control parameter Subgraph Number, we put more emphasis in discussing its theoretical

background in this section. k-Neighbours, which uses $k$ as the control parameter, is used as a comparison to the performance of STC in section 7.4.

A k-Neighbours graph is a type of proximity graph introduced in the theory of computational geometry which is based on the proximity relationships between nodes. They are called proximity graphs because the set of links incident in any node $u$ of the computed graph can be calculated on the basis of the position of the neighbour nodes. Thus, proximity graphs can be constructed in a fully distributed and localised way.

k-Neighbours, introduced by Santi et.al., is one of the first algorithms to use $k$ as the control parameter in topology control [BLRS03]. k-Neighbours is intended to develop an interference bounded TC algorithm using $k$. Their goal is to generate a graph $G_k$ such that $G_k$ is strongly connected with high probability (w.h.p.).

We can deduce the theoretical minimum $k$ by using the equation of critical density as in eqn.5.4.4 as follow.

Providing the radio transmission range $r$ is a symmetric line (1D), circle (2D) or sphere (3D), the number of neighbours required for connectivity is,

- 1D space,

$$k_c = 2 \times \frac{n}{l/r} = 2klogl_n \tag{5.4.5}$$

- 2D space,

$$k_c = \pi \times \frac{n}{(l/r)^2} = \pi klogl_n \tag{5.4.6}$$

- 3D space,

$$k_c = \frac{4}{3}\pi \times \frac{n}{(l/r)^3} = \frac{4}{3}\pi klogl_n. \tag{5.4.7}$$

The above equation gives us the critical $k$ for connectivity dependent on $l_n$. On the other hand, Xue and Kumar [XK04] has developed another set of results for $k$ for connectivity in terms of $n$. They have also found the explicit values for $c_1$ and $c_2$

to be $c_1 = 0.074$ and $c_2 > 5.1774$ [XK04]. Since $k{=}\Theta(logn) = o(n)$, its asymptotic value for $n \to \infty$ is independent on $n$, and thus, $k$ is interference bounded regardless of the number of nodes. Hence, the theoretical optimal $k$ value for $n = 100$ is $0.34 < k < 23.8$. Recently, the upper bound of $c_2$ has been improved to $\beta e \log n$. [WY04].

Santi et. al. has further evaluated the value of $k$ by simulations [BLRS03]. They found that $k = 9$ produces a symmetric graph which is connected with probability at least 0.95 for values of $n$ in the range 50-500. If the requirement relaxes to only 95% of the nodes are connected in the largest component in the graph, then $k = 6$ is sufficient due to the giant component phenomenon.

They also found that the preferred value of $k$ is determined only by the number of nodes $n$ in the network and not by the size of the region $[0, l]^2$ in which the nodes are deployed. This is consistent with the calculations in Section 5.4.1. In fact, as $(n, l) \to \infty$, $k$ should be independent of both $n$ and $l$ as the critical density, as defined in Section 5.4.1, becomes asymptotically independent of $n$ and $l$, $k = \Theta(Clogn)$ or $k = \Theta(Clogl)$, as the network grows. As a result, using $k$ as the parameter can generate an interference bounded topology connected w.h.p.

Node degree $k$ represents local normalised density and indicates where the network is more or less dense. By keeping $k$ as a constant the variation of node density is compensated by the change of the transmission range. However, the optimal value of $k$ is not necessarily the same for different dimensional networks (1D, 2D or 3D). We have performed a simple k-Neighbours simulation looking at the requirement of $k$ for connectivity against different network shapes, keeping the size and density of the simulated environment constant. The relationship between $k$ and the fraction of connected networks is plotted in Figure 5.5. With the same value of $k$, the fraction of connected networks is less for 1D than 2D or 3D networks. Hence, the requirement of $k$ for connectivity w.h.p. is higher for 1D networks. This is because of the

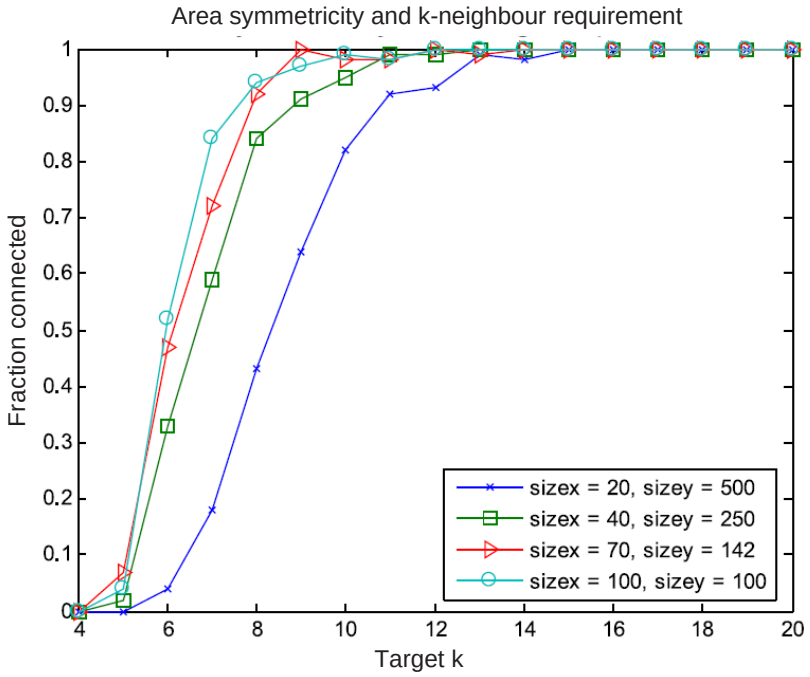



Figure 5.5: Relation of percentage of connected topology $G_k^-$ and different value of $k$. Each data point is the average result of 100 simulations

*giant component* phenomenon in 2D and 3D networks. Therefore, for networks with irregular terrain the preferred value of $k$ should vary accordingly.

### 5.4.3 Existing Topology Control Algorithms

In this section we evaluate several popular TC algorithms by their control parameters and the performance of the algorithms in terms of preserving network connectivity and their suitability in deploying in the SECOAS project. Out of the several approaches for topology control, we place emphasis on evaluating k-Neighbours as it does not utilise special hardware and has similar implementation requirements as our proposed algorithm. k-Neighbours is used as a performance benchmark to our algorithm in Section 7.4.

Physical TC can also be seen as a *range assignment* problem to find out the minimum power cost in the network to maintain network connectivity. Mathematically, we consider a set of $n$ points in a d-dimensional region denoting the node positions

with transmission powers $u$. The problem is to determine a connecting range assignment (RA) of minimum energy cost, such that $\sum (RA(u))^{\alpha}$ is minimum. The algorithm should be distributed. Hence, each node would have to make local decision not knowing the current network state.

Cone-Based Topology Control (CBTC) [LHB+05] is based on an angular separation parameter $\alpha$, and distance estimate between the nodes and the neighbour. The fundamental concept of CBTC is that, a node $u$ tries to find the minimum power $p_{u,\alpha}$ such that transmitting with $p_{u,\alpha}$ ensures that in every cone of degree $\alpha$ around $u$, there is some node that $u$ can reach with power $p_{u,\alpha}$. It was proven in [LHB+05] that if $\alpha \leq 5\pi/6$ then connectivity is preserved. CBTC also has an optimisation stage (logical TC) to identify energy inefficient stages. CBTC guarantees connectivity provided that the network is connected when all nodes are transmitting with their maximum power. However, the major problem of implementing CBTC is the requirement for directional information, which may not be available in common sensor nodes.

XTC [WZ04] is a logical TC that aims at generating a graph optimised for routing. It uses a general control parameter "link quality" of neighbours, which can be signal attenuation, Euclidean distance or packet arrival rate to evaluate the quality of a neighbour connection. The operation of XTC starts with neighbours ordering by their link quality. The ordering is then broadcasted to all the neighbours so all nodes will have a copy of this information about their neighbours. Then every node chooses their edges according to this local information. XTC is a simple algorithm that preserves the connectivity of the original graph and does not require special hardware. However, it is a logical TC algorithm and has different objective to our proposed algorithm.

Local Minimum Spanning Tree (LMST) designed by Li et al [LHS05] is another logical TC that chooses energy efficient edges in the final communication topology.

The concept is similar to finding the minimum spanning tree (MST) for a graph, except that the trees are constructed locally using direct neighbours within a node's maximum transmission range $R$. It was found that a topology constructed using LMST has a maximum degree of six and network connectivity is preserved [LHS05]. However, its major disadvantage is the requirement for location information. Although the author proposes that the location requirement can be substituted by nodes estimating the distance to all the visible nodes and then exchanging the list, this solution involves a lot more overheads and is less scalable. Moreover, a tree-like structure is not desirable in WSN as it increases the network diameter (the maximum of the shortest path between any two nodes) and is vulnerable to a single point of network failure.

The k-Neighbours protocol [BLRS03] is based on the control parameter, node degree $k$ as discussed in section 5.4.2, with the additional distance information estimated by radio signal strength or time of arrival. k-Neighbours is a physical TC algorithm for generating networks connected with high probability (w.h.p.). The basic algorithm requires initially that every node broadcasts its ID at maximum power. Upon receiving broadcast messages from other nodes, every node keeps track of its neighbours and estimates the distance associated with them. The nodes then computes its k-closest neighbours and these becomes their k-Neighbours list. The nodes exchanges their neighbour lists at maximum power and hence, each nodes would know the symmetric neighbours in the neighbourhood. Unsymmetrical neighbours are deleted. k-Neighbours has also proposed a logical optimisation stage to remove energy inefficient edges. Simulation results show that $k = 9$ is required for 95% connectivity [BLRS03] for $50 - 500$ number of nodes.

k-Neighbours is a simple algorithm that does not require special hardware but does not guarantee connectivity. It is also degree-bounded by $k$. The number of messages exchanged in each update is exactly $2n$, where $n$ is the number of nodes.

However, the initialisation and update of k-Neighbours involve all the nodes broadcast at maximum power, which can cause serious contention problem if CSMA is used for medium access.

## 5.5 Chapter Summery

In this Chapter, we reviewed the background of topology control (TC). TC is a distributed way of controlling the network topology using local knowledge of the nodes to preserve network connectivity, minimise average node degree and conserve power.

In general physical TC algorithms determine the optimal transmission range of nodes, while logical TC algorithms determine the optimal neighbour sets of nodes. We consider mainly physical TC solutions in the SECOAS project. The control parameters are used by TC algorithms to evaluate local connectivity and may require the sensor nodes to equip with special hardware for the information.

Critical transmission range is the minimum homogenous range in the network to achieve connectivity. We argued that the transmission range, number of nodes and network size were all related factors in a WSN and introduced the concept of critical density for connectivity, which is also related to $k$ (node degree) for connectivity, that k-Neighbours is based on, and subgraph number $\eta$ in our proposed algorithm, .

We reviewed and evaluated several TC algorithms. In the SECOAS project, the sensor nodes are not equipped with location or directional information. Hence, CBTC and LMST are not suitable for the network. XTC is a second stage logical TC algorithm that should be implemented on an optimised physical topology especially in a dense network because of the communication overheads and memory requirement. k-Neighbours is a simple algorithm, that only uses neighbour information (neighbour ID) and distance estimate. However, a topology generated by k-Neighbours does not guarantee connectivity. Moreover, k-Neighbours has different optimal $k$ for different

dimensional networks. We aim to develop a topology that utilises similar information to the k-Neighbours algorithm but has a better performance in term of connectivity and can be used on different dimensional network. We have developed Subgraph Toplogy Control (STC) based on the above requirement, which will be discussed in the following chapters. The performance of STC against k-Neighbours is included in Chapter 7.

# Chapter 6

# Subgraph Topology Control Algorithm

In the last chapter, we evaluated different topology control algorithms and found that some of them require information including location and direction that may not be available in sensor nodes due to hardware restraints. k-Neighbours uses the simplest form of information, which is node ID and the operation is simple. However, k-Neighbours does not guarantee connectivity and its performance degrade with higher node density.

In the light of this, we propose a neighbour-based algorithm that utilises basic node IDs for the control parameter, but has improved performance over the k-Neighbours protocol.

This Chapter begins with an introduction to subgraphs in the local neighbourhood. We propose a new control parameter *Subgraph Number*, which uses 2-hops neighbouring information, and can give more width (see explanation in Section 6.1) to our local views than node degree $k$. Subgraph Topology Control (STC) is then introduced in full detail. We demonstrate the Subgraph Number is a better metric for measuring local connectivity, thus, STC has an improved performance compared to the k-Neighbours protocol. Details of the performance comparison will be shown in Chapter 7.

The control parameter is very important to a TC algorithm. It measures the status of local connectivity and trigger topology control actions. The control parameter should have a direct relationship with the global objective. Since individual nodes only measure their own parameters and exchange this information with their direct neighbours and adjust their transmission range correspondingly, they have a limited view on global connectivity. The nodes need to coordinate with each other and be self-organised.

Most existing algorithms have connectivity as their goal and their control parameters do not measure redundancy and reliability. Hence, in this section, we introduce a new parameter called Subgraph Number, which is developed to represent local network redundancy. The metric is formed by 2 hops neighbours information and hence, it provides extra insight into the network as compare to node degree used by our competitive algorithm k-neighbours. We analyse its suitability for topology control at the end of the section. We then introduce the Subgraph Topology Control (STC) and describe its approach to tackle the TC problem.

## 6.1 Subgraph Number

In graph theory, $k$, the node degree, is one of the parameter that has been examined often to help our understanding of the nature of the graph. $k$ can be seen as a summary of the 1-hop neighbourhood. We look for an alternative parameter to $k$ that can be calculated locally and would give us more information about the local neighbourhood.

The clustering coefficient (CC) $\gamma$, a parameter that can measure local redundancy was studied at first. However, it is not suitable because it is a relative metric. It is the number of neighbour connections in relation to the maximum allowed neighbour links. When a node has small node degree, CC can give a false impression that it is very well connected while the truth is that a group of nodes have formed an *island*

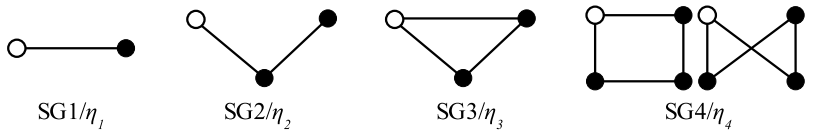



Figure 6.1: Subgraphs 1-4 formed by 1 and 2 hops neighbours. The white circles are the originating node $u$ and the black circles represent 1 and 2 hops neighbours $v_i$ & $w_i$. SG1 represents a *branch* that terminates at node $v_i$; SG2 is an *angular* subgraph extending to the outside world other than the immediate neighbourhood; SG3 is a *triad* where the immediate neighbours are connected to each other; SG4 is a *quadrilateral* and the second order neighbours are connected to each other. All other subgraph formed by 1 and 2 hops neighbours are a combination or breakdown of SG1-4.

that is isolated from the rest of the graph. Moreover, whilst CC represents link redundancy in the neighbourhood, it is not directly related to transmission power, that most 2-dimensional GRG has an intrinsic $\gamma$ value of 0.75. Hence, we cannot use CC to represent connectivity or redundancy and a new metric is required.

The use of subgraph to examine the nature of a graph has been used in literature, such as using *cycle probability* of various length in analysing the structure of neuro-networks in [EG01] and [STE00]; significant subgraphs of length 3 have also been used to analyse network structures in [MIK$^+$04]. Subgraphs formed by nodes that are within 2 hops away can easily be measured locally. We consider the properties of 2-hops subgraphs that give us extra information beyond that of $k$.

We observe that by exchanging neighbouring table with the immediate neighbours, a node $u$ can also obtain a list of its 2-hop neighbours. This extra width of knowledge that can be obtained at very little extra cost, should provide us a more accurate view on the connectivity status within the network than $k$. The subgraph formed by 1 and 2 hops neighbours ($\eta_1$ to $\eta_4$) are shown in figure 6.1.

We explore the idea of using the number of subgraph $\eta$ formed by the links between nodes in the neighbourhood to measure the connectivity of the network. The white node in figure 6.1 is the node $u$ that is counting the subgraphs in the neighbourhood. Note that when we are counting $\eta_1$, we do not include the link in $\eta_2$,

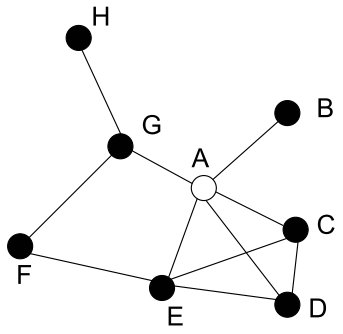



Figure 6.2: An example of subgraphs counting from node A. $\eta_1 = 1$ (AB), $\eta_2 = 1$ (AGH), $\eta_3 = 3$ (ACD, ACE, ADE), $\eta_4 = 1$ (AGFE)

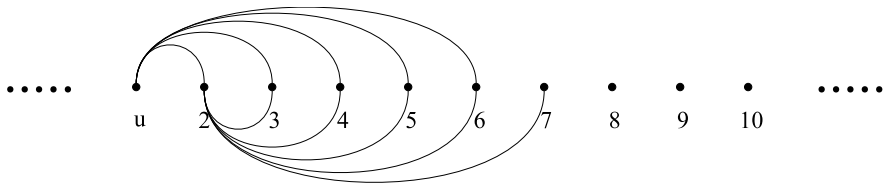



Figure 6.3: A linear network of counting subgraphs with $R = 5$

which extends further into the network. Hence, $\eta_1$ is $k$ minus all the neighbours that are connected to each other. Similarly, $\eta_3$ only includes all the 2 hops nodes that are not 1-hop and are not already counted in the forming $\eta_4$. $\eta_4$ also does not include any quadrilaterals formed by 1-hop nodes. Figure 6.2 is an example of subgraph counting. Note that link AG does not count as a $\eta_1$, and quad ACDE does not count as $\eta_4$.

We have chosen to use $\eta_4$ as the control parameter in our TC algorithm. The extended information obtained by 2 hops neighbours gives us a more accurate view of the network connectivity in the neighbourhood than the 1 hop parameters $\eta_1$ and $\eta_3$. Moreover, the number of $\eta_4$ reflects the amount of redundancy in 2 hops, or the number of 2 hops alternative links to the same neighbour. $\eta_4$ is a density based metric, which can reflect the local node density of the network. Its characteristic will be shown in Section 6.1.2.

| 1-hop | 2-hop | $\eta_4$ |
|---|---|---|
| 2 | 7 | 4 |
| 3 | 7,8 | 4+3 |
| 4 | 7,8,9 | 4+3+2 |
| 5 | 7,8,9,10 | 4+3+2+1 |
| 6 | 7,8,9,10,11 | 4+3+2+1+0 |

Table 6.1: Neighbour table for node $u$

### 6.1.1 Theoretical Results of 1-D network

$\eta_4$ is obtained by node $u$ exchanging neighbour tables with immediate neighbours $N = \{v_1, v_2, ....v_n\}$. The subgraphs formed by 1 and 2 hops neighbours are illustrated in figure 6.1 and $\eta_{1-4}$ is the number of SG 1-4

We use a linear uniform network to derive the theoretical result of $\eta_4$. Figure 6.3 shows the neighbours of node $u$ on one side with $R = 5$. Table 6.1 depicts the neighbouring relationship for node 1. Node 1 has $\Gamma(1) = \{2, 3, 4, 5, 6\}$ 1-hop neighbours and its 2-hop neighbours that do not belong to 1-hop members are node {7,8,9,10,11}. From node 2, node 7 is also connected to 4 other 1-hop nodes (3,4,5) and hence, $\eta_4(1-2-7)$ is 4. From node 3, node 8 is connected to 3 other 1-hop nodes and $\eta_4(1-3-8)$ is 3 and so on. We can fill in all the $\eta_4$ for all the 1 to 2 hop routes. Note that after we have counted all the $\eta_4$ for each route, each SG4 has been considered twice in the table and the final sum needs to be divided by 2 (however, this graph only shows one side of the linear network). We can derive an expression $\eta_4$ for node $u$ in relation to range $R$,

$$
\begin{aligned}
\eta_4 &= \sum_{k=0}^{R-1} k + \sum_{k=1}^{R-1} k + \sum_{k=2}^{R-1} k + ..... \sum_{k=R-1}^{R-1} k \\
&= \sum_{r=1}^{R} \sum_{k=R-r}^{R-1} k \\
&= \sum_{r=1}^{R} \frac{(R-r+R-1)\cdot(R-1-R+r+1)}{2} \\
\eta_4 &= \frac{1}{3}(R^3 - R)\ . \qquad (6.1.1)
\end{aligned}
$$

Equation 6.1.1 shows the relation of $\eta_4$ in relation to transmission range. We can see that in the 1-dimensional case, $\eta_4$ is a degree 3 polynomial of $R$. Hence, the number $\eta_4$ rapidly increases as the transmission range increases.

The above assumes that node density of the network is 1. Hence, there is 1 node in 1 unit range ($R$ nodes in range $R$). Similar to $k$, $\eta_4$ is also a density based metric. It is dependent on the normalised density, largely defined by the number of nodes and the transmission range of the network. Hence, the theoretical result for critical density derived in section 5.4.1 also applies to $\eta_4$ with the relationship of range in equation 6.1.1.

### 6.1.2 Simulated characteristics of Subgraph $\eta_4$

We examine the behaviour of $\eta_4$ with increasing transmission range $R$. In order to compare the results from 1D, 2D and 3D network we plot the number $\eta_4$ against normalised density $\delta = \frac{n}{(l/r)^d}$. Figure 6.4 shows the results of the experiments. We vary the radio transmission range from 0 to 150 units, keeping the number of nodes and size of the space the same. We observed that when normalised node density $\delta$ is less than 55, $\eta_4$ rapidly increases. The results deviate at high $\delta$ as boundary conditions become predominant. Note that $\eta_4$ decreases to zero when a graph is fully-connected. Hence, beyond the point of $\delta = 55$ (60 and 65 for the case of 2D and 3D), $\eta_4$ decreases quickly to zero.

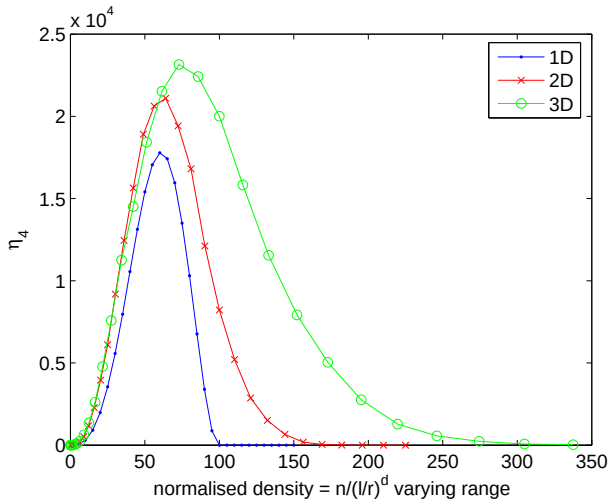



Figure 6.4: Characteristics of $\eta_4$ with increasing range. No. of nodes = 100 and size of the simulation area = $[0, 100]^d$. Each point is the mean of 100 simulations.

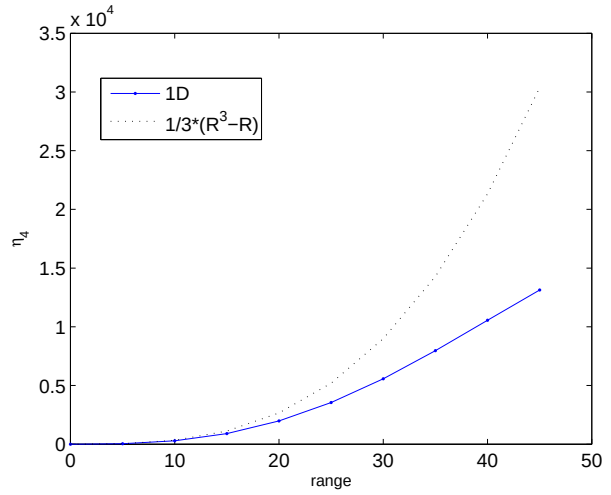
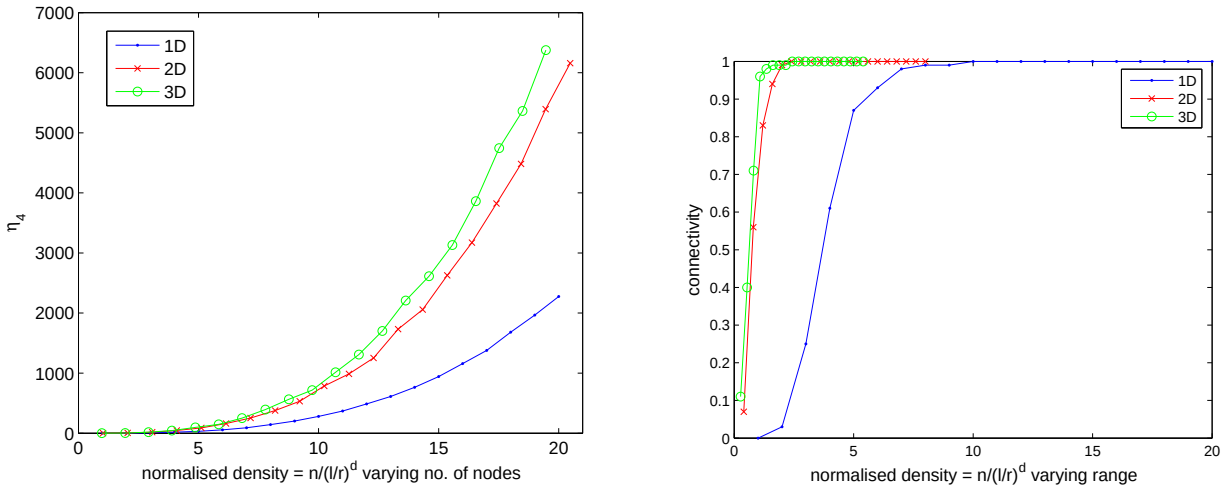



Figure 6.5: Comparison of experimental 1-D $\eta_4$ with increasing range and theoretical result derived in eq. 6.1.1. Each point is the mean of 100 simulations

Figure 6.5 is a comparison of the experimental $\eta_4$ against the theoretical result derived in eq. 6.1.1 with varying range from 0 to 50. At low $\delta$, $\eta_4$ for 1D network follows the theoretical result shown in black dotted line closely. The deviation becomes greater as $R$ increases and the boundary effect becomes more obvious. Boundary effect is only observed in simulation by varying the size of the world or radio range. When the radio range is large, comparatively the simulated environment becomes




(a) (b)

Figure 6.6: Characteristics of (a) $\eta_4$, (b) Connectivity with increasing number of nodes. Transmission range is 50 for 1D, 160 for 2D, and 230 units for 3D. The size of the world is $[0, 500]^d$. Each point is the mean of 100 simulations

small. The asymmetry of the boundary nodes then becomes dominant among all the nodes.

Node density is the varying factor in Figure 6.6(a). The boundary condition does not change throughout the experiment and $\eta$ increases with $n$ and therefore, $\delta$. $\eta_4$ for 2D and 3D graphs are considerably higher than the 1D graph, which is observed in Figure 6.6(b). We can see that the required $\eta_4$ for connectivity for a 1D graph is a lot higher than 2D and 3D graphs. This is due to the *giant component* phenomenon (see Section 5.4.1). Hence, in the case of 2D and 3D graphs the nodes quickly form an entity containing most of the nodes and this boosts the $\eta_4$ number.

We investigate $k$, node degree, which is used in the k-neighbours algorithm. $k$ in Figure 6.7 increases linearly with relative density and is bounded by the size of the network. Thus, the boundary effect will stabilise $k$ at high $\delta$, while $\eta_4$ decreases when the boundary effect becomes dominant. If the graph is of infinite size, $k$ would just increase linearly with $\delta$.

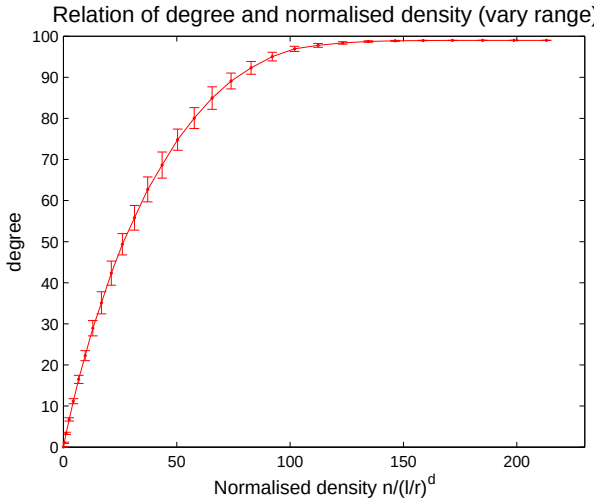



Figure 6.7: Degree and normalised density for a 2D network with increasing range. Network dimension = $[0, 100]^2$. Number of nodes in the network is 100.

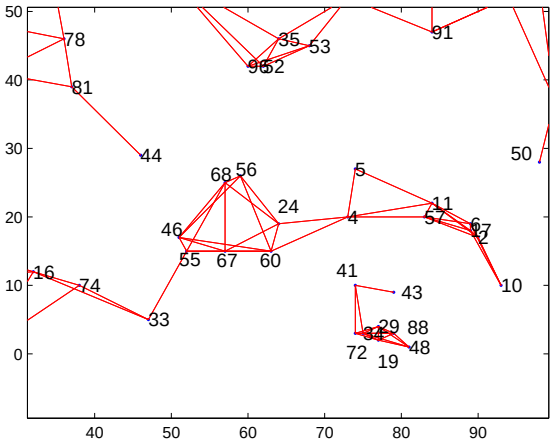



Figure 6.8: Island Phenomenon. Topology generated by k-Neighbours $k = 6$. We can see that node 4 and 41 are in separate clusters. They consider themselves "well connected" and do not extend their radio range to connect to each other, as only the closest 6 neighbours are considered.

### 6.1.3 Suitability for Topology Control

Subgraph number $\eta_4$ is calculated using the neighbour tables obtained from nodes' 1-hop neighbour. It is simple and is derived in a distributed manner. It is a second order metric that contains information of the connectivity 2 hops away.

$\eta_4$ and $k$ both measure the relative density in the network in terms of range

| Parameter | Degree $k$ | Direction | Distance | Location | SC |
|---|---|---|---|---|---|
| Source | Neighbour table | Angular information | Signal Strength | Location | Neighbour table |
| Info. reach | 1-hop | 1-hop | 1-hop | 1-hop | 2-hop |
| Info. of individual neigh | No | Yes | Yes | Yes | No |

Table 6.2: Comparison on the parameters

and node density at the local neighbourhood. Their information only extends to 1-hop for $k$ and 2-hops for $\eta_4$ neighbourhood. In a specific case where a highly clustered group of nodes are positioned next to a very sparse area, these nodes may consider themselves to be well-connected while they are separated from the rest of the network. We call this the *Island phenomenon*. Figure 6.8 illustrates an example of this. Although we cannot prevent island phenomenon with $\eta_4$, as a second order metric, $\eta_4$ stretches further into the network and should show an improvement towards the overall connectivity compare with $k$.

CBTC, LMST and XTC do not have this problem as they assume the original maximum graph is connected and optimised from this graph. It is a fair assumption to make, although we must note that not all sensor networks are connected at maximum graph as the positions of sensor nodes may not be manipulated. Moreover, as $\eta_4$ requires only node ID in its calculation, it is suitable to be used as the control parameter for TC in the SECOAS project.

The number $\eta_4$ rises very quickly with the density $\delta$, however, we observe that in figure 6.6 the requirement of $\delta$ for connectivity for 2-D and 3-D network is about 2-3. The characteristic $\eta_4$ for $\delta = 3$ is about 10-20. Hence, we are looking at a much lower number $\eta_4$ to work with in our TC algorithm.

## 6.2 Characteristics of the STC Algorithm

Apart from the use of different control parameters, the approaches to the development of a TC algorithm can be divided into 2 main streams.

- **Optimised from the maximum graph**

  Algorithms including k-Neighbours, XTC and LMST derive their topology from the maximum graph. They evaluate the 'best' connectivity when all nodes are transmitting at maximum power, and reduce redundancy while maintaining connectivity. When updating, the process is the same as when topology is first derived from the maximum graph.

- **Iterative approach** Either starting from minimum or maximum transmission power, the algorithms apply changes to the transmission power one step at a time and evaluate their local connectivity after each change. The process continue until the network is connected or defined criteria are satisfied. An example of an iterative TC algorithm is CBTC. Nodes exchange messages to detect if there are any changes to the local connections, which would violate the set criteria and initiate local changes, which then spread to further changes in the network.

We adopt the iterative approach in our TC implementation. Optimising from the maximum graph can be problematic in a network. A typical transceiver can have the transmission range of approximately $10m$ to more than $100m$ in an open space. Assume the node density is 1 per $10m^2$, then each node would have just less than 100 neighbours in sight for a maximum graph. This would cause a lot of congestion and interference problems for the radio communication. Moreover, each update requires the construction of a maximum graph, while this method may be simple and incur little delays as suggested in [San], it is not the most energy efficient way to achieve

topology control.

Whilst an iterative approach by definition is likely to take longer to first reach stability depending on the node density, maintaining the topology requires the node only to exchange messages with its neighbours until there is a change of state. The updating of the topology is also localised involving only the nodes within the change. A TC algorithm that starts with minimum broadcast power is more energy efficient especially in a high density network.

Subgraph Topology Control (STC) is designed taking reliability into consideration. Each node evaluates the local connectivity and redundancy, adjusting its radio transmission power to fulfil the global requirement of connectivity, adaptivity, reliability and energy conservation.

Subgraph number (SN) is used as the control parameter to measure the local connectivity. The characteristics and theoretical results of SN are included in Section 6.1. SN utilises the proportion of local subgraphs to evaluate the amount of redundancy in the neighbourhood. The calculation of SN only requires nodes exchanging their neighbour tables.

STC is probabilistic and iterative. Some distance estimation technique is required for a node to match the furthest neighbour's power, however, overall STC is a very simple algorithm and requires minimum hardware configuration to operate. It is a distributed system with individuals making local optimum decisions, hence, it is scalable and adaptive.

## 6.3 Basic Algorithm

The concept of Subgraph Topology Control (STC) is very simple. It uses the subgraph number $\eta_4$ to evaluate the connectivity of individual nodes, and then tries to keep the SN within the thresholds $k_L \leq \eta_4 \leq k_H$. Each node $u$ calculates $\eta_4$ based on its own neighbour tables $\Gamma_u$ and receives neighbours' tables $\Gamma_{v_i}$, where

$\Gamma = \{v_1, v_2 ... v_n\}$ is the neighbourhood of $u$.

STC uses an iterative approach with nodes gradually increasing their power to improve connectivity. Neighbour tables are updated and the process is repeated until $\eta_4$ in individual nodes reaches a pre-defined threshold $k_L$ (which will be discussed in chapter 7, or the nodes are transmitting with maximum power.

The transmission power used for topology control in topology control $P_t$ is different from the one used for communication $P_c$. $P_t$ is higher as a node may want to discover new neighbours, while $P_c$ is only required to be as high as to reach the furthest node in the communication list. $P_c$ is used in our cost evaluation of the algorithms.

The basic transmission of STC uses an update broadcast packet to communicate with the nodes in communication range. A node would transmit an update broadcast packet when there are changes to its circumstances, which includes,

1. A node increases or decreases its transmission power,
2. A node notices a change of states, which includes,
    - addition and removal of immediate neighbours,
    - disconnection or connection to the base station from its neighbours,
    - a change of its transmission power.

An update broadcast packet is received by all the nodes in the transmission range, hence, it is considered to be broadcast. The different types of update broadcast packets to fulfil the requirement of STC topology control are listed follow.

1. **General update broadcast (Type 1)**

    The purpose of the packet is to inform the neighbours that some of the nodes states have been changed. Nodes receiving the packet do not require actions other than to update the information in their neighbour tables, if the sender node is within their communication range. The packet is ignored otherwise.

2. **Increase power request update broadcast (Type 2)**

   This packet includes a request to the nodes in range to increase their power to reach the sender node. In the basic algorithm, nodes which have received the packet are forced to increase their power such that they can communicate with the sender node if they are not already a neighbour of the sender node. The receiver nodes also update the sender's information in the neighbour table.

3. **Decrease power notification update broadcast (Type 3)**

   This packet informs the existing neighbours that the sender node $u$ is going to decrease its transmission power in the next round. After $u$ has decreased its power, it would then send out a Type 1 broadcast to find the neighbours that are still in range.

   Upon receiving the Type 3 packet, node $v$ would remove $u$ from its neighbour table. This would trigger $v$ to initiate an Type 1 broadcast, which would reach $u$. If the $v$ is still in range of $u$ after the power decrease then a link is formed. $v$ will decide whether $u$ is still a neighbour when it receives the type 1 broadcast from $u$ after it has decreased its power.

In the basic algorithm, each node tries to maintain $k_L \leq \eta_4 \leq k_H$. An upper bound is necessary for the nodes to adjust to the dynamic nature of the system. We study the effect of $k_H$ on the stability of the system in Chapter 7.

Initially nodes that have $\eta_4$ below the threshold would increase their power and initiate an increase power request. The receiving nodes will increase their power to communicate with the sender node, if it is not already within the communication range. A node will broadcast a Type 1 packet if there is a change to its states. It will only issue a decrease power notification if its $\eta_4$ is above the threshold $k_H$.

**Pseudocode:**

*Initialisation*

$\Gamma_u \leftarrow \emptyset$ //1-hop neighbours of $u$
$\eta_u \leftarrow 0$ //Subgraph Number of $u$
$P_t = P_0$ //Available transmission power $P_t = \{P_0, P_1, ..., P_{max}\}$;
$p_r$ //Reply probability
Initialise $k_L$ and $k_H$

*Iteration*

FOR $i = 1$ to maximum Round
IF $(\eta_u < k_L)$
*Proc UpdateBroadcast(Type 2)*
ELSEIF $(\eta_u > k_H)$
*Proc UpdateBroadcast(Type 3)*
END
END

*Proc UpdateBroadcast(Type a)* $a = 1, 2, 3$*:*

FOR all $v$ = Neighbour in range
*ReceiveUpdateBroadcast(Type a, u, v)*
END

*Proc ReceiveUpdateBroadcast(Type a):*

IF $a == 2$ //this is an increase power request
$p$ = random number from 0 to 1
IF $p < p_r$
Add $u$ as neighbour, increase power to match if needed.
END
ELSEIF $a == 3$ //this is a decrease power request
Drop neighbour $u$
END

Update $\eta$
Update Status

IF Change to $\Gamma$ or status
*UpdateBroadcast(Type 1)*
END

Table 6.3: pseudocode of STC

## 6.4 Chapter Summary

In this chapter, we introduced the Subgraph Number $\eta_4$ (or $\eta$ in the following chapters) as the control parameter of our TC algorithm. $\eta$ is a measure of the 2-hops redundancy connectivity, the number of *quadrilateral* a node can see. $\eta$ is a density based metric and a degree 3 polynomial of the transmission range. Hence, the number increases rapidly with the number of neighbours. However, the simulated results indicate that we are dealing with $0 < \eta < 200$ for the guaranteed whole network connectivity. When $\eta$ is used in the Subgraph Topology Control (STC) algorithm, we are dealing with the range of $1 < \eta < 10$, which will be elaborated in the next chapter when we look at the performance of STC.

We also discussed the design and characteristics of the STC algorithm. STC is an iterative algorithm that sensor nodes only increase or decrease one power step at a time to adjust their connectivity. The algorithm intends to bound $\eta$ in each node between $k_L$ and $k_H$. An update message is broadcasted if a change occurs in a node or a neighbour. The metric $\eta$ used by STC can represent local density and connectivity. Its iterative and distributed natures result in adaptivity and scalability, which are the prime requirements in the SECOAS project.

# Chapter 7

# Performance Evaluation of Subgraph Topology Control

## 7.1 Introduction

### 7.1.1 Simulation Environment

We adopt the parameters from a Chipcon CC1020 804MHz radio in our simulations with Matlab v7 because of its wide range of transmission power. The programmable transmission power is $P$ = -20 to 10dBm inclusively with an incremental step of $\delta_p = 1dBm$. We use a log-distance path model $P_r(d) \propto \frac{P_t}{d^\alpha}$ to estimate the transmission range to be 3m to 100m with an $\alpha$ value of 2 representing outdoor space and receiver sensitivity of -110dBm. The increments of the range is not linear.

We adopted a Unit-Disk Graph model in a symmetric 2-dimensional area in the experiment unless stated otherwise. The dimension of the simulation environment is $[100, 100]$ metres.

The assumptions are,

- symmetric environment, in the basic simulation, the dimension of the space is symmetric. In this case, $100m \times 100m$. We use a simple symmetric 2-D dimension for prove of concept;

- open space,

- flat terrain, no physical obstruction to the radio path,
- nodes are randomly and uniformly distribution,
- maximum graphs are connected.

### 7.1.2 Evaluation Criteria

In order to compare the results of the experiments, we use the following quantities for,

- *degree*, we include the measurements of minimum degree, average symmetrical logical degree and physical asymmetrical degree. Whilst average degree reflects on average how many communicable neighbours a node has, physical degree reflects the actual radio coverage of a node that contributes to the interference in the neighbourhood.

  Minimum node degree can indicate the reliability of a graph $G$ as it has the relation $k(G) \leq k'(G) \leq \delta(G)$, where $k(G)$ indicates $k$ vertices in $G$ cut set, $k'(G)$ indicates $k'$ edges in $G$'s edge cut set, $\delta(G)$ is the minimum degree of the graph.

- *connectivity*, this is the percentage of connected networks after STC optimisation of all the simulations. In a single simulation, this is the percentage of nodes in the largest component;

- *average transmission power*, this is measured as the average transmission power or range used in the network. Lower transmission power is more desirable.

- *number of broadcast messages*, this is fixed in the k-Neighbours algorithm to be $2n$. However, STC is an iterative algorithm and the number of messages depends on the density of the network and parameter settings. It can be used to evaluate the effectiveness of the algorithm.

- *time to settle*, this is the number of iterative rounds for the topology to reach steady state. It is important to have a short settling time because of the need to adjust to network changes.

## 7.2 Temporal Analysis of Single Simulation

Since STC uses an iterative approach, the study of temporal behaviour and the final topology of a single simulation gives us the opportunity to observe the delay, time to stability and oscillation, if any, of the system. We can also check if the algorithm is behaving the way we expected, compare the generated topology to the ones generated by other algorithms and observe the result.

As an example, we have 100 nodes randomly located in a $100 \times 100$ space. Each node has the same radio configuration but is allowed to select their transmission power individually. We observe the dynamic through node degree, connectivity and transmission power in the network.

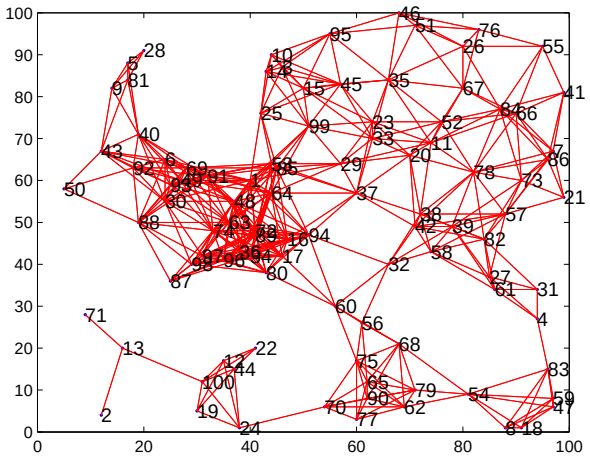



Figure 7.1: Homogenous network generated with critical transmission range

Figure 7.1 shows a homogenous network with every node transmitting at the critical transmission range for this particular network, which is the minimum digitised range that would connect the graph. We can see the uneven distribution of nodes due

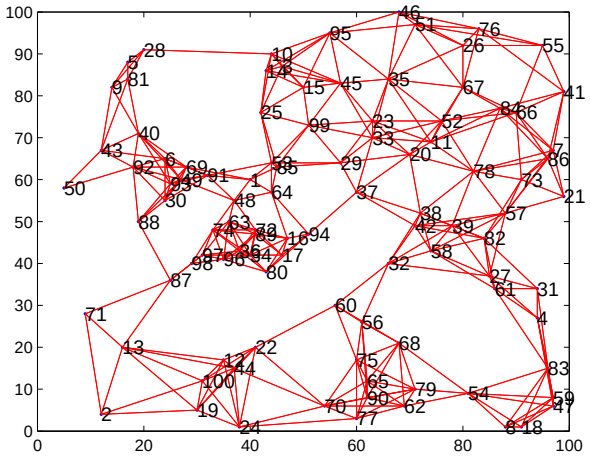


Figure 7.2: Network optimised with k-Neighbours $k = 9$ algorithm

to the random effect. The nodes in the middle (around node 1) are highly clustered with the highest node density. The density at the bottom left corner is low and thus nodes are sparsely connected. Nodes at the top right corner form a nearly regular meshed network.

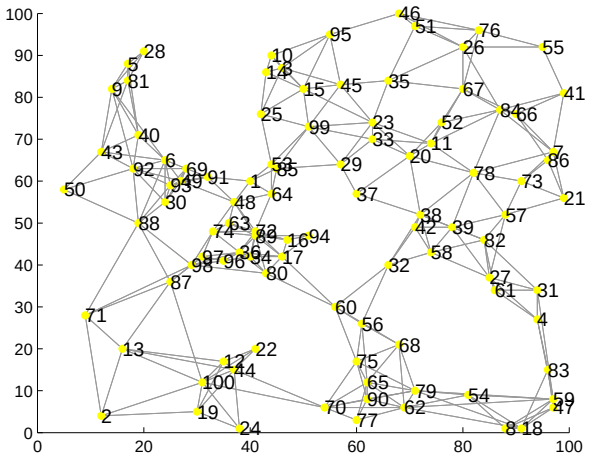


Figure 7.3: Network optimised with STC $\eta = 3$ algorithm

Figure 7.2 and 7.3 shows the optimisation results using k-Neighbours $k = 9$ and STC $k_L = 3$, which are comparative parameters in terms of connectivity as illustrated in chapter 7. Both algorithms are density based to try to identify the

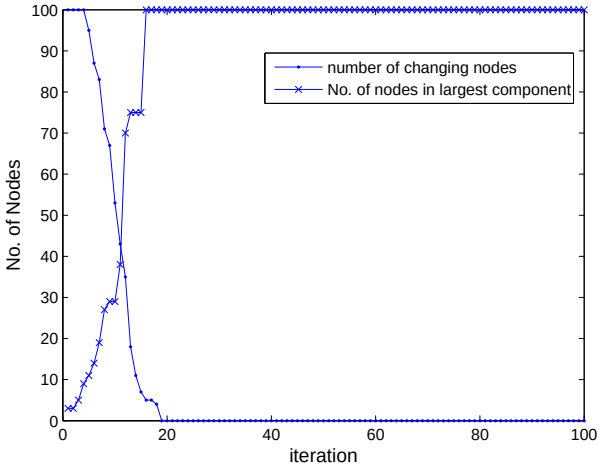
difference in density in the network. We can see that while the connectivity of the optimised graphs at the right-hand corner of the network is quite similar to the homogenous CTR graph, both graphs are less clustered around node 1 and have introduced more connections where the density of nodes is low. Both graphs have achieved connectivity via local density measurement in this case. It is noted that k-neighbours and STC do not always generate a connected network. In this particular network the graph would not be connected using kneigh $k = 6$.

When analysing the time series of STC (Figure 7.4) of the graph, we can see that the system reaches stability after 19 iterations. However, in Figure 7.4(a), we observed that the network is connected after 16 iterations. The delay in achieving stability can be explained by the lack of global knowledge, and the extra reliability that is put into the network given that the final graph has a minimum degree of 2. $k_H$, set to 20 in this experiment has no effect on the simulation since the final network has $\eta_4 < 20$ for all nodes.

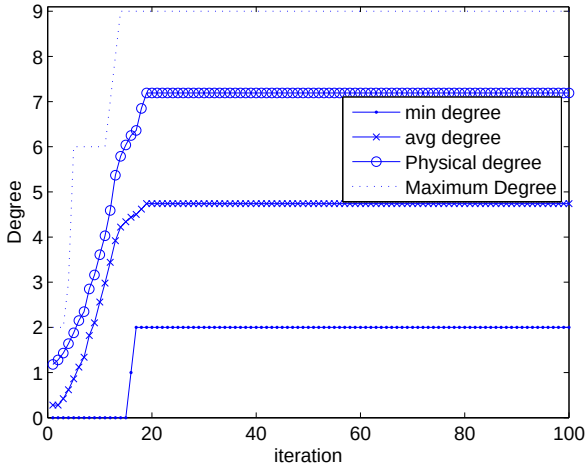
In Figure 7.4(c), the number of increased power request broadcast packets (type 2) is maximum at the beginning when all nodes try to increase their connectivity and decreases with time as more and more nodes have their connectivity requirements satisfied. The number of update broadcast type 1 indicates the node activity in the network. There are most changes in the network at around 12-13 cycles and the network quickly settles after that.

The time it takes a simulation to settle is highly dependent on the network capacity, or the average distance between nodes. Since the nodes only increase 1 power step at a time, in a more dense network a node is more likely to achieve the required connectivity with less incremental steps.

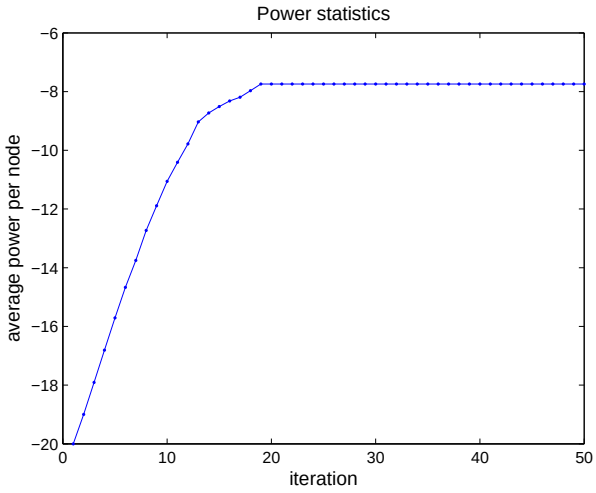
Figure 7.5 shows the metric $\eta_4$ distribution in the final optimised network. While the majority of the nodes have $\eta_4$ value much higher than $k_L$, this is inevitable in a Unit Disk Graph (UDG) type of network as a graph would have an intrinsic $\eta_4$



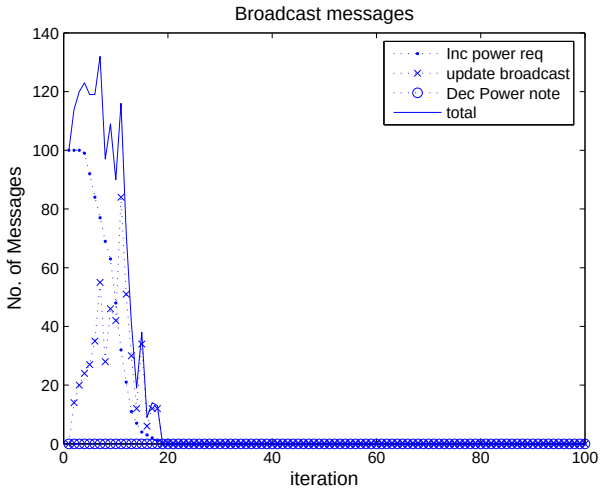

(a) node dynamic and connectivity




(b) degree statistics




(c) average transmission power




(d) number of broadcast messages

Figure 7.4: Temporal analysis for a single simulation using STC $k_L = 1$. The time to settle in this simulation is 19 cycles.

value. We can however obtain an idea of how to set $k_H$ for further optimisation.

## 7.3 Parameter and Density

We discuss in detail the effect of the parameter settings in the algorithm and the node density on the resulting networks in the following sections.

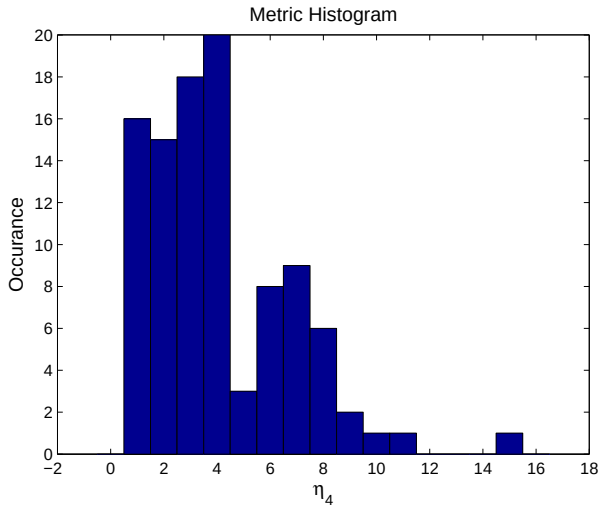


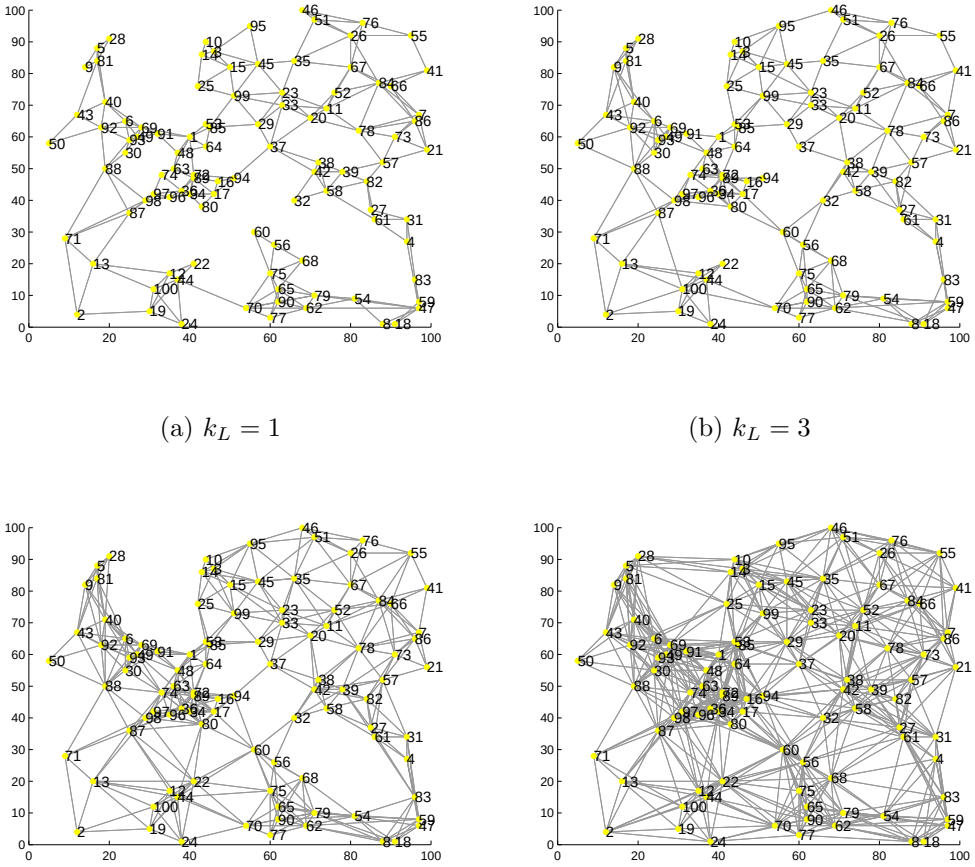

Figure 7.5: The final distribution of $\eta_4$ in the network, $k_L = 1$

### 7.3.1 $k_L$ - the lower threshold

Similar to the parameter $k$ in k-Neighbours where $k$ controls the link density of the network, $k_L$ in STC controls the link redundancy and hence, robustness of the network. A larger $k_L$ will give a higher probability of connectivity in a given topology. However, this comes at a higher cost of transmission power and node degree. Therefore, the optimal $k_L$ should be set to result in a high level of connectivity ($> 98\%$) and small node degree.

We use the normal setting in this set of experiments where 100 nodes are placed in $[100, 100]$ space. $k_H$ is set to a very high value such that nodes do not decrease their power due to too much redundancy.

Figure 7.6 shows the topology generated by different values of $k_L$. It is observed that link density increases with $k_L$. In Figure 7.6 (a) and (b) some areas appears to be the same. This is because the intrinsic $\eta_4$ to satisfy $k_L = 1$ of these parts of the network is already above $k_L = 3$ as observed in Figure 7.5. The networks generated with $k_L = 10$ and $k_L = 50$ are a lot more dense. It is also observed that comparing the results of $k_L = 1$ and $k_H = 3$, links are added to the more sparse and asymmetric



(a) $k_L = 1$ (b) $k_L = 3$

(c) $k_L = 10$ (d) $k_L = 50$

Figure 7.6: Network Topology generated with different $k_L$. $N = 100$, $k_H = 10,000$

part of the network.

In Figure 7.7 we show the connectivity results of STC with varying $k_L$. In general connectivity improves with higher $k_L$. It is more noticeable in a higher density network. The basic STC algorithm cannot guarantee connectivity and the networks connect with high probability only. The probability is calculated to exceed 0.9 for all cases with $k_L > 2$ and exceed 0.99 for $k_L > 4$. Since the probability for connectivity is high at large $k_L$, the difference for varying density is not obvious for the number of simulations we carried out.

In Figure 7.8, we observed that node degree, minimum degree and physical degree

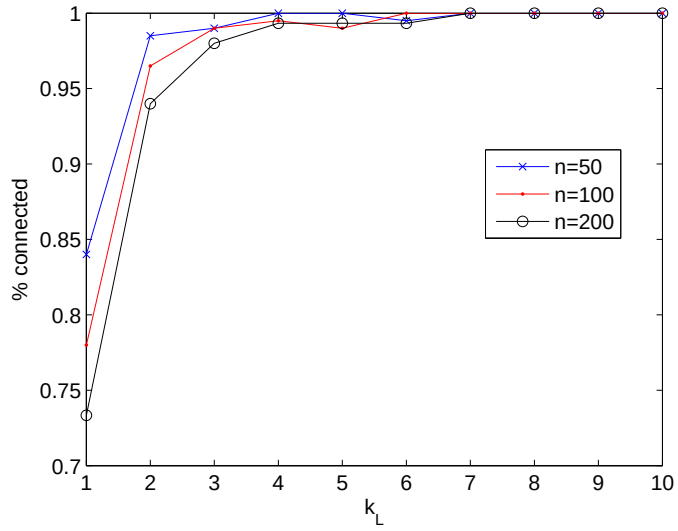



Figure 7.7: connectivity varies with $k_L$ for $n = 50$, $n = 100$ and $n = 200$. The simulations were run for 200 times to obtain the result in each case.

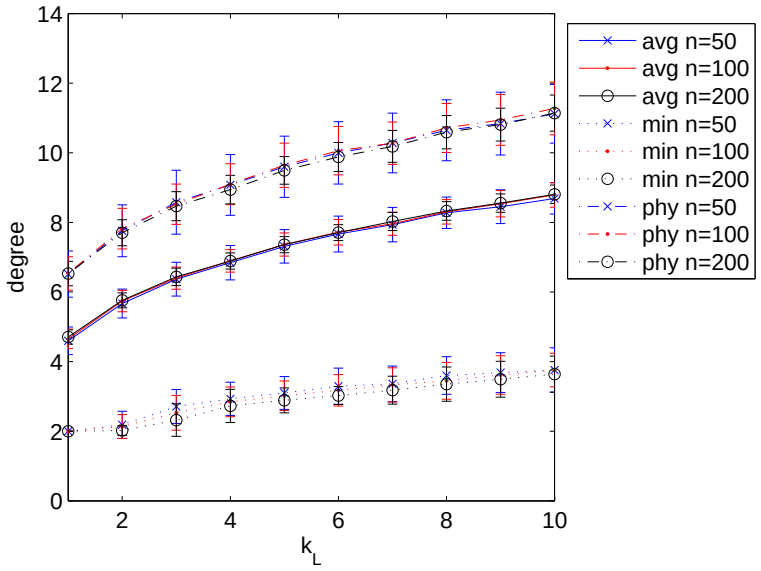



Figure 7.8: Average node degree, minimum degree and physical degree varies with $k_L$ for $n = 50$, $n = 100$ and $n = 200$. The simulations were run for 200 times to obtain the result in each case.

also increases with higher $k_L$, although the increase slows down as $k_L$ becomes progressively higher. It means that by increasing $k_L$ we can increase the link density in the network and enhance the reliability by the improvement in the minimum degree.

One interesting phenomenon is that the degree characteristics for $n = 50$, $n = 100$ and $n = 200$ are nearly the same. This means that we can control the degree characteristics in the network using $k_L$ in a manner similar to the k-Neighbours protocol. However,with STC we do not see the same level of connectivity degradation with increasing node density as in k-Neighbours.

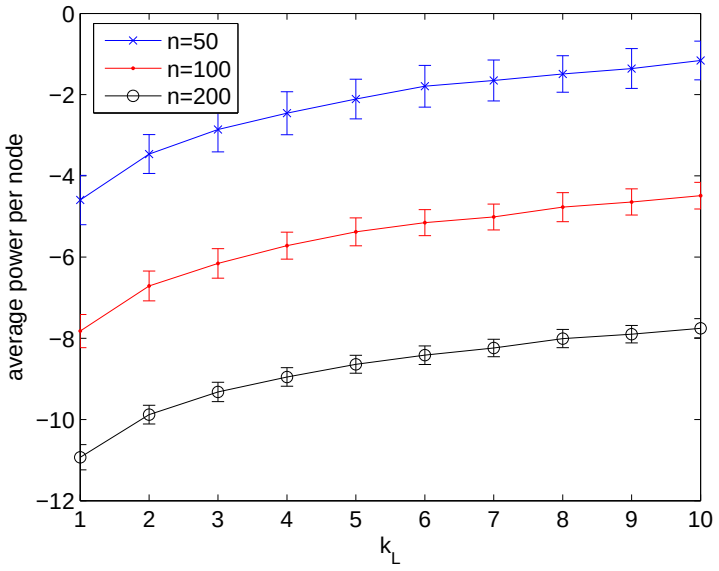



Figure 7.9: Average node degree, minimum degree and physical degree varies with $k_L$ for $n = 50$, $n = 100$ and $n = 200$. The simulations were run for 200 times to obtain the result in each case.

The results relating to average power usage per node (shown later in Figure 7.20(c)) are consistent with the degree characteristics. The average power required increases with $k_L$ to achieve the increase in link density. Again, the increase slows down with high $k_L$. The power requirement is less in a dense network because the average distance between nodes is less.

Figure 7.10 shows the average $\eta_4$ value in the final graph generated from STC. The resulting $\eta_4$ is always above the required $k_L$ as the incremental steps of $\eta_4$ of a graph is not continuous, but is related to how the nodes are distributed. At low $k_L$ the resulting $\eta_4$ are nearly the same for all node density. At high $k_L$, $\eta_4$ is higher for

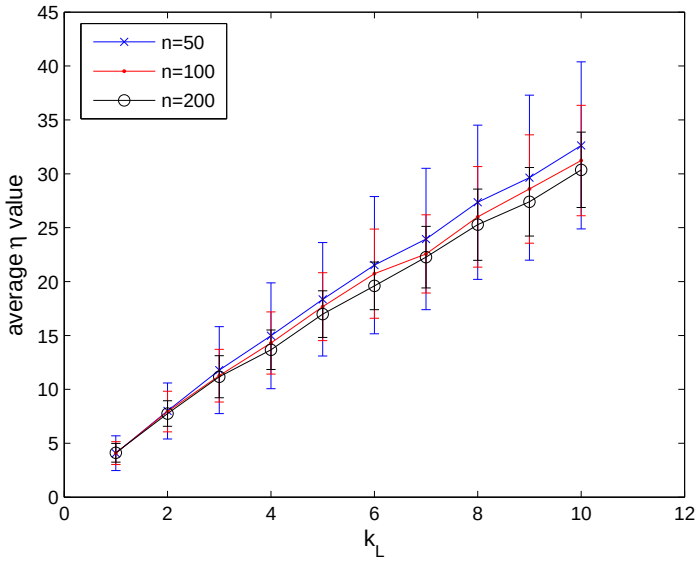



Figure 7.10: Average $\eta_4$ in the final graph varies with $k_L$ for $n = 50$, $n = 100$ and $n = 200$. Each point shows the mean and standard deviation of 200 simulations.

lower node density and the different becomes more obviously as $k_L$ increases.

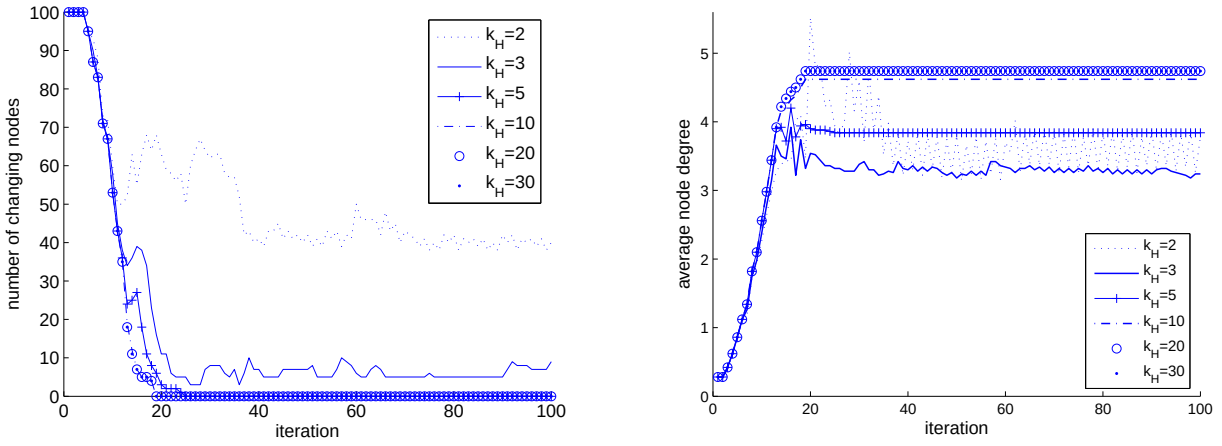
### 7.3.2 $k_H$ - the higher threshold and stability of the network

The parameter $k_H$ acts as an upper boundary to prevent a node from being too clustered. It is necessary because STC uses a second-order metric $\eta_4$, a parameter update would take at most 2 hops, or 2 updating cycles, to reach all the nodes involved. A node may have issued an increased power request too quickly before it gets the message that its neighbour has already done so. Another scenario is when new nodes are introduced to the network, a node needs to know if the local area has become too clustered.

The parameter $k_H$ affects the performance of STC in two ways. It is possible for $k_H$ to help further in optimising the node degree and power setting in the network such that the normalised node density is more evenly distributed. However, if the parameter is set too low it is possible for the network to oscillate between states and

not settle [1].

We take an example network in Figure 7.6 with $k_L = 1$ as an example to illustrate the effect of $k_H$. We have noticed that the network may go into oscillation for any $k_H < 20$ in this case. The probability of oscillation is higher for lower $k_H$ value. If oscillation occurs, it is possible that the resulting network would not be connected.




(a) Number of changing nodes

(b) average node degree

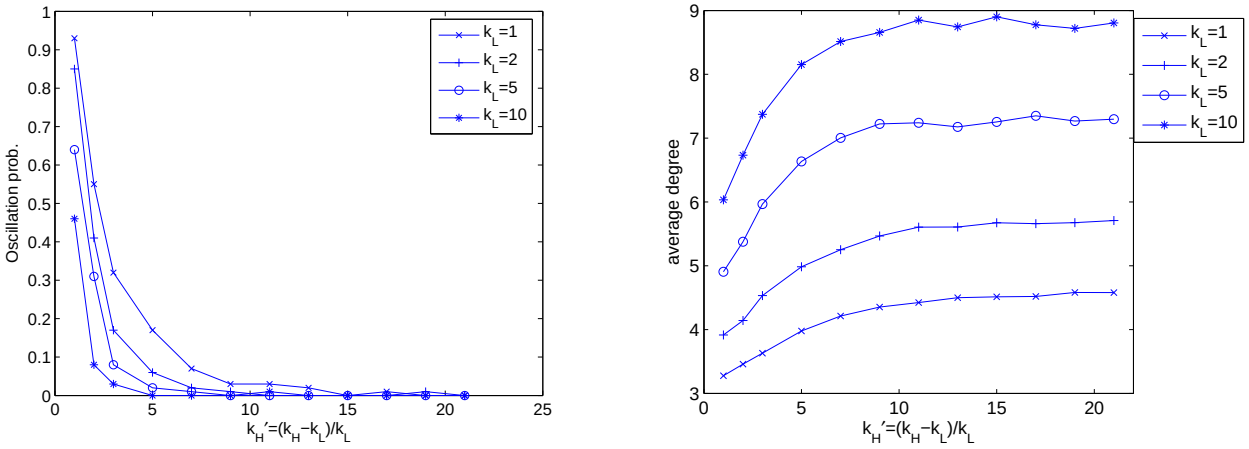
Figure 7.11: Effect of $k_H$ on node degree and number of changing nodes over time. Network dimension=$[100, 100]$, $n = 100$

Figure 7.11 shows the number of changing nodes and average node degree over time with different $k_H$ values for the same sample network. We can see that oscillation occurs with $k_H = 2$ and $k_H = 3$, but not with $k_H > 5$ in this case. In this case, $k_H = 5$ does not result in network oscillation, however, the resulting degree statistics is much lower (3.8 instead of 4.7) than the network without the $k_H$ restriction. $k_H = 10$ also gives a little optimisation on the node degree (4.6 instead of 4.7). There is no optimisation for $k_H = 20$ and $k_H = 30$ and they behave the same way as network with no $k_H$ boundary.

We observed that time to settle for all node density is within 40 iterations as

[1]Oscillation happens in a localised area where the nodes would alter their transmission powers between some values even if there is no state change in the network

shown later in Figure 7.22. If there are still nodes changing their transmission power after 100 iterations in a single simulation, we define this parameter set as oscillating.




(a) Probability of oscillation

(b) average node degree

Figure 7.12: Effect of $k_H$ on oscillating probability and node degree over 100 simulations. Network dimension=$[100, 100]$, $n = 100$

Figure 7.12 shows the effect of $k_H$ on oscillating probability and the resulting node degree of the simulations. To compare the result with different $k_L$ on the same graph, we have created a new metric $k'_H = (k_H - k_L)/k_L$ to align the curves. We can see that the oscillating probability decays very quickly with increasing $k_H$, especially with high $k_L$. With $k_L = 10$ it decreases to $< 0.01$ after $k'_H = 5$. On the other hand, the node degree increases in a much slower rate. The node degree only returns to the level without the $k_H$ restriction with $k'_H > 10$ with $k_L = 10$. Hence, we can an improvement on node degree for $5 < k'_H < 10$ for $k_L = 10$ with very low probability of oscillation. The advantage for small $k_L$ is less obvious as can be seen in the graph.

### 7.3.3 Network node density

In this section, we investigate the performance of STC against network density and test the robustness of the protocol. We investigate node density in the range of 10

to 200 in a $[100 \times 100]$ space to cover the more common density range of sensor networks.

Broadly speaking, STC and k-Neighbours have the same characteristics in that the probability of connectivity decreases with higher network node density. Experimenting with the node density we can observe other interesting characteristics with STC.

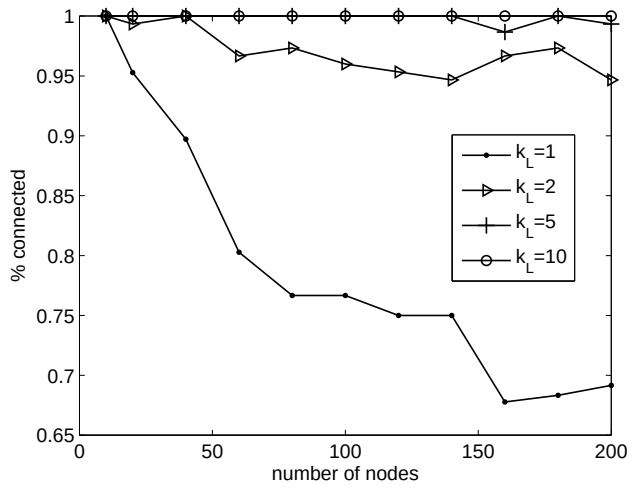
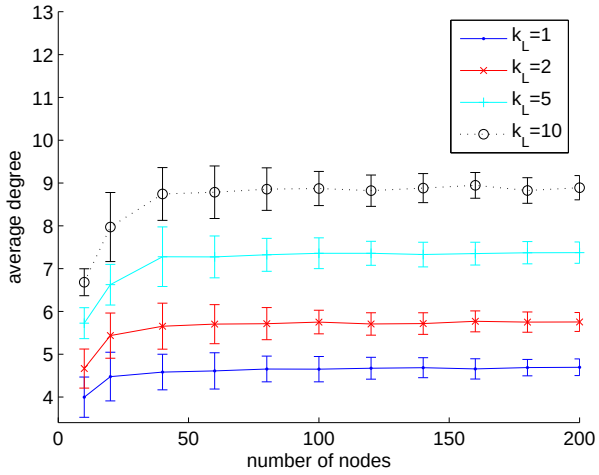


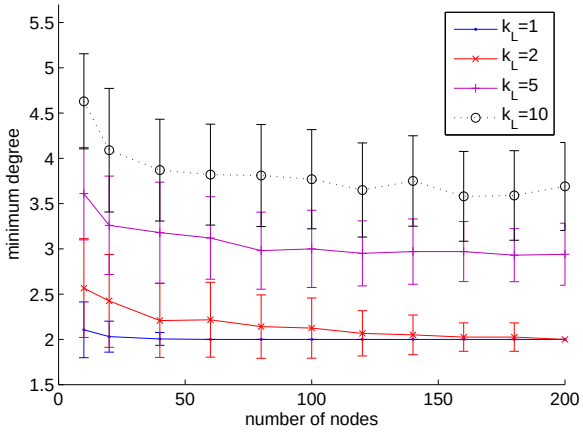

Figure 7.13: Connectivity vs node density for $k_L = [1, 2, 5, 10]$ over 150 simulations

Figure 7.13 shows the connectivity of STC with varying number of nodes. We can see that at $k_L = 1$, the connectivity is very sensitive to the node density and drops to below 0.7 for $n > 150$ or $d > 0.015$. The connectivity improves with higher $k_L$ and all the networks in the experiment are connected for $k_L > 10$. STC does not guarantee connectivity, thus, we expect the probability to drop at a higher node density even for $k_L = 10$.

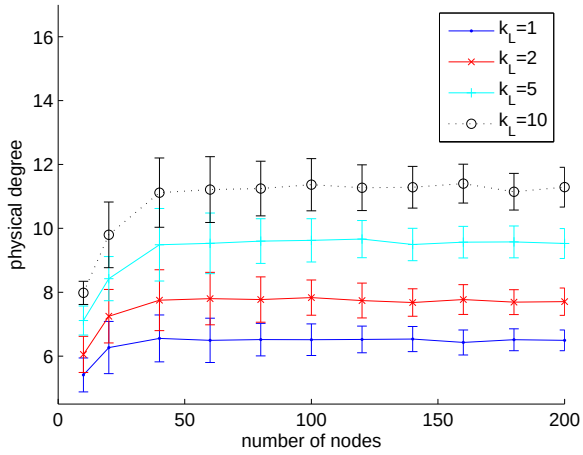
Figure 7.14 shows the degree statistics with varying node density. We can see that the average degree, minimum degree and physical degrees become quite stable with increasing $n$ for $n > 50$. The initial variation for $n < 50$, or $d < 0.005$, is due to the *edge effect* of the graph since the majority of the nodes are *edge nodes* and are connected to less neighbours than the middle nodes. This result is consistent with




(a) average degree




(b) minimum degree




(c) physical degree

Figure 7.14: Degree statistics vs node density for $k_L = [1, 2, 5, 10]$ over 150 simulations

Figure 7.8 in that the degree of the final graph is insensitive to different $n$. The other observation is that the minimum degree of the graphs decreases slightly with higher node density. The possible explanation for this could be the decrease in connectivity, thus reliability reflected by the minimum degree.

In Figure 7.15, the average transmission power used per node and the time to

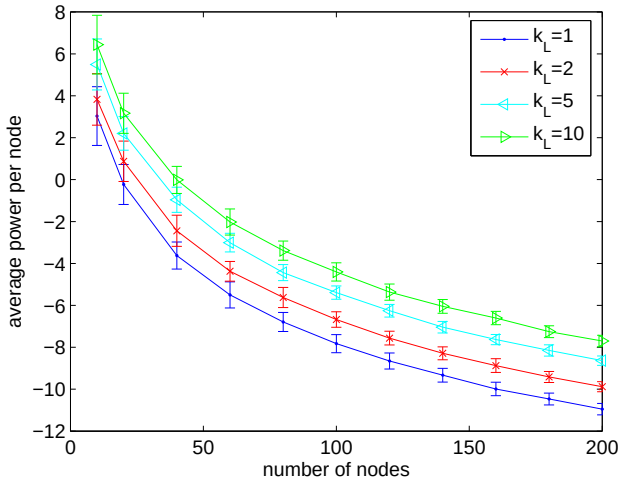



Figure 7.15: Average transmission power used per node vs node density for $k_L = [1, 2, 5, 10]$ over 150 simulations for each points

settle of the algorithm in Figure 7.22 have similar behaviour. As expected, they both decrease with increasing node density as the average distance between nodes decreases. The decrease slows down at the higher end of node density. Link density increases with higher $k_L$ and therefore the power required for transmission is higher for larger $k_L$. We can conclude that $\eta$ reflects the density of the network and that different values of $k_L$ also represent the different node density requirements.

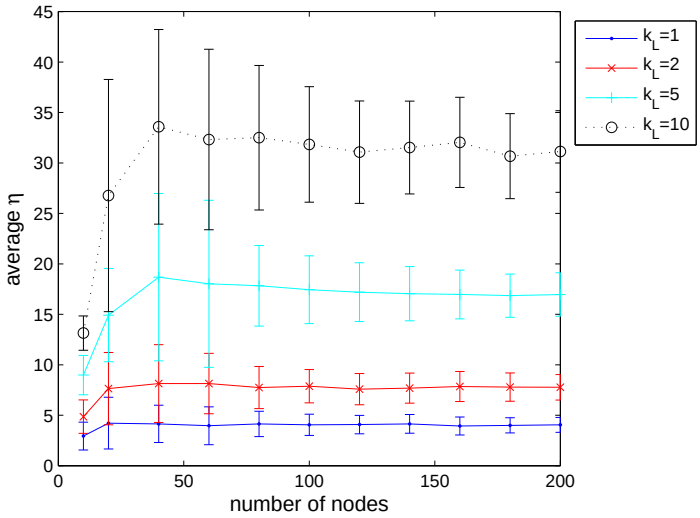



Figure 7.16: Resulting $\eta$ vs node density for $k_L = [1, 2, 5, 10]$ over 150 simulations

Figure 7.16 shows the final average $\eta$ in the network. Note again that each network has their intrinsic $\eta$ value to satisfy the required $k_L$, which is density dependent and therefore the resulting $\eta$ is always higher than the required $k_L$. $\eta$ decreases slightly with increasing number of nodes and this declining trend is more obvious with high $k_L$, which was also observed in Figure 7.10. This is again due to the edge effect of the networks. The initial increase of $\eta$ and the tuning is consistent with the result in Figure 7.14 and is contributed by the edge effect. This experiment suggests that the final $\eta$ is independent of node density. Thus, STC is scalable.

### 7.3.4 Scalability

We investigated the scalability of STC by increasing number of nodes and keeping the network density the same. $k_L$ is set to 2 for a general comparison. The results are shown in Figure 7.17 and Figure 7.18. We can see that the average degree of the network remains the same ($k$ 5.7) for $n > 50$ (At $n < 50$ the result is dominated by boundary effect). This implies $\eta$ has a direct relationship with node degree and is not affected by the network size. The connectivity decreases slightly with the increased number of nodes (or network size). This is because the probability of island forming increases with more nodes in the network. However, the connectivity remains above 0.95 in the region of $50 < n < 200$.

## 7.4 Comparisons to k-Neighbours Algorithms

In this section, STC is compared to the k-Neighbours protocol because both algorithms require similar information, *i.e.* Node ID, to operate. The control parameters $k$ and $\eta_4$ are both density-based metrics (see Section 5.4.1) and both algorithms aim to generate a topology that is connected with high probability. They are both simple, fully-distributed and are examples of physical TC algorithms.

The major difference between the two algorithm is that $k$ in k-Neighbours is a

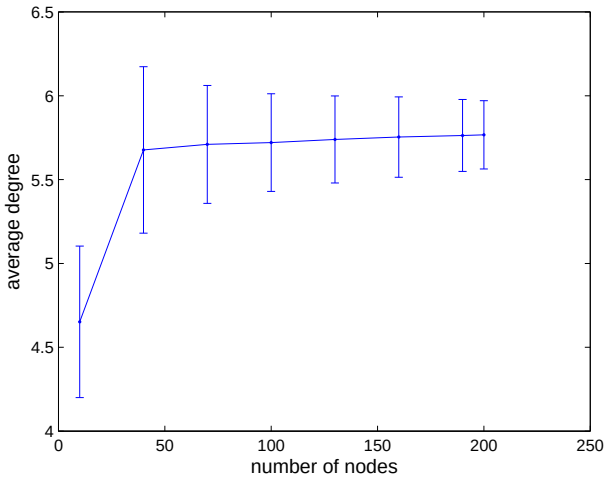



Figure 7.17: Average degree vs number of nodes. Network density = 0.01 $m^{-2}$, $\eta = 2$. 1000 simulations are performed for each point

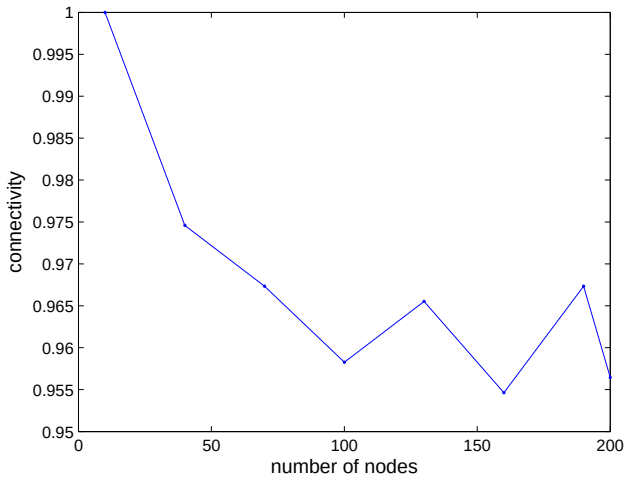


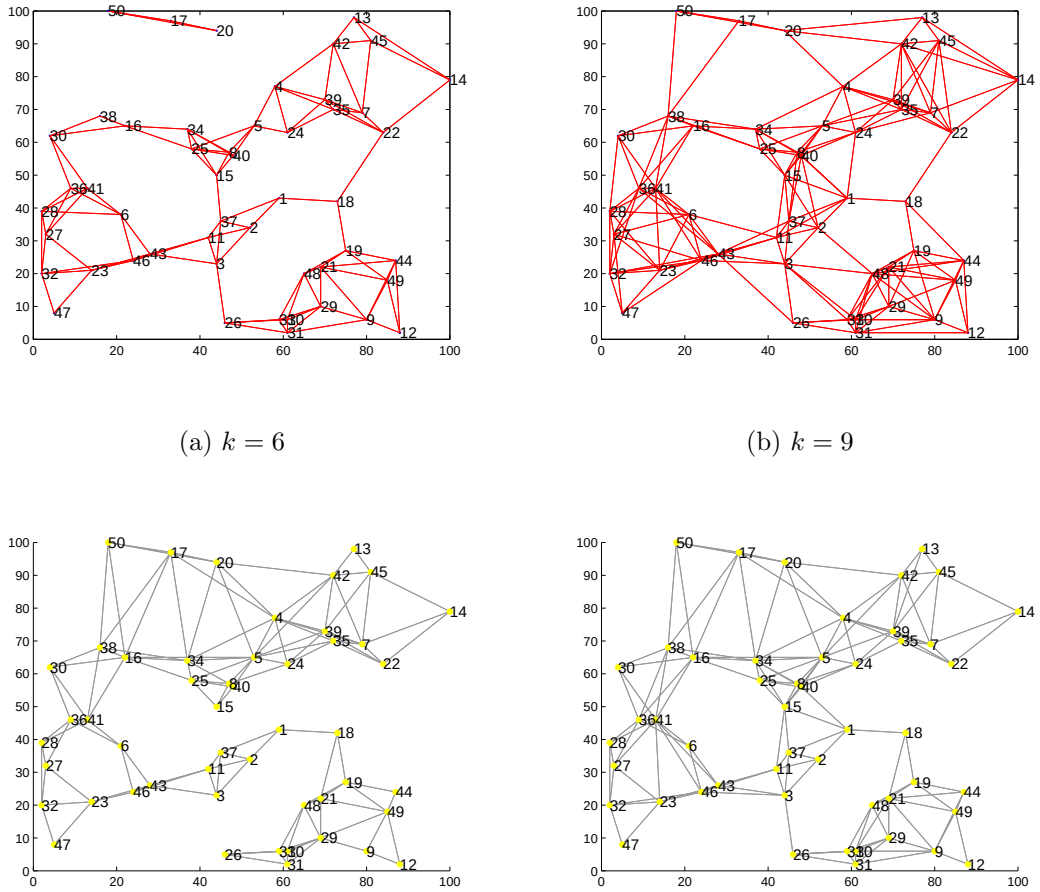

Figure 7.18: Percentage of connected network vs number of nodes. Network density = 0.01 $m^{-2}$, $\eta = 2$. 1000 simulations are performed for each point

zero order metric, while $\eta_4$ in STC uses a second-order metric. Although the extra requirement for distance information means that it takes longer for the algorithm to stabilise because of its iterative nature, we expect the second order metric would give us an improvement in connectivity over the k-Neighbours algorithm.

We use STC $k_L = 2$ to compare with k-Neighbours $k = 9$, which is proposed in [BLRS06] for 95% connected topology, to fulfil a higher requirement of connectivity.

A lower requirement STC $k_L = 1$ is compared with k-neighbours $k = 6$, proposed in [BLRS06] such that 95% of the nodes are connected in the largest component for lower connectivity requirement.

In addition, the minimum transmission range (*critical transmission range(CTR)*) required for connectivity of each network is also simulated for comparison. This helps us to understand the behaviour of the theoretical minimum of the networks.



(a) $k = 6$ (b) $k = 9$

(c) $k_L = 1$ (d) $k_L = 2$

Figure 7.19: Topology generated by k-Neighbours and STC $n = 50$

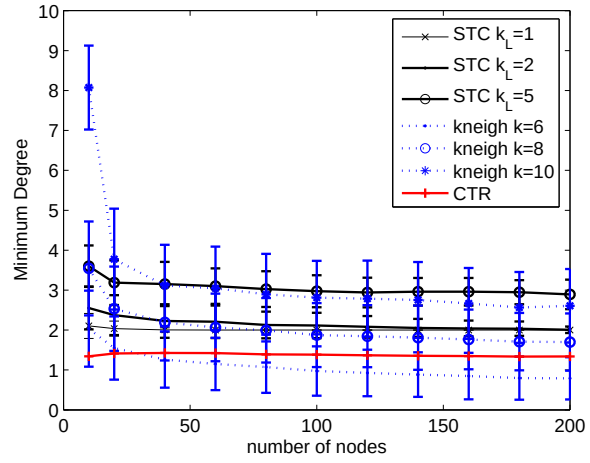
Figure 7.19 shows the topology generated by k-Neighbours and STC in a network of 50 nodes with the same placement. Among the four topology, k-Neighbours $k = 6$

generated a topology with the lowest degree and there is an isolated island in the least dense area and this particular network is not connected. k-Neighbours $k = 9$ generates a connected network with a higher average node degree. The networks generated by STC were both connected in this case and the graph with $k_L = 1$ has lower degree than $k_L = 2$. The topology generated by STC has different shapes than k-Neighbours because of the difference in local connectivity evaluation.

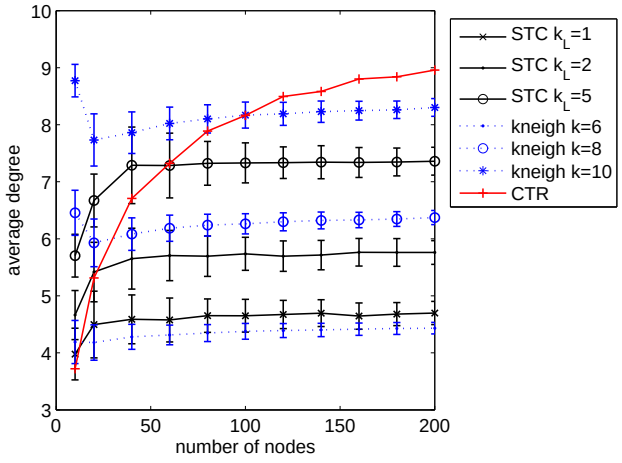
Figure 7.20 shows the simulation results of the different protocols with varying network densities. We observe that in Figure 7.20(a) STC maintains a similar level of minimum degree throughout all densities whilst the minimum degree of a k-Neighbours graph decreases with density. This suggests that the STC topology has more resilience to single point or link failures. Statistically the minimum degree is the lowest for the CTR graphs, which are obtained from setting the lowest transmission range to generate a connected network. The transmission range is the same among all the nodes, while the node density varies at different part of the network.

In Figure 7.20(b), the density-based TC k-Neighbours and STC maintain similar level of average degree with the increase in density, with CTR having an increasing node degree. The result of CTR agrees with the theoretical results that the degree should be a function of $\log n$. k-Neighbours maintains the level of average degree at around $k - 2$. STC also maintains the average node degree at a lower value than k-Neighbours for similar connectivity. It is observed that k-Neighbours has a higher node degree than STC at very low node density (number of nodes $< 10$) and the phase change at $10 < n < 50$ is due to the boundary effect as discussed before. During the phase change when boundary condition is dominant, k-Neighbours still requires the node to achieve the same node degree, which results in higher node degree than necessary for connectivity as compared to STC and CTR.

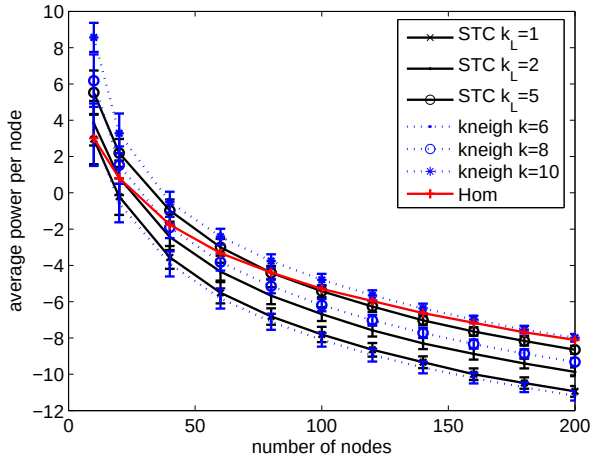
Figure 7.20(c) confirms the result of Figure 7.20(b). In general for all control algorithms the average power used per node decreases with node density. The start-



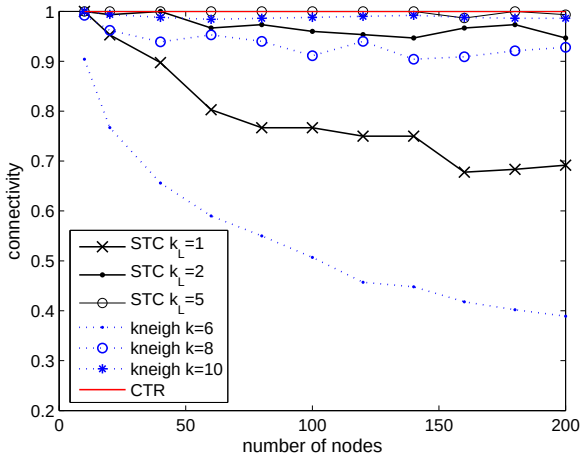

(a) Minimum Degree




(b) Average Degree




(c) Average tran. Power




(d) Fraction of network connected

Figure 7.20: Simulation results. The simulations were repeated 200 times STC and 1000 times for k-Neighbours and CTR. The results of the different protocols were compared using average and minimum node degree, transmission power and connectivity.

ing power consumption of CTR is the lowest but the decrease is much slower when density increases. The decreasing rate for k-Neighbours and STC is about the same.

Comparing the result of k-Neighbours and STC in Figure 7.20, we can see that STC generates graphs with a higher minimum degree. We observe that STC $k_L = 2$

produces a graph with a lower node degree than k-Neighbours $k = 8$ but with better connectivity (STC> 0.95, k-Neighbours> 0.91). The same is observed for STC $k_L = 5$ and k-Neighbours $k = 10$. It is harder to observe the performance of STC and k-Neighbours with lower parameter setting as the lowest setting for $k_L$ in STC is 1. However, we see from the result of Figure 7.13 that in order to achieve connectivity> 95%, setting $k_L > 2$ is the minimum requirement.

## 7.5 Complexity of STC

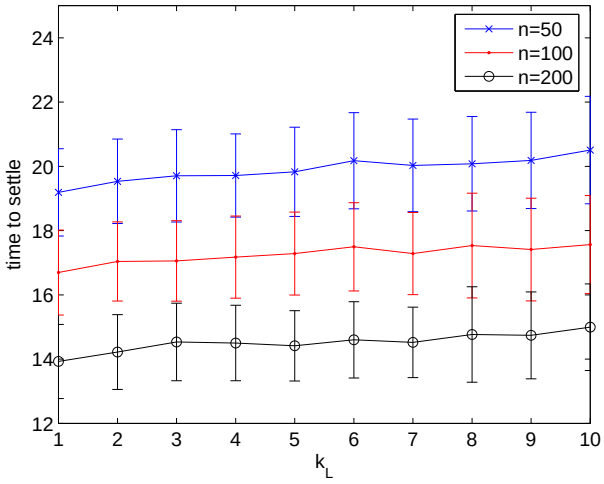



Figure 7.21: Time to settle varies with $k_L$ for $n = 50$, $n = 100$ and $n = 200$ for the same size of space. The simulations were run for 200 times to obtain the result in each case. The time to settle parameter in 7.21 is not much affected by the $k_L$ parameter but the node density.

STC is an iterative algorithm based on nodes exchanging their neighbour tables with 1 hop neighbours. Only node IDs are required for the implementation of STC. As nodes adjust their transmission power based on $\eta$, their neighbours are also changing, which would trigger re-calculation of $\eta$ in the node. In the simulations in this chapter the amount of time required for the network to stabilise, meaning that no more nodes in the network changed their transmission power given a static condition, is within 10-30 iterations dependent on mainly the network density as illustrated on

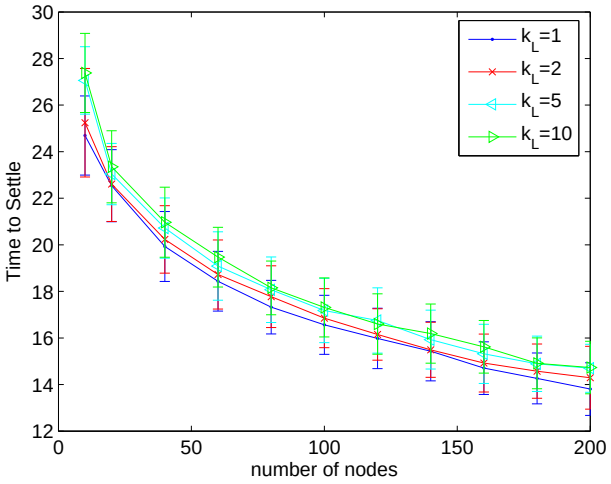



Figure 7.22: Time to settle used per node vs node density for $k_L = [1, 2, 5, 10]$ over 150 simulations

Figure 7.22. However, it varies only slightly with different $k_L$ as shown in Figure 7.21. High density networks require less time to achieve stability because nodes are closer together in a high density network. With STC, each node optimises from the lowest power level, therefore, in a high density network it would take less number of iterations to reach network stability.

Capacity and network contention is a major concern in the SECOAS project, where CSMA is used for multiple media access. Although k-Neighbours states that the initialisation and updates should only take up 2 cycles, one cycle for each node to transmit its own id and one for transmit its neighbour list, measures such as random delay have to be introduced to prevent all the nodes transmitting with maximum power at the same time. Therefore, the length of one cycle should increase with the network density to fulfil the requirement of SECOAS. This is different from STC where the settling time decreases with node density.

The initialisation period of STC is longer than k-Neighbours because of its iterative nature, however, network updating in STC topology only involves 1 and 2 hop immediate neighbours that are affected by the change, whilst updating with k-

Neighbours involves all the nodes transmitting with maximum power and rebuilding the neighbour tables. With STC, nodes send out updating requests at random to detect any changes in the network. The probability for sending out requests depends on the rate of change in the network. Moreover, $\eta$ is bounded below and above such that any remedial actions, such as a power increase to cover a failing node, can be reversed at a later time when a new node is introduced.

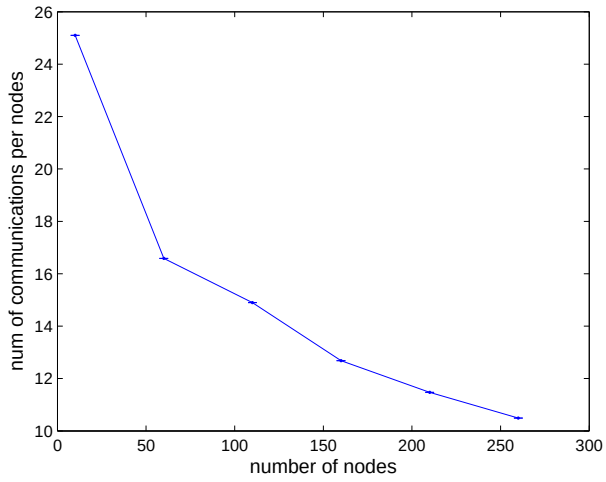



Figure 7.23: Average number of messages exchanged per sensor nodes. Size of the network is kept the same.

The number of messages exchanged with k-Neighbours is exactly $2n$ for each update. In the case of STC, this is a function of network density as shown in Figure 7.23. The number of messages each node needs to exchange decreases with increasing node density, as the number of iterations required to reach stability decreases.

## 7.6 Chapter Summary

In this chapter, we investigated the performance of Subgraph Topology Control (STC) in terms of average degree, connectivity, transmission power and complexity. The results of STC are compared to k-neighbours, a competitive algorithm which has the same hardware requirement as STC, and critical transmission range,

which has provided a theoretical benchmark for comparison.

In our experiments, we found that by setting $k_L = 3$, the networks are connected with probability greater than 0.97 for $50 < n < 200$. This gives us an average node degree of 6.1. Setting a larger $k_L$ value can improve connectivity, however, would also increase the network power consumption and node degree. In fact, setting $k_L \geq 7$ results in 100% connectivity in all our simulations for $50 < n < 200$, and the resulting node degree is around 8. From equation 6.1.1 we can see that $\eta_4$ is a degree 3 polynomial of the transmission range, therefore, node degree rises very slowly with the increase of range.

We investigated the scalability of STC by keeping the node density constant and increase the number of nodes in the network. It was found that the resulting node degree remains constant beyond the network size dominated by the boundary condition. The connectivity only decreased very slightly to 0.95 at a network size of 200 nodes.

When compare to the performance of k-neighbours algorithm, we found that STC has higher probability of connectivity with similar resulting degree and using similar transmission power in 2-dimensional networks. STC also produces higher minimum degree, which means that the resulting network is more resilience.

STC is iterative in nature. For a network density of $0.01m^{-2}$, STC takes less than 18 iterations to settle for this particular type of radio. It can take up to 30 iterations for node density less than $0.001m^{-2}$. However, from Figure 7.4(a) we can see that the majority of the nodes settle their power a lot more quickly than this depending on their local density.

# Chapter 8

# Adaptability of Subgraph Topology Control

In the last chapter, we have proven STC has an improvement performance in terms of connectivity and node degree over the k-Neighbours algorithm in a static environment. However, the SECOAS environment is dynamic and changes over time due to the following factors.

- *Node position drift.* SECOAS nodes are not designed to be mobile. However, they drift from their original positions with waves and tides in the short term, and with the movement of the sandbank in the longer term.

- *Node status change.* Nodes batteries can run down or the nodes may be damaged or malfunction from time to time. On the other hand, new nodes can be added by the user for measuring new points of interest, replacement of faulty nodes or to improve the existing communication structure. These changes need to be observed by the neighbouring nodes and a new topology is formed.

- *Link dynamics.* The wireless connections between nodes change with time. The arrival signal strength varies due to fast and slow fading in the environment and moving obstacles. A connection can disappear momentarily if it is blocked by passing objects such as boats, buoys or animals.

For waves and tides movement, the nodes are likely to move slightly to and fro from their original position. For sandbank movement, nodes are likely to drift in the same direction relative to each other. However, we consider the position drift to be relatively small (order of cm) compared to the separation between nodes (50-100m) because the sensor nodes are attached to a 10kg weight to prevent them from being washed away and we will not include node position drift in our simulations.

In this Chapter, we evaluate the adaptivity of STC with addition, removal of nodes. We also simulate the situation when the nodes are added to the network one by one and observe the performance of STC.

## 8.1 Removal of Sensor Nodes

In this section, we consider the effect of the addition and removal of sensor nodes upon the performance of STC. We measure the time for the network to settle after an event (addition or removal of nodes) happens. We also measure the number of nodes involved in the re-configuration, and the amount of messages passed between the nodes.

When a node $u$ disappears from the network, the event can be noted by the neighbouring nodes only if they have already been receiving messages from the node. This is achieved by nodes regularly broadcast their beacon/ heartbeat messages. These messages contain the essential information to configure a network, including its Tx power, hop count from the gateway and synchronisation information. If a neighbouring node $v$ does not hear a beacon from $u$ for $l$ number of times, it declares $u$ no longer exists and inform its neighbours of the change. The number $l$ depends on the reliability of a beacon being received. A collision based MAC scheme would have higher $l$ value as a node would not receive acknowledgements for the beacon it transmits.

When node $v$ discovers that neighbour $u$ has disappeared from its neighbourhood,

it updates its subgraph metric and broadcasts the new neighbours list. Subsequently this would trigger a series of power changes and update broadcasts from this neighbourhood.

Figure 8.2 shows an example of the self-healing property of STC. $k_L$ is set to 5 and $k_H$ is set to 50. Sub-figure 8.1 shows the topology after the network is optimised and settled. In Figure 8.2(a), node 41 fails at time instance 70 and it neighbouring nodes (5, 13, 21,28, 37, 42, 47, 50) immediately detect that its metric is under the threshold. They start the power increase procedure to reinforce the network structure and their neighbouring nodes which have low metric subsequently follow. In this scenario, the set of nodes that issue update broadcast messages are (1,5,7,9,13,14,15,16,17,21,28,34,37,38,42,47,50), which are all within 3 hops neighbours of node 41 before its failure. The failing of node 41 affects a large number of nodes in this case as a middle node and it has a major effect on the whole structure. The failing of an edge node has less effect on its neighbourhood in general.

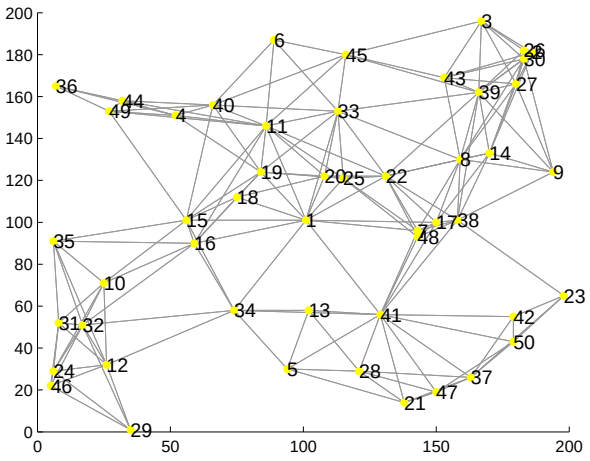
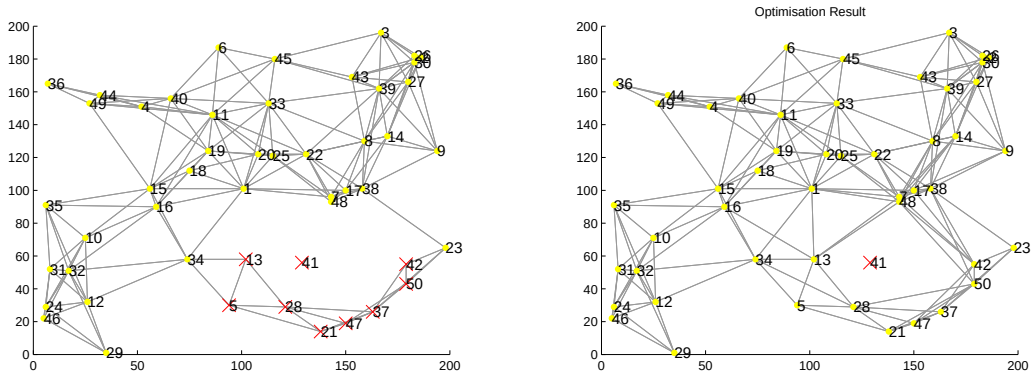



Figure 8.1: Network topology after STC optimisation

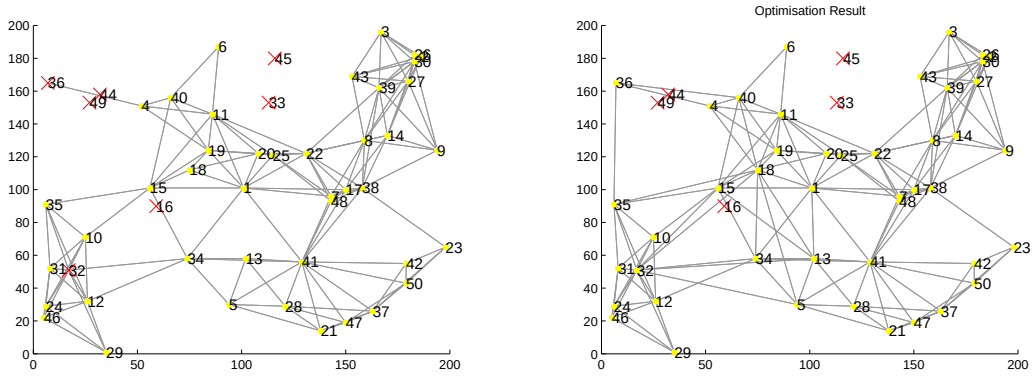
In another example we failed five of the nodes at random $[16, 33, 44, 45, 49]$ within the same network as in figure 8.1. The result is shown in figure 8.3(a) and 8.3(b). A total of 22 nodes are involved in the updating procedure this time. The re-




(a) Node 41 fails

(b) Self-healing in the neighbourhood

Figure 8.2: Simulation results for removal of a node in the network.




(a) Node 16, 33, 44, 45, 49 fails

(b) Self-healing

Figure 8.3: Simulation results for removal of 5 nodes in the network.

structuring mainly occurs near node 36. The assumed failure of node 33 and 45 does not have much effect on the neighbourhood as all the neighbouring nodes have their parameters satisfied, even with the missing connections.

We repeated the simulation 200 times for different networks and randomly failed 1 node in each case. The results are summarise in Figure 8.4(a), 8.4(b), 8.4(c). In most of our simulations, the nodes involve in the update are just those within the direct neighbourhood - hence, the low number of nodes (between 4 and 16) sending update

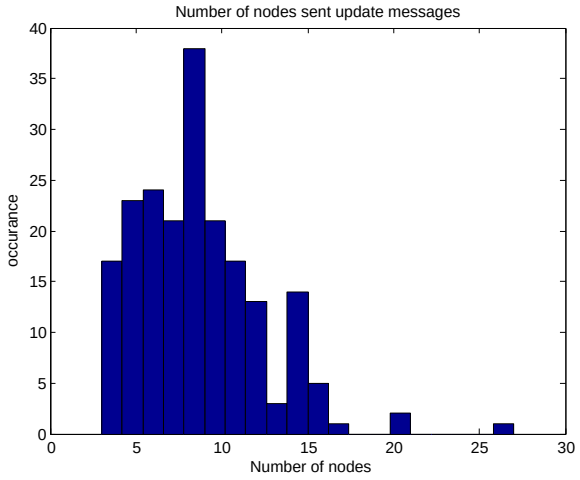
messages. In most cases, the failure of one nodes does not cause any power changes at all as depicted in figure 8.4(c) because STC has already provided redundancy in the network. However, in some rare cases, failure of a middle node can cause a major change in the network structure and over half of the nodes in the network issue broadcast messages. In all the simulated cases, the networks remain connected after STC optimisation.

## 8.2 Addition of Sensor Nodes

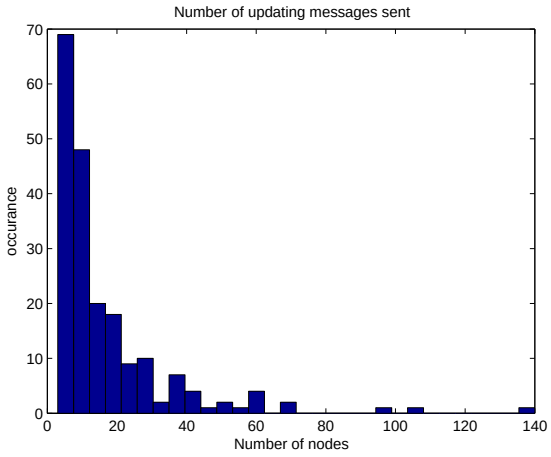
In this section, we investigate the effect of adding nodes to the network. The addition of sensor nodes should not affect the connectivity of the network, however, it increases the local node density and node degree in the neighbourhood. Hence, the neighbouring nodes may choose to decrease their power to bring the average node degree down, or maintain their power if the local metric is still acceptable.

We reverse the process of the previous section to simulate the effect of the addition of a node. We assume that at the beginning of the simulation node 41 is faulty and then it becomes alive at time instance 70. The optimisation results before node 41 comes alive is very similar to the one in Figure 8.5(b). However, when node 41 becomes active, the final result is slightly different from the original optimised network from Figure 8.1, which was optimised when all the nodes were active. This is because all the nodes around node 41 have their metric requirement satisfied before it joins and only node 41 needs to increase its power until its own metric is satisfied. In this simulation, we have set $k_L$ to 5 and $k_H$ to 50. None of the surrounding nodes need to decrease their powers to keep their parameters within the boundary.

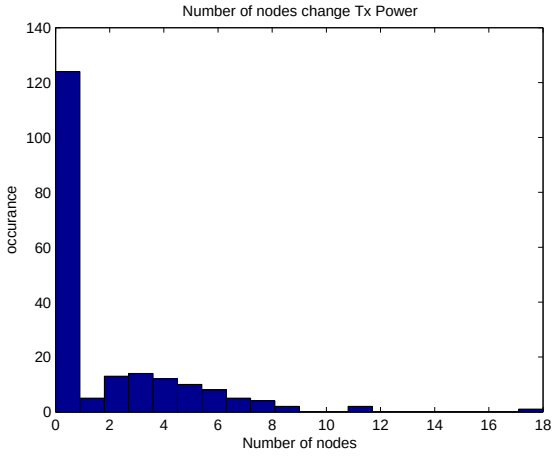
In the simulated results shown in Figure 8.6(c), the number of nodes involved in the addition of one node is a lot less than removal of nodes in the last section. The purpose of the update messages are mainly to inform the neighbourhood that there is a new node in the network. There is a margin between $k_L = 5$ and $k_H = 50$ to




(a) Number of Nodes send out update broadcast messages.



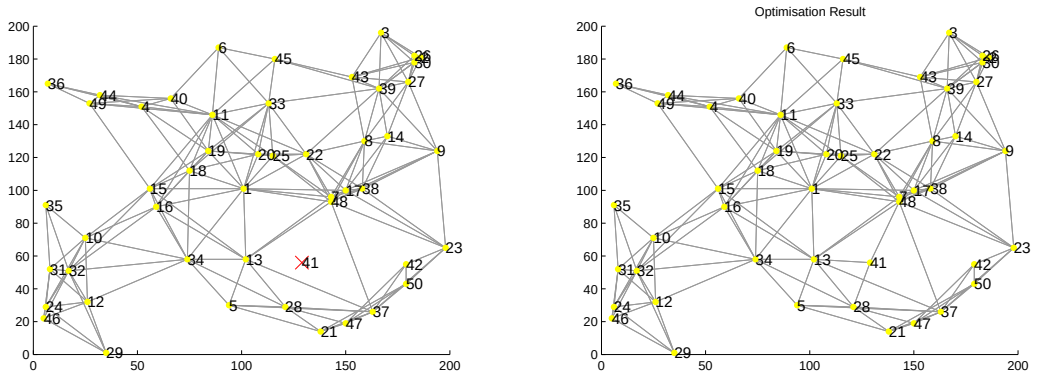

(b) Number of total messages sent for the update.




(c) Number of Nodes that change their transmission power.

Figure 8.4: Histograms of 200 simulation results for removal of a node in the network. $k_L = 5$, $k_H = 50$, $n = 50$ and the network size is $200x200$

prevent oscillation in the network and hence, the nodes do not reduce their Tx power unless the local area becomes very congested indicated by the metric exceeding the $k_H$ threshold.




(a) optimised network

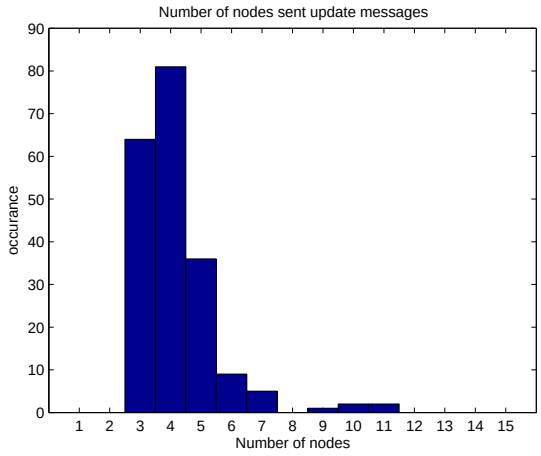
(b) Addition of sensor node

Figure 8.5: Simulation results for addition of 1 node in the network, showing the network before and after the additional node

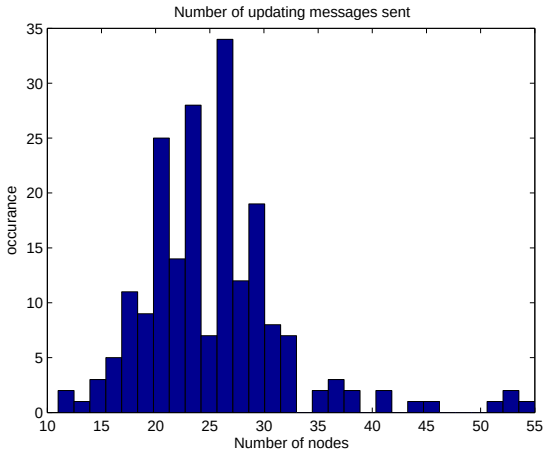
## 8.3 Deployment Scenario

In all our simulations so far, we have assumed that all the nodes are in place at the beginning and STC starts to optimise the transmission power of all nodes at the same time. This is unlikely in the deployment scenario, unless the nodes are initially deployed in the field without broadcasting and the base station then issues a command for the nodes to start performing STC at the same time. However, before STC takes place the nodes would need to be transmitting at maximum power to ensure connectivity, which results in power wastage and creates contention problem with our CSMA scheme.

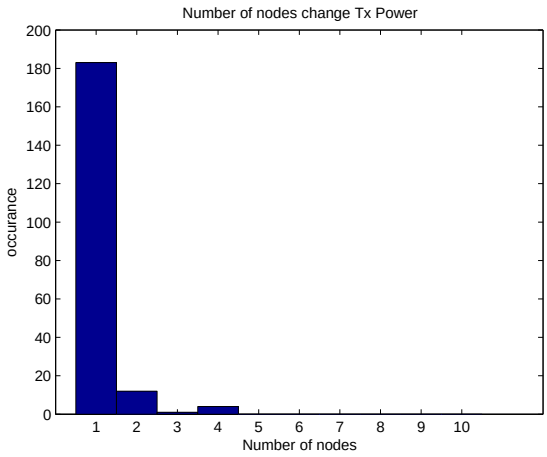
Hence, in this section, we simulate the actual scenario where the nodes are deployed one by one. We assume that the node are deployed at one minute rate, and each iteration of STC takes 10 seconds. The 10 seconds iterative period is taken from the current duty cycles in our sensor nodes. Nodes are put to sleep unless they need to communicate with each other to conserve power. Hence, we assume that the nodes are deployed every 6 STC cycles, which is equal to 60 seconds. We have tested the algorithm with the deployment rate between 1 (10 seconds) and 60 duty cycles




(a) Number of Nodes send out update broadcast messages.




(b) Number of total messages sent for the update.

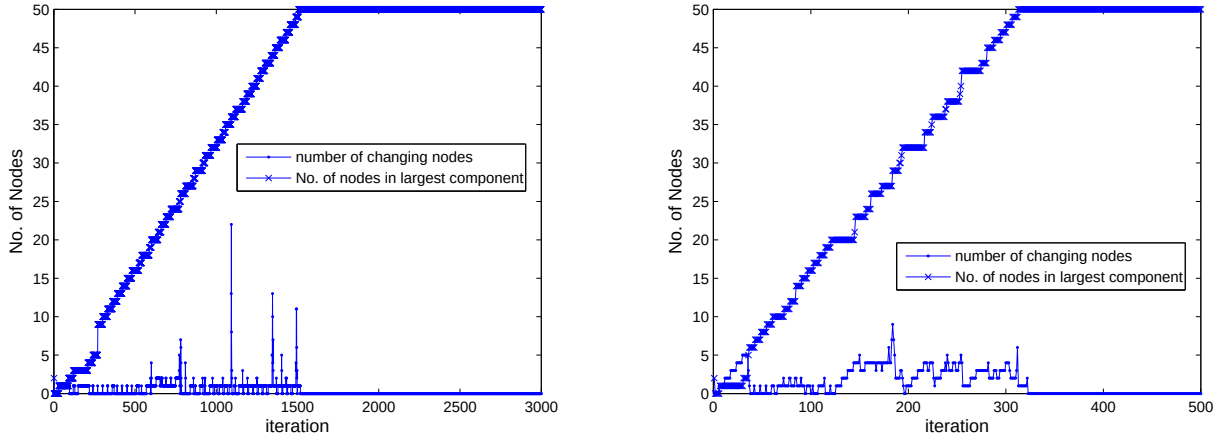



(c) Number of Nodes that change their transmission power.

Figure 8.6: Histograms of 200 simulation results for removal of a node in the network. $k_L = 5$, $k_H = 50$, $n = 50$ and the network size is $200 \times 200$

(10 minutes) and they show very similar results, but with a longer or shorter settling time proportional to the deployment rate.

In Figure 8.7 we show the dynamics of the node deployment over time at 1 and 5 minute deployment rates. We see that both scenario will eventually result in connectivity. There are always nodes changing their Tx power at the 1-minute



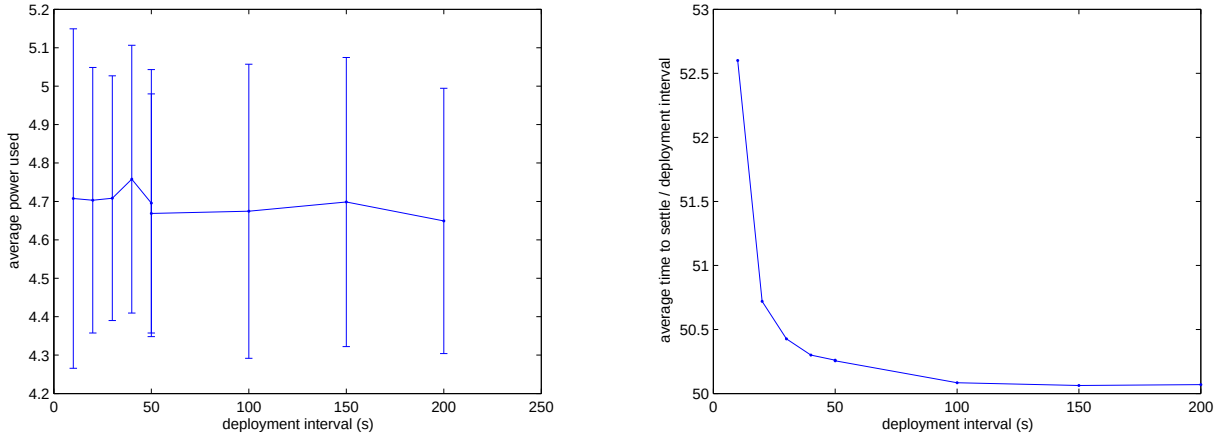

(a) Node deployment rate = 5 minutes

(b) Node deployment rate = 1 minute

Figure 8.7: Number of changing nodes. Each iteration takes 10 seconds

deployment rate; whilst the number of changing nodes are observed in bursts in the 5-minute deployment rate as the long time between deployment allows the network to settle before new nodes are added. The changes are restricted to the local area unless the newly added node causes a major increase in the local node density. The neighbouring nodes reduce their transmission powers, which in turn causes further changes in the network. We observe such major change at time instance 1100 in Figure 8.7(a) where 23 of the 36 deployed nodes change their transmission power.

The result of the deployment experience is shown in Figure 8.8. We observe that the deployment rate/ interval has little effect on the final average power in the resulting optimised network, as depicted in Figure 8.8(a). Figure 8.8(b) depicts the relationship between the time to settle normalised by the deployment interval and the deployment interval itself. The normalised settling time is dependent on the deployment interval and converges to the number of nodes in the network, which is 50 in this case, for large intervals. When the deployment rate is slow, the network spends most of the time staying at the last stabilised condition and only reacts when new nodes are added.




(a) Average power used

(b) Time to settle

Figure 8.8: Statistics for the deployment scenario where 50 nodes are deployed one by one separated by the time intervals. Each point is the averaged result of 50 simulations.

These results have demonstrated that STC can handle different deployment rates from 1 to 200 seconds, and should continue maintain the performance for deployment intervals more than 200 seconds.

## 8.4 Link Failure

In this chapter, we have demonstrated that STC is a dynamic algorithm, which reacts to scenario where the nodes are deployed in the field one by one; and to changes in the environment with the addition and removal of sensor nodes over time.

In fact, STC will be able to react to changes in the link dynamic as well as position drift of the sensor nodes which are not simulated in this chapter. To a sensor node, if a link fails and then recovers within the predefined time required for it to be regarded as a lost neighbour[1], then this would not cause a topology change as the neighbour tables are not altered. The temporarily failed link should not affect message relaying

[1]All nodes periodically broadcast a beacon/heartbeat, if a node is not heard for $l$ times it is regarded as lost, see Section 8.1.

as the topology created by STC provides redundancy and alternative paths in the network for combating this type of instability.

If a link has failed for longer than the allowed period and a neighbour $v$ is regarded as lost from node $u$, $u$ will evaluate its own metric and see if it still satisfies the $k_L$ requirement. The failure link may or may not cause node $u$ to increase its Tx power depending on its resulting metric without node $v$. In the case when node $u$ increases the Tx power to connect to further neighbours, it may not need to reduce its Tx power when the connection to $v$ revived. This is because of the margin between $k_L$ and $k_H$, which is important to ensure network stability.

## 8.5 Conclusion to Topology Control

Subgraph Topology Control (STC) presents a novel topology control algorithm in two ways. Firstly, a new control parameter subgraph Number $\eta_4$ is proposed for use in topology control, which is obtained from nodes exchanging neighbours tables with immediate neighbours and is a measure of local redundancy. The goal of STC is to generate a connected network with high probability and resilience as the underlying physical topology, and do not require special hardware such as location and direction determination.

STC is an iterative algorithm that nodes slowly increase or decrease transmission power to improve their connectivity until each node has satisfied their local connectivity requirement. STC sends out probabilistic update request to detect network changes and only nodes that are local to the change will be involved in the update. The probability and efficiency of the update should be related to node mobility and system changing rate and would require further investigation. The results has shown that the time taken for the system to reach stability depends on mostly node density.

In our experiments, we found that by setting $k_L = 3$, the networks are connected with probability greater than 0.97 for $50 < n < 200$. This gives us an average node

degree of 6.1. Setting a larger $k_L$ value can improve connectivity, however, would also increase the network power consumption and node degree. In fact, setting $k_L \geq 7$ results in 100% connectivity in all our simulations for $50 < n < 200$, and the resulting node degree is around 8. From equation 6.1.1 we can see that the relation is $\eta_4 \in O(R^3)$, therefore, node degree rises very slowly with the increase of range.

When compare to the performance of k-neighbours algorithm, we found that STC has higher probability of connectivity with similar resulting degree and using similar transmission power in 2-dimensional networks. STC also produces higher minimum degree, which means that the resulting network is most resilience.

STC is iterative in nature. For a network density of $0.01m^{-2}$, STC takes less than 18 iterations to settle for this particular type of radio. It can take up to 30 iterations for node density less than $0.001m^{-2}$. However, from Figure 7.4(a) we can see that the majority of the nodes settle their power a lot more quickly than this depending on their local density.

We simulated the effect of the network dynamic to the performance of STC. STC adapts well when nodes are added or removed from the network. It can also adapt to the case when nodes are added to the network one by one. Moreover, because of the iterative nature of STC, only local nodes are involved in the adaptation. This, makes STC very attractive in terms of power saving and ensures scalability.

# Part IV

# Conclusion

# Chapter 9

# Conclusion and Future Work

## 9.1 Conclusion

The work in this thesis was initiated by the SECOAS wireless sensor network (WSN) project. The aim of the project was to collect oceanographic data to analyse the impact of the on-shore windfarm to the nearby sandbank movement. The project focused on developing a robust communication structure that could cope with different sea conditions and deriving an energy efficient data gathering strategy. The WSN must be power efficient such that it would last for months at a time before batteries are required to be replaced.

My contributions to the project and the research in WSN is divided into two parts, data handling and topology control. They are concluded upon separately below.

### 9.1.1 Data Handling

In this part of the thesis, a thorough time series data analysis for the SECOAS project was included to establish an understanding to the environment and to verify the results of the algorithms. The analysed parameters are temperature, conductivity, pressure and turbidity (sediment concentration). The data were analysed for their short and medium term characteristics.

Prior to the data analysis,the performance of 3 Hurst estimators, Periodogram,

Least Square Fit Estimator (LSE) and Wavelet Weighted Least Square Fit Estimator (WLSE), were studied in the presence of linear trends and periodicity so that the validity of the estimators are known. It was found that WLSE is the most robust estimator against linear trends. When a period is present and its amplitude is small (less than 1:1 to the False Gaussian Noise (fGn)), periodogram is the most reliable predictor. However, when the periodic signal amplitude is large, all predictors overestimate the hurst parameter.

It was found that all four oceanographic parameters exhibited periodic characteristics contributed by tides. The pressure data also shows short term periods of wave movement, which can be characterised by spectral periods and spectral width. Temperature, conductivity and pressure data also show long term trend, which is likely to be the yearly variation of the environment. These trends and periods are important features that are required to be preserved in the data compression and sampling strategies.

Wavelet multi-resolution analysis was used to break down the sediment concentration data by wavelet scales. A different perspective of the data in terms of time is observed in each wavelet scale. This has helped us to separate the signal from noise in each scale and to apply wavelet threshold compression technique to the data.

Two methods of finding the thresholds for suppressing the wavelet coefficients were proposed, first by using a fix ratio of the largest coefficient(FRC), and secondly by using deviation from a normal distribution (DND) derived from the coefficient distribution. Both methods can achieve a compression ratio of over 45% and incur very small error. Depending on the data quality requirement, a compression ratio up to 90% can be achieved. FRC gives more consistent compression ratio over time compared to DND. However, DND is better at preserving anomalies than FRC. FRC is preferred in the SECOAS project because it is simpler to implement than DND and it gives comparable compression ratios.

We applied Hurst estimators to the turbidity data and found that there is strong evidence that self-similarity is present in the short term data sets. The Hurst parameter ($H$) averages at 0.89 for all the sets and this indicates the presence of long-range dependency (LRD) ($H > 0.5$). The presence of self-similarity characteristics has assisted us to analyse the performance of the adaptive sampling algorithms.

The development of an adaptive sampling scheduler was then reported. A sampling scheduler can be viewed as an alternative solution to data compression, in that data are only sampled if they are predicted to be useful to the user. It is a mean of conserving battery power by sampling less often and reducing the amount of transmitted data. Most existing adaptive sampling algorithms require the use of a prediction model, which has restricted their use to the specific environments they are designed for. We sought an alternative algorithm that can be operated on most data types.

Basic Adaptive Sampling Scheduler (BASS) is developed for adapting the sampling rate based on environmental changes and for rare-event detection. BASS is a 2-stage state-machine, which calculates the probability of sampling and idling using the statistic Exponentially-Weighted Moving Average (EWMA). The most recent measurement and the thresholds for event detection are used for the adaption mechanism. It also has a spatial aspect to quicken up event detection by neighbour coordination. BASS can be adapted for absolute and relative threshold detection.

We compared the performance of BASS and eSense and found that eSense shows better results for data with fast periods, i.e. SECOAS pressure data; while BASS performs better for data that is deviated from normal distribution, i.e. turbidity data. Both eSense and BASS produce similar results on the temperature and conductivity data. Despite the differences, BASS is the preferable algorithm in the SECOAS project because it does not rely on a prediction model. eSense requires model training and re-training for prediction accuracy. This requires substantial memory require-

ment, which may be performed on a central node, however, this would make eSense a less scalable solution. Moreover, eSense shows poor performance if the data is deviated from the random walk model, while BASS is consistent with all data types.

### 9.1.2 Topology Control

In a WSN, nodes are connected to neighbouring nodes within their transmission range. There is an incentive to reduce the transmission power in each node to conserve power and to reduce node degree in the local area. This is because high node degree results in low traffic throughput in a collision based multiple access scheme. On the other hand, it is important to ensure connectivity in the network and that the network has redundancy for alternative routes. Topology control (TC) studies the formation of a WSN network structure by adjusting the transmission power in individual nodes.

The search for a suitable TC control parameter, which can accurately report the local connectivity was first reported. Existing Algorithms that use the angle of arrival (CBTC) and location (XTC and LMST) can guarantee connectivity in the network, however, SECOAS sensor nodes are not equipped with directional antennas and location-aware devices and hence, these algorithms are not applicable. The algorithm k-neighbours uses the number of neighbours as the control parameter, which is easily available in sensor nodes. However, k-neighbours does not guaranteed connectivity and is particularly prone to the island phenomenon, where a cluster of highly dense nodes that are far away from the rest of the network would form an unconnected island.

Subgraph Topology Control (STC) was introduced using subgraph number $\eta_4$ as the control parameter, which is deduced by the node's neighbour tables and its neighbour's neighbour tables. Although STC does not guarantee connectivity, the increased depth of information given by the 2nd hop neighbours compared to node

degree has helped to improve judgement on local connectivity as compared to node degree.

STC is an iterative algorithm. The node starts with one transmission range, and then increment or decrement its power according to the reported $\eta_4$ until $\eta_4$ is within $k_L$, the lower boundary and $k_H$, the upper boundary.

We evaluated STC against k-neighbours and critical transmission range (CTR), which is the minimum homogenous power in the network for connectivity. It was found that in all our simulated 2 dimensional networks, STC produces better connectivity (STC> 95% using $k_L = 2$ as compared to k-neighbours> 91% using $k = 8$, and $STC > 99.8\%$ using $k_L = 5$ as compared to k-neighbours> 99% using $k = 10$) whilst using less transmission power in the nodes. Moreover, STC generates networks with higher *minimum degree*, which indicates the networks have better resilience toward node and link failure than the k-neighbours networks.

Although STC initially takes a longer time to reach stability compared to k-neighbours due to its iterative nature. Studies in the adaptivity of STC for adding and removing nodes have shown that only the local neighbourhood is involved in the updating of the change, which makes STC a very scalable solution. We have also simulated that STC can adjust quickly to node and link failure and the deployment scenario where nodes are added to the network one by one.

## 9.2 Future Work

### 9.2.1 Data Compression

We suggest the following future investigation for the data compression strategy.

- The focus on the compression strategy in SECOAS should be put in developing method 1, FRC compression. Further work can be carried out on the investigation of the relationship between the threshold percentage and compression

ratio. This will enable the users to adjust the data quality, which leads to the corresponding compression ratios and errors.

- The effect of the number of wavelet scales used in DWT on the compression results can be investigated. Moreover, the use of different wavelets with a higher number of vanishing moments can be more beneficial than the Haar wavelet, as wavelets with higher number of vanishing moments are known to give more consistent results.

- At the moment sediment level is obtained by summarising the data over a period, e.g. 20 minutes by mean and standard deviation. This is a kind of compression to send back statistics instead of raw data. We can compare the compression ratio, quality of reconstructed data and robustness in a lossy communication environment using wavelet compression and statistics and observe the difference.

- From the result of section 3.4.1 we see a linear relationship between temperature and conductivity values. We can use this property for further data compression.

### 9.2.2 Adaptive Sampling Scheduler

We suggest the following future investigation for the data sampling scheduler.

- The four oceanography data sets are not ideal for the experiments. The smallest granularity of the baseline data is 1 hour and hence, the variation and trend between data points are missed out. These missing trends are important to maximise the performance of BASS. Continuous data with higher frequency (e.g. 1 per second) can be collected to observe and compare the performance of BASS.

- Data sets, such as embankment tilt readings for landslide detection can be used as experiment to observe BASS's ability for rare event detection

- The delay of detection of the three sampling mechanisms in relation to the rate of change in data should be investigated in the presence of an event. Miss ratio can reflect part of the delay of detection, but cannot accurately describe it.

- The threshold parameter $\rho$ needs to be adjusted for different type of data in the experiment. We should be able to adapt this parameter to the current data measured similar to adapting $s$. One possibility is to combine $\rho$ and $s$ into one parameter.

- The adaptation of $s$ to relative and absolute thresholds may be combined to simplify the algorithm.

- The advantage of the spatial aspect of BASS has not been fully investigated due to the lack of spatial data in our experiment. The introduction of spatial feature should significantly improve the delay and the miss ratio of the BASS algorithm.

### 9.2.3 Topology Control

The major goal of topology control in this thesis concerns the tradeoffs between connectivity and energy consumption. There are, however, other factors that may be related to topology control that can be investigated.

- We have investigated topology control for connectivity. There are other aspects of topology control, which optimise the network based on data traffic and routing [GBH+06]. We can combine both and get a more complete picture of the network structure.

- We have used one subgraph $\eta_4$ in our algorithm because it represents redundancy in the network. We may be able to use other subgraphs to replace or complement $\eta_4$ in the algorithm.
- We can evaluate STC further in terms of node mobility.
- We can investigate the use of distributed biological algorithm such as game theory, greedy algorithms, genetic algorithm, etc to obtain an optimal balance between energy cost, traffic, connectivity and other factors.

# Appendix A

# Map of Scroby Sands

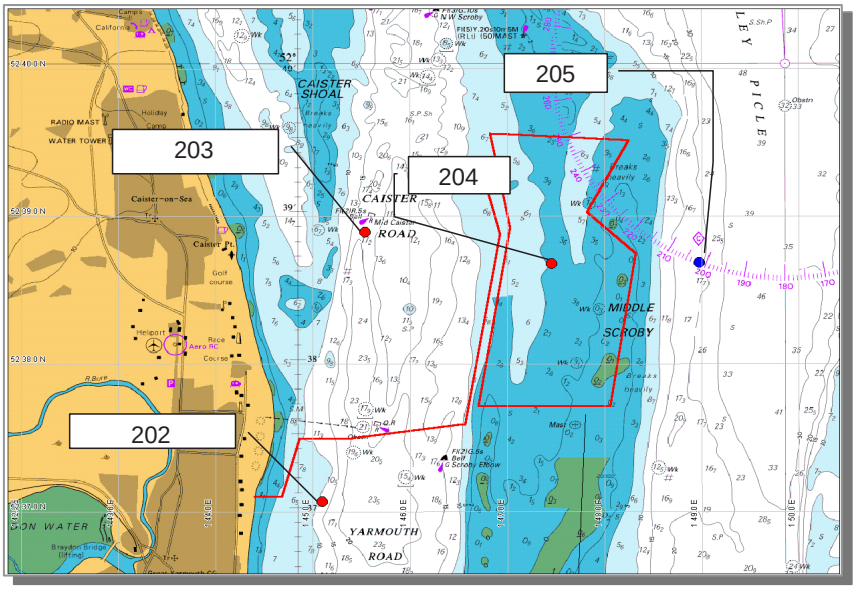



Figure A.1: map of Scroby Sands and location of the data loggers

# Bibliography


[ADJ+02] M. Alan, C. David, P. Joseph, S. Robert, and A. John. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages pp. 88–97, Atlanta, Georgia, USA, 2002.

[APP93] P. Abry, P.Goncalves, and P.Flandrin. Wavelet-based spectral analysis of 1/f processes. In *Proc. IEEE-ICASSP93*, volume vol. III, pages P.237–240, 1993.

[ASSC02] I. Akyildiz, W. Su, T. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, August 2002.

[AV98] P. Abry and D. Veitch. Wavelet analysis of long range dependent traffic. *Trans. Info. Theory*, Vol. 44, No.1:pp. 2–15, Jan 1998.

[AWSC02] I.F. Akyildiz, W.Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, Vol 38:p.393–422, 2002.

[BBB04] J. Burrell, T. Brooke, and R. Beckwith. Vineyard computing: sensor networks in agricultural production. In *Pervasive Computing, IEEE*, volume 3, pages pp. 38–45, 2004.

[BCar] A. Boukerche and I. Chlamtac. Broadcasting and topology control in wireless ad hoc networks. *Handbook of Algorithms for Mobile and Wireless Networking and Computing, CRC Press*, to appear.

[BD96] Peter J. Brockwell and Richard A. Davis. *Introduction to Time Series and Forcasting*. Springer-Verlag New York, Inc., 1996.

[Ber92] Jan Beran. Statistical methods for data with long-range dependency. *Statistical Science,*, Vol 7, No.4:pp.402–416, Nov 1992.

[Ber98] Jan Beran. *Statistics for Long-Memory Processes*. Chapman & Hall CRC Press, 1998.

[BLRS03] Douglas M. Blough, Mauro Leoncini, Giovanni Resta, and Paolo Santi. The k-neighbors approach to interference bounded and symmetric toplogy control in ad hoc networks. In *Proceedings of ACM MobiHoc 03*, 2003.

[BLRS06] Douglas M. Blough, Mauro Leoncini, Giovanni Resta, and Paolo Santi. The k-neighbors approach to interference bounded and symmetric topology control in ad hoc networks. *IEEE Transactions on Mobile Computing*, 5(9):1267–1282, 2006.

[CAHS05] Qing Cao, Tarek Abdelzaher, Tian He, and John Stankovic. Towards optimal sleep scheduling in sensor networks for rare-event detection. In *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*, page 4, Piscataway, NJ, USA, 2005. IEEE Press.

[EEHDP04] C. C. Enz, A. El-Hoiydi, J-D. Decotignie, and V. Peiris. Wisenet: An ultralow-power wireless sensor network solution. *IEEE Computer*, 37, Issue:8, August 2004.

[EG01] G. M. Edelman and J. A. Gally. Degeneracy and complexity in biological systems. *Proc Natl Acad Sci U S A*, 98(24):13763–13768, November 2001.

[ET01] William J. Emery and Richard E. Thomson. *Data Analysis Methods in Physical Oceanography*. Elsevier Science B.V., 2001.

[EYE] Eyes project, http://eyes.eu.org/cons.htm.

[GBH+06] Q Gao, K J Blow, D J Holding, I Marshall, and X H Peng. Radio range adjustment for energy efficient wireless sensor networks. *Ad-Hoc Networks*, Vol. 4(ISSN: 1570-8705.):pp 75 – 82., January 2006.

[GEH02] Deepak Ganesan, Deborah Estrin, and John Heidemann. Dimensions: Why do we need a new data handling architecture for sensor networks? In *Proceedings of the ACM Workshop on Hot Topics in Networks*, pages 143–148, Princeton, NJ, USA, October 2002. ACM.

[GGP+03] D. Ganesan, B. Greenstein, D. Perelyubskiy, D. Estrin, and J. Heidemann. An evaluation of multi-resolution storage for sensor networks. In *Proceedings of first ACM conference on embedded networked sensor systems*, 2003.

[Gra95] Amara Graps. An introductions to wavelets. In *IEEE computational science and engineering*, volume vol.2, 1995.

[Haw99] Mohammed F. Hawa. A study of deterministic trends in web-cache traffic using wavelet analysis. In *London Communications Symposium*, 1999.

[HCH00] W. Heinzelman, A. Chandrakasan, and H.Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the Hawaii International Conference on System Sceinces*, January 4-7, 2000.

[HKB99] Wendi Rabiner Heinzelman, Joanna Kulik, and Hari Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Mobile Computing and Networking*, pages 174–185, 1999.

[HLTC06] Chi-Fu Huang, Li-Chu Lo, Yu-Chee Tseng, and Wen-Tsuen Chen. Decentralized energy-conserving and coverage-preserving protocols for wireless sensor networks. *ACM Trans. Sen. Netw.*, 2(2):182–187, 2006.

[HSI+01] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, and D. Ganesan. Building efficient wireless sensor networks with low-level naming. In *Proceedings of symposium on operating systems principles*, pages pp. 146–159, Oct 2001.

[IGE+03] Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, John Heidemann, and Fabio Silva. Directed diffusion for wireless sensor networking. *IEEE/ACM Trans. Netw.*, 11(1):2–16, 2003.

[LCS06] Haiyang Liu, Abhishek Chandra, and Jaideep Srivastava. esense:energy efficient stochastic sensing framework for wireless sensor platforms. *Proceedings of the fifth international conference on Information processing in sensor networks*, pages p.235–242, 2006.

[LHB+05] Li Li, Joseph Y. Halpern, Paramvir Bahl, Yi-Min Wang, and Roger Wattenhofer. A cone-based distributed topology-control algorithm for wireless multi-hop networks. *IEEE/ACM Trans. Netw.*, 13(1):147–159, 2005.

[LHS05] Ning Li, Jennifer C. Hou, and Lui Sha. Design and analysis of an mst-based topology control algorithm. *IEEE Transactions on Wireless Communications*, 4(3):1195–1206, May 2005.

[LKR04] G. Lu, B. Krishnamachari, and C.S. Raghavendra. An adaptive energyefficient and low-latency mac for data gathering in wireless sensor networks. *Proceedings of 18th International Parallel and Distributed Processing Symposium*, page 224, 2004.

[LL07] Jinbao Li and Jianzhong Li. Data sampling control, compression and query in sensor networks. *International Journal of Sensor Networks*, 2 No. 1/2:53–61, 2007.

[LM99] Meng-Jang Lin and Keith Marzullo. Directional gossip: Gossip in a wide area network. In *European Dependable Computing Conference*, pages 364–379, 1999.

[LQW04] P. Lin, C. Qiao, and X. Wang. Medium access control with a dynamic duty cycle for sensor networks. *IEE Wireless Communications and Networking Conference,*, 3:1534 – 1539, 2004.

[Mat04] Wavelet toolbox users guide, http://www.mathworks.com/access/helpdesk/help/toolbox/wavelet/wavelet.shtml. Mathwork, 2004.

[May79] Peter S. Maybeck. *Stochastic models, estimation and control*, volume Vol.1, Chapter 1. Academic Press, 1979.

[MFH02] S. Madden, M. Franklin, and J. Hellerstein. Tag: a tiny aggregation service for ad hoc sensor networks. In *5th Annual Symposium on Operating Systems Design and Implementation (OSDI)*, Dec, 2002.

[MH03] Gerhard Masselink and Michael Hughes. *Introduction to coastal processes & Geomorphology*. Oxford University Press Inc., 2003.

[MHO04] Kirk Martinez, Jane K. Hart, and Royan Ong. Environmental sensor networks. *IEEE Computer*, 37 Issue 8:50–56, Aug 2004.

[MIK+04] Ron Milo, Shalev Itzkovitz, Nadav Kashtan, Reuven Levitt, Shai Shen-Orr, Inbal Ayzenshtat, Michal Sheffer, and Uri Alon. Superfamilies of evolved and designed networks. *Science*, 303(5663):1538–1542, March 2004.

[MTL88] Hedetniemi S. M., Hedetniemi S. T., and Liestman A. L. A survey of gossiping and broadcasting in communication networks. *Networks*, 18:319–349, 1988.

[Net06] Dust Networks. Technical overview of time synchronised mesh protocol (tsma). 2006.

[Pen97] M. D. Penrose. The longest edge of the random minimal spanning tree. *Annals of Applied Probability*, vol. 7, no. 2, 1997.

[Pen99] Mathew D. Penrose. On k-connectivity for a geometric random graph. *Random Struct. Algorithms*, 15(2):145–164, 1999.

[PHC+03] Jianping Pan, Y. Thomas Hou, Lin Cai, Yi Shi, and Sherman X. Shen. Topology control for wireless sensor networks. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 286 – 299, 2003.

[PK00] G. J. Pottie and W. J. Kaiser. Wireless integrated network sensors. *Commun. ACM*, 43(5):51–58, 2000.

[PMMM04] G. Powell, D. Marshall, R. Milliken, and K. Markham. A data fusion system for object recognition based on transferable belief models and kalman filter. In *Proceedings of International conference on information fusion*, Stockholm, Sweden, June 2004.

[PMR+05] P. Padhy, K. Martinez, A. Riddoch, H. L. R. Ong, and J. K. Hart. Glacial environment monitoring using sensor networks. In *Real-World Wireless Sensor Networks*, Stockholm, Sweden, 2005.

[PW00] Donald B. Percival and Andrew T. Walden. *Wavelet Methods for Time Series Analysis*. Cambridge University Press, 2000.

[Rap02] T. S. Rappaport. *Wireless communications principles and practices*. Prentice-Hall, 2002.

[Res] Wolfarm Research. Mathworld. http://mathworld.wolfram.com/.

[RLW+04] S. Ren, Q. Li, H. Wang, X. Chen, and Zhang X. Probabilistic coverage for object tracking in sensor networks. Poster session, Mobicom 2004, 2004.

[RSE+06] N. Ramanathan, T. Schoellhammer, D. Estrin, M. Hansen, and M. Srivastava T. Harmon, E. Kohler. The final frontier: Embedding networked sensors in the soil. Technical report, CENS Technical Report 68, November 2006.

[San] Paolo Santi. The critical transmitting range for connectivity in mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 4(3):pp. 310–317.

[San05a] Paolo Santi. *Topology Control in Wireless Ad Hoc and Sensor Networks*. John Wiley and Sons, Ltd, England, 2005.

[San05b] Paolo Santi. Topology control in wireless ad hoc and sensor networks. *ACM Comput. Surv.*, 37(2):164–194, 2005.

[SB03] Paolo Santi and Douglas M. Blough. The critical transmitting range for connectivity in sparse wireless ad hoc networks. *IEEE Transactions on Mobile Computing*, 2(1):25–39, 2003.

[SBM+00] David C. Steere, Antonio Baptista, Dylan McNamee, Calton Pu, and Jonathan Walpole. Research challenges in environmental observation and forecasting systems. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 292–299, New York, NY, USA, 2000. ACM Press.

[SBW+03] L. Sacks, M. Britton, I. Wokoma, A. Marbini amd T. Adebutu, I. Marshall, C. Roadknight, J. Tateson, D. Robinson, and A. Velazque. The development of a robust, autonomous sensor network platform for environmental monitoring. *IoP Sensors & their Applications (S&A XII)*, 2003.

[SC07] R. Sherwood and S. Chien. Sensor web technologies: A new paradigm for operations. In *International Symposium on Reducing the Cost of Spacecraft Ground Systems and Operations (RCSGSO 2007)*, Moscow, Russia, June 2007.

[SCA] Scadd: Scalable coordinate architecture for deeply distributed systems, http://www.isi.edu/scadds/.

[SCI+01] Eugene Shih, Seong-Hwan Cho, Nathan Ickes, Rex Min, Amit Sinha, Alice Wang, and Anantha Chandrakasan. Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks. pages 272–287, 2001.

[SGAP00] K. Sohrabi, J. Gao, V. Ailawadhi, and G.J. Pottie. Protocols for self-organization of a wireless sensor network. *IEEE Personal Communications*, 7, no. 5:16–27, Oct 2000.

[SM97] P. Stoica and R.L. Moses. *Introduction to Spectral Analysis*. Prentice-Hall, 1997.

[SOP+04] R. Szewczyk, E. Osterweil, J. Polastre, M Hamilton, and D. Estrin A Mainwaring. Habitat monitoring with sensor networks. volume 47, pages 34–40, June 2004.

[SPMC04] Robert Szewczyk, Joseph Polastre, Alan Mainwaring, and David Culler. Lessons from a sensor network expedition. In *1st European Workshop on Wireless Sensor Networks (EWSN '04)*, pages 307–322, Berlin, Germany, January 2004.

[Spr03] Julien Clinton Sprott. *Chaos and Time Series Analysis*. Oxford University Press, 2003.

[SR03] P. Smets and B. Ristic. Kalman filter and joint tracking and classification in the tbm framework. In *Proceedings of the 6th International conference on information fusion*, June 2003.

[STA04] Nist/sematech e-handbook of statistical methods, http://www.itl.nist.gov/div898/handbook/, 2004.

[STE00] O. Sporns, G. Tononi, and G.M. Edelman. Theoretical neuroanatomy: Relating anatomical and functional connectivity in graphs and cortical connection matrices. *Cereb. Cortex*, 10(2):127–141, 2000.

[Ste03a] Robert H. Stewart. *Introduction to Physical Oceanography.* Department of Oceanography, Texas A&M University, 2003.

[Ste03b] Robert H. Stewart. Introduction to physical oceanography. 2003.

[SWA+04] L. Shum, I. Wokoma, T. Adebutu, A.Marbini, L. Sacks, and M. Britton. Distributed algorithm implementation and interaction in wireless sensor network. 2004.

[TM06] D. Tulone and S. Madden. Paq: Time series forecasting for approximate query answering in sensor networks. *LECTURE NOTES IN COMPUTER SCIENCE*, 3868:21–37, 2006.

[TT97] V. Teverovsky and M. Taqqu. Testing for long range dependence in the presence of shifting means or a slowly declining trend, using a variance type estimator. *Journal of Times Series Analysis*, Vol 18 No3, 1997.

[VA99] Darry Veitch and Patrice Abry. A wavelet based joint estimator of the parameters of long-range dependence. *IEEE Trans. Information Theory*, Vol. 45, No. 3, 1999.

[WAJR+05a] G. Werner-Allen, J. Johnson, M. Ruiz, J. Lees, and M. Welsh. Monitoring volcanic eruptions with wireless sensor network. *Proceedings of Second European Workshop on Wireless Sensor Network*, pages pp. 108–120, Istanbul, 2005.

[WAJR+05b] Geoffrey Werner-Allen, Jeff Johnson, Mario Ruiz, Jonathan Lees, and Matt Welsh. Monitoring volcanic eruptions with wireless sensor network. In *Proceedings of Second European Workshop on Wireless Sensor Network*, Istanbul, Feb 2005.

[WLP+02] I Wokoma, I Liabotis, O Prnjat, L Sacks, and I Marshall. A Weekly Coupled Adaptive Gossip Protocol for Application Level Active Networks. In *3rd International Workshop on Policies for Distributed Systems and Networks (POLICY 02)*, Monterey, California, unknown 2002.

[WSSM05] I. Wokoma, L. Shum, L. Sacks, and I. W. Marshall. A biologically-inspired clustering algorithm dependent on spatial data in sensor networks. In *2nd European Workshop on Wireless Sensor Networks*, Istanbul, Turkey, 2005.

[WY04] Peng-Jun Wan and Chih-Wei Yi. Asymptotic critical transmission radius and critical neighbor number for k-connectivity in wireless ad hoc networks. In *MobiHoc '04: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, pages 1–8, New York, NY, USA, 2004. ACM Press.

[WZ04] R. Wattenhofer and A. Zollinger. Xtc: A practical topology control algorithm for ad-hoc networks. In *18th International Parallel and Distributed Processing Symposium (IPDPS 2004)*, 2004.

[XK04] Feng Xue and P. R. Kumar. The number of neighbors needed for connectivity of wireless networks. *Wirel. Netw.*, 10(2):169–181, 2004.

[YHE02] Wei Ye, J. Heidemann, and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. In *INFOCOM 2002. Twenty-First*

*Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1567– 1576, 2002.