BAYESIAN NETWORK BASED INTELLIGENT MOBILITY STRATEGIES FOR WIRELESS SENSOR NETWORKS

MATTHEW DAVID COLES

A thesis submitted in partial fulfilment of the requirements for the award of the degree of Doctor of Philosophy of the University of Portsmouth

March 2009
Abstract

This thesis is concerned with the design and analysis of new Bayesian network based mobility algorithms for mobile Wireless Sensor Networks (WSNs). The hypothesis for the work presented herein is that incorporating Artificial Intelligence (AI) at the level of the sensor nodes will improve their performance (coverage, connectivity and lifetime) and result in fault tolerance capabilities, in the face of uncertainty associated with incomplete information regarding the network.

Two types of mobility strategy are presented and investigated. Firstly, a new gazing mobility strategy is presented which is biologically inspired from herbivores grazing pastures. As part of the latter strategy, and instead of deploying a large number of static sensor nodes to cover a region of interest, a smaller number of mobile nodes are deployed which migrate around the region to achieve coverage over time. To enable the performance evaluation of this strategy a new coverage measure called Coverage Against Time was created. A new decentralised Bayesian network based grazing mobility algorithm called BNGRAZ is presented which uses evidence derived from neighbouring nodes to predict the probability of performance (coverage and connectivity) changes associated with moving in a particular direction. Evidence is also obtained from a new Coverage Approximation (CA) algorithm which enables each sensor node to approximate the WSN coverage in order to determine areas in need of servicing. The performance of BNGRAZ is compared to a fixed path mobility technique, Random Waypoint (RWP) mobility model, and a new Grazing Reference Point Group Mobility (GRPGM) algorithm developed as part of this work.

Secondly, a self-healing strategy which physically relocates sensor nodes to repair coverage holes, due to the failure of sensor nodes, is presented. A new decentralised Bayesian network based mobility algorithm called BayesMob, which uses local neighbour information, was created to coordinate the self-healing motion. The algorithm enables sensor nodes to predict the probability of an increase in coverage given a move in a particular direction to repair coverage holes.

In addition, the thesis outlines the development of a WSN simulator. The latter provides a tool for evaluating the performance of mobile WSNs. All mobility strategies and algorithms discussed herein were simulated using the new WSN simulator.
Declaration

Whilst registered as a candidate for the above degree, I have not been registered for any other research award. The results and conclusions embodied in this thesis are the work of the named candidate and have not been submitted for any other academic award.
Acknowledgements

I would like to express my gratitude to my supervisors Dr. Djamel Azzi and Dr. Barry Haynes for their continued support and advice.

Also I would like to thank Dr. Alan Hewitt for his useful suggestions.

Lastly I would like to thank my family and friends for their continued support and unwavering belief in me.
Table of Contents

Abstract........................................................................................................................................... i
Declaration......................................................................................................................................... ii
Acknowledgements.................................................................................................................. iii
Table of Contents...................................................................................................................... iv
Glossary........................................................................................................................................... viii
List of Figures............................................................................................................................ ix
List of Tables...................................................................................................................................... xvii
Chapter 1.......................................................................................................................................... 1
INTRODUCTION .......................................................................................................................... 1
  1.1 Wireless Sensor Networks................................................................................................. 1
    1.1.1 Sensing Module........................................................................................................... 6
    1.1.2 Processing Module...................................................................................................... 6
    1.1.3 Communication Module.............................................................................................. 7
  1.2 Mobile Wireless Sensor Networks.................................................................................... 9
    1.2.1 Motivation for Implementing a Mobile WSN........................................................... 10
    1.2.2 Challenges when Implementing a Mobile WSN....................................................... 12
  1.3 Applications........................................................................................................................ 14
    1.3.1 Military Applications................................................................................................. 15
    1.3.2 Environmental Applications...................................................................................... 16
    1.3.3 Health and Home Applications................................................................................ 17
    1.3.4 Other Commercial Applications............................................................................. 17
  1.4 Objective and Overview of the Thesis.............................................................................. 18
Chapter 2......................................................................................................................................... 21
LITERATURE REVIEW.................................................................................................................. 21
  2.1 Introduction.......................................................................................................................... 21
  2.2 WSN Simulators................................................................................................................ 21
    2.2.1 SENSE_2.0.............................................................................................................. 22
    2.2.2 The Network Simulator (Ns2).................................................................................. 22
    2.2.3 Glomosim.................................................................................................................. 23
  2.3 QoS and Performance Evaluation...................................................................................... 24
<table>
<thead>
<tr>
<th>Chapter 2</th>
<th>Mobile WSNs</th>
<th>31</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.3.1</td>
<td>Coverage</td>
<td>26</td>
</tr>
<tr>
<td>2.3.2</td>
<td>Connectivity</td>
<td>27</td>
</tr>
<tr>
<td>2.3.3</td>
<td>Lifetime</td>
<td>28</td>
</tr>
<tr>
<td>2.4</td>
<td>Mobile WSNs</td>
<td>31</td>
</tr>
<tr>
<td>2.4.1</td>
<td>Mobility Models</td>
<td>31</td>
</tr>
<tr>
<td>2.4.2</td>
<td>Accommodating the Physical Relocation of Sensor Nodes Associated with Mobile WSNs</td>
<td>32</td>
</tr>
<tr>
<td>2.4.3</td>
<td>Controlled Mobility</td>
<td>34</td>
</tr>
<tr>
<td>2.5</td>
<td>Artificial Intelligence</td>
<td>44</td>
</tr>
<tr>
<td>2.5.1</td>
<td>Intelligence – the general consensus</td>
<td>44</td>
</tr>
<tr>
<td>2.5.2</td>
<td>Artificial Intelligence – definition and trends</td>
<td>45</td>
</tr>
<tr>
<td>2.5.3</td>
<td>AI and WSN</td>
<td>47</td>
</tr>
<tr>
<td>2.5.4</td>
<td>Bayesian Networks</td>
<td>48</td>
</tr>
<tr>
<td>2.6</td>
<td>Summary</td>
<td>52</td>
</tr>
</tbody>
</table>

Chapter 3 | WSN Simulator Development | 53 |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Introduction</td>
<td>53</td>
</tr>
<tr>
<td>3.2</td>
<td>A New WSN Simulation Environment (WSN Simulator)</td>
<td>53</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Motivation</td>
<td>53</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Simulator Engine &amp; Code Structure</td>
<td>54</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Performance Evaluation</td>
<td>68</td>
</tr>
<tr>
<td>3.3</td>
<td>Summary</td>
<td>73</td>
</tr>
</tbody>
</table>

Chapter 4 | Coverage Against Time | 75 |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>75</td>
</tr>
<tr>
<td>4.2</td>
<td>Motivation</td>
<td>75</td>
</tr>
<tr>
<td>4.3</td>
<td>Concept of Coverage Against Time</td>
<td>76</td>
</tr>
<tr>
<td>4.4</td>
<td>Performance Metrics</td>
<td>78</td>
</tr>
<tr>
<td>4.5</td>
<td>Summary</td>
<td>80</td>
</tr>
</tbody>
</table>

Chapter 5 | Grazing Mobility Strategy | 81 |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Introduction</td>
<td>81</td>
</tr>
<tr>
<td>5.2</td>
<td>Principle of WSN Grazing</td>
<td>81</td>
</tr>
<tr>
<td>5.3</td>
<td>Design Considerations</td>
<td>83</td>
</tr>
</tbody>
</table>
5.4 Mobility Algorithms ...................................................................................... 84
  5.4.1 Fixed Path Mobility Algorithm ................................................................. 84
  5.4.2 Grazing Reference Point Group Mobility (GRPGM) Algorithm .............. 86
  5.4.3 BNGRAZ Algorithm ................................................................................. 89
  5.4.4 Simulation & Results ............................................................................... 111
5.5 Summary ...................................................................................................... 131

Chapter 6 ............................................................................................................ 132
SELF-HEALING MOTION ........................................................................................ 132
  6.1 Introduction .................................................................................................. 132
  6.2 The Principle of Self-healing ...................................................................... 132
  6.3 Bayesian network Mobility (BayesMob) ..................................................... 134
  6.4 Simulation & Results ................................................................................... 137
    6.4.1 Fixed Deployment .................................................................................... 138
    6.4.2 Random Deployment ............................................................................... 139
    6.4.3 Energy Considerations .......................................................................... 141
  6.5 Summary ...................................................................................................... 141

Chapter 7 ............................................................................................................ 143
CONCLUSION AND FUTURE WORK ...................................................................... 143
  7.1 Conclusions .................................................................................................. 143
  7.2 Future Work ................................................................................................. 147

References ........................................................................................................... 148
Appendix A ........................................................................................................... 157
PUBLICATIONS ....................................................................................................... 157
Appendix B ........................................................................................................... 178
SOURCE CODE ....................................................................................................... 178
Appendix C ........................................................................................................... 179
ANGLE PROBABILITY DENSITY FUNCTION ....................................................... 179
Appendix D ........................................................................................................... 182
CA ALGORITHM MAXIMUM ERROR .................................................................. 182
Appendix E ........................................................................................................... 185
CONDITIONAL PROBABILITY TABLES ................................................................. 185
  E.1 BNGRAZ CPT ........................................................................................... 185
    E.1.1 Bn1 Probability of a Decrease in Connectivity .................................. 185
    E.1.2 Bn2 probability of discovering un-serviced area ......................... 187
E.1.3  Bn3 Probability of Maintaining or Increasing Performance ............. 188
E.2 BayesMob CPT ......................................................................................... 189
Appendix F ....................................................................................................... 191
ANGLE DISTRIBUTION APPROXIMATION ...................................................... 191
Appendix G ....................................................................................................... 197
SERVICE INTERVAL DISTRIBUTIONS................................................................. 197
  G.1 Varying Sensor Node Speed ................................................................. 197
  G.2 Varying the Size of the WSN (number of sensor nodes) ...................... 199
  G.3 Varying the Configuration Duty Time .................................................. 204
  G.4 Varying the Size of the Vicinity Around the Reference Point .............. 209
  G.5 Desired Coverage Period ...................................................................... 212
  G.6 MDT (BNGRAZ) .................................................................................... 214
Glossary

The following abbreviations are used throughout this thesis:

- **WSN**: Wireless Sensor Network
- **CN**: Commander Node (mobile base station)
- **EDR**: Energy Discharge Rate
- **GRPGM**: Grazing Reference Point Group Mobility
- **RPGM**: Reference Point Group Mobility model
- **RWP**: Random Way-Point mobility model
- **BNGRAZ**: Bayesian Network GRAZing
- **BayesMob**: Bayesian network Mobility
- **CA**: Coverage Approximation
- **CIT**: Contacts Information Table
- **GPS**: Global Positioning System
- **MAC**: Medium Access Control
- **CoFi**: Coverage Fidelity self-healing algorithm
- **μ**: Mean
- **SD**: Standard Deviation also referred to as sigma (σ)
- **p.d.f.**: probability density function
- **MDT**: Motion Decision Threshold
- **CPT**: Conditional Probability Table
List of Figures

Figure 1.1 Overview of a WSN ................................................................. 2
Figure 1.2 Architecture of a generic sensor node hardware platform ........... 4
Figure 1.3 Example of Mica motes and the Golem Dust (Smart Dust project) ... 4
Figure 1.4 WSN communication protocol stack ......................................... 8
Figure 2.1 Example of a Bayesian network .................................................. 50
Figure 3.1 The interrelationship between the parts/functions of the WSN simulator ... 55
Figure 3.2 Flow diagram of the simulator’s initialisation and run procedure ........ 56
Figure 3.3 Flow diagram for concurrent.m ................................................... 57
Figure 3.4 The GUI of the WSN Simulator .................................................... 58
Figure 3.5 Concurrency within WSN simulator using time division of processor ... 61
Figure 3.6 Contacts Information Table (CIT), local to sensor nodes ............... 67
Figure 3.7 Generating the coverage map by the grid intersections (pixel points) using the user specified resolution .................................................. 69
Figure 3.8 Flow diagram for sim_conn_scan.m ............................................. 72
Figure 4.1 Example of the accumulated coverage value, indicating the coverage period .................................................................................... 77
Figure 4.2 Example of the service interval ................................................... 77
Figure 4.3 Variation between Coverage Period and Service Interval ............... 78
Figure 4.4 Example of pixel point service interval distributions .................... 79
Figure 4.5 Examples of coverage periods and mean, max service intervals ........ 80
Figure 5.1 Grass height relationships to the coverage weight value ................ 83
Figure 5.2 Predefined fixed path grazing mobility approach ......................... 86
Figure 5.3 BNGRAZ structure ................................................................. 91
Figure 5.4 Bayesian network 3 structure ................................................................. 92
Figure 5.5 Connectivity decrease Bayesian network 1 structure ......................... 93
Figure 5.6 Angle distribution with normal fit \((x \text{ or } y) \text{ SD } = 1 \text{ m}).................. 97
Figure 5.7 Angle distribution with normal fit \((x \text{ or } y) \text{ SD } = 31 \text{ m})............... 97
Figure 5.8 Angle distribution SD plot ...................................................................... 98
Figure 5.9 Angle distribution SD plot, including approximated values, using the angle SD model ........................................................................................................ 98
Figure 5.10 CA algorithm scalability plot ............................................................... 105
Figure 5.11 CA coverage plot for 10 nodes deployed ............................................. 107
Figure 5.12 CA coverage plot for 15 nodes deployed ............................................. 107
Figure 5.13 CA coverage error plot for 10 nodes deployed ..................................... 107
Figure 5.14 CA coverage error plot for 15 nodes deployed ..................................... 107
Figure 5.15 Discovering uncovered area Bayesian network 2 structure ............... 108
Figure 5.16 Fraction of un-covered pixel points in each cardinal direction .......... 109
Figure 5.17 Connectivity vs. sensor node speed ..................................................... 114
Figure 5.18 Coverage vs. sensor node speed ......................................................... 114
Figure 5.19 Mean service interval vs. sensor node speed ....................................... 115
Figure 5.20 Service interval SD vs. sensor node speed .......................................... 115
Figure 5.21 Lifetime vs. sensor node speed ............................................................ 116
Figure 5.22 Remaining energy vs. sensor node speed .............................................. 116
Figure 5.23 Connectivity vs. number of sensor nodes ........................................... 117
Figure 5.24 Coverage vs. number of sensor nodes ................................................. 117
Figure 5.25 Mean service interval vs. number of sensor nodes ......................... 118
Figure 5.26 Service interval SD vs. number of sensor nodes .............................. 118
Figure 5.27 Lifetime vs. number of sensor nodes ............................................... 119
Figure 5.28 Remaining energy vs. number of sensor nodes .............................. 119
Figure 5.29 Connectivity vs. desired coverage period ................................................. 120
Figure 5.30 Coverage vs. desired coverage period ..................................................... 120
Figure 5.31 Mean service interval vs. desired coverage period ................................... 120
Figure 5.32 Service interval SD vs. desired coverage period ................................... 120
Figure 5.33 Lifetime vs. desired coverage period .................................................... 121
Figure 5.34 Remaining energy vs. desired coverage period ...................................... 121
Figure 5.35 Connectivity vs. vicinity size ............................................................... 122
Figure 5.36 Coverage vs. vicinity size .................................................................... 122
Figure 5.37 Mean service interval vs. vicinity size ................................................... 122
Figure 5.38 Service interval SD vs. vicinity size ...................................................... 122
Figure 5.39 Lifetime vs. vicinity size ..................................................................... 123
Figure 5.40 Remaining energy vs. vicinity size ........................................................ 123
Figure 5.41 BNGRAZ connectivity vs. configuration duty time ............................. 124
Figure 5.42 BNGRAZ coverage vs. configuration duty time ..................................... 124
Figure 5.43 Mean service interval vs. configuration duty time ................................... 125
Figure 5.44 Service interval SD vs. configuration duty time ..................................... 125
Figure 5.45 BNGRAZ lifetime vs. configuration duty time ....................................... 125
Figure 5.46 BNGRAZ remaining energy vs. configuration duty time ..................... 125
Figure 5.47 BNGRAZ connectivity vs. configuration duty time ............................. 126
Figure 5.48 BNGRAZ coverage vs. configuration duty time ..................................... 126
Figure 5.49 Mean service interval vs. configuration duty time ............................... 127
Figure 5.50 Service interval SD vs. configuration duty time ....................................... 127
Figure 5.51 BNGRAZ connectivity vs. MDT ............................................................ 128
Figure 5.52 BNGRAZ coverage vs. MDT ................................................................. 128
Figure 5.53 BNGRAZ mean service interval vs. MDT ............................................. 128
Figure 5.54 BNGRAZ service interval SD vs. MDT .................................................. 128
Figure 5.55 BNGRAZ lifetime vs. MDT ................................................................. 129
Figure 5.56 BNGRAZ remaining energy vs. MDT ................................................ 129
Figure 6.1 The self-healing of a WSN ................................................................. 133
Figure 6.2 Coverage increase Bayesian network structure ............................. 135
Figure 6.3 Coverage loss plot for a fixed deployment of 64 nodes ................. 139
Figure 6.4 Coverage loss plot for a fixed deployment of 81 nodes ................. 139
Figure 6.5 Coverage loss plot for a fixed deployment of 100 nodes ............... 139
Figure 6.6 Coverage loss plot for a random deployment of 64 nodes ............. 140
Figure 6.7 Coverage loss plot for a random deployment of 81 nodes ............. 140
Figure 6.8 Coverage loss plot for a random deployment of 100 nodes .......... 140
Figure 6.9 Motion energy overhead plot for CoFi vs. BayesMob ................. 140
Figure C.1 One function of two random variables (region where inequality $y/x \leq z$ is satisfied) .................................................. 180
Figure D.1 Coverage error for the CA algorithm ........................................... 183
Figure F.1 Flow diagram for $x\_y\_angle.m$ function ................................... 191
Figure F.2 Angle distribution with normal fit ($x$ or $y$) SD = 1 m ............... 192
Figure F.3 Angle distribution with normal fit ($x$ or $y$) SD = 1.5 m ............. 192
Figure F.4 Angle distribution with normal fit ($x$ or $y$) SD = 2 m ............... 192
Figure F.5 Angle distribution with normal fit ($x$ or $y$) SD = 2.5 m ............ 192
Figure F.6 Angle distribution with normal fit ($x$ or $y$) SD = 3 m ............... 192
Figure F.7 Angle distribution with normal fit ($x$ or $y$) SD = 3.5 m ............ 192
Figure F.8 Angle distribution with normal fit ($x$ or $y$) SD = 4 m ............... 193
Figure F.9 Angle distribution with normal fit ($x$ or $y$) SD = 4.5 m ............ 193
Figure F.10 Angle distribution with normal fit ($x$ or $y$) SD = 5 m ............. 193
Figure F.11 Angle distribution with normal fit ($x$ or $y$) SD = 6 m ............. 193
Figure F.12 Angle distribution with normal fit ($x$ or $y$) SD = 7 m ............. 193
Figure F.13 Angle distribution with normal fit (x or y SD = 8 m) .................................. 193
Figure F.14 Angle distribution with normal fit (x or y SD = 9 m) .................................. 194
Figure F.15 Angle distribution with normal fit (x or y SD = 10 m) ............................... 194
Figure F.16 Angle distribution with normal fit (x or y SD = 11 m) ............................... 194
Figure F.17 Angle distribution with normal fit (x or y SD = 12 m) ............................... 194
Figure F.18 Angle distribution with normal fit (x or y SD = 13 m) ............................... 194
Figure F.19 Angle distribution with normal fit (x or y SD = 15 m) ............................... 194
Figure F.20 Angle distribution with normal fit (x or y SD = 17 m) ............................... 195
Figure F.21 Angle distribution with normal fit (x or y SD = 19 m) ............................... 195
Figure F.22 Angle distribution with normal fit (x or y SD = 21 m) ............................... 195
Figure F.23 Angle distribution with normal fit (x or y SD = 23 m) ............................... 195
Figure F.24 Angle distribution with normal fit (x or y SD = 25 m) ............................... 195
Figure F.25 Angle distribution with normal fit (x or y SD = 27 m) ............................... 195
Figure F.26 Angle distribution with normal fit (x or y SD = 29 m) ............................... 196
Figure F.27 Angle distribution with normal fit (x or y SD = 31 m) ............................... 196
Figure F.28 Angle SD against x or y SD ........................................................................ 196
Figure G.1 BNGRAZ service interval distributions ....................................................... 197
Figure G.2 GRPGM service interval distributions ......................................................... 198
Figure G.3 RWP service interval distributions ............................................................... 198
Figure G.4 Fixed path service interval distributions ....................................................... 199
Figure G.5 BNGRAZ 9 nodes service interval distribution ............................................ 199
Figure G.6 BNGRAZ 15 nodes service interval distribution ......................................... 199
Figure G.7 BNGRAZ 20 nodes service interval distribution ......................................... 200
Figure G.8 BNGRAZ 25 nodes service interval distribution ......................................... 200
Figure G.9 BNGRAZ 30 nodes service interval distribution ......................................... 200
Figure G.10 GRPGM 9 nodes service interval distribution ............................................ 200

xiii
Figure G.11 GRPGM 15 nodes service interval distribution ....................................... 200
Figure G.12 GRPGM 20 nodes service interval distribution ....................................... 201
Figure G.13 GRPGM 25 nodes service interval distribution ....................................... 201
Figure G.14 GRPGM 30 nodes service interval distribution ....................................... 201
Figure G.15 RWP 9 nodes service interval distribution .............................................. 201
Figure G.16 RWP 15 nodes service interval distribution ............................................ 201
Figure G.17 RWP 20 nodes service interval distribution ............................................ 202
Figure G.18 RWP 25 nodes service interval distribution ............................................ 202
Figure G.19 RWP 30 nodes service interval distribution ............................................ 202
Figure G.20 fixed path 9 nodes service interval distribution ....................................... 202
Figure G.21 fixed path 15 nodes service interval distribution ....................................... 202
Figure G.22 fixed path 20 nodes service interval distribution ....................................... 203
Figure G.23 fixed path 25 nodes service interval distribution ....................................... 203
Figure G.24 fixed path 30 nodes service interval distribution ....................................... 203
Figure G.25 BNGRAZ configuration duty time 10 s (speed 0.1 m/s) ........................ . 204
Figure G.26 BNGRAZ configuration duty time 30 s (speed 0.1 m/s) ........................ . 204
Figure G.27 BNGRAZ configuration duty time 60 s (speed 0.1 m/s) ........................ . 204
Figure G.28 BNGRAZ configuration duty time 90 s (speed 0.1 m/s) ........................ . 204
Figure G.29 BNGRAZ configuration duty time 120 s (speed 0.1 m/s) ........................ . 205
Figure G.30 BNGRAZ configuration duty time 10 s (speed 0.05 m/s) ...................... . 205
Figure G.31 BNGRAZ configuration duty time 30 s (speed 0.05 m/s) ...................... . 205
Figure G.32 BNGRAZ configuration duty time 60 s (speed 0.05 m/s) ..................... .. 205
Figure G.33 BNGRAZ configuration duty time 90 s (speed 0.05 m/s) ..................... .. 205
Figure G.34 BNGRAZ configuration duty time 120 s (speed 0.05 m/s) ................... .. 206
Figure G.35 BNGRAZ configuration duty time 10 s (15 nodes) .............................. .. 206
Figure G.36 BNGRAZ configuration duty time 30 s (15 nodes) .............................. .. 206
Figure G.37 BNGRAZ configuration duty time 60 s (15 nodes) ..................................... 207
Figure G.38 BNGRAZ configuration duty time 90 s (15 nodes) ..................................... 207
Figure G.39 BNGRAZ configuration duty time 120 s (15 nodes) .................................... 207
Figure G.40 BNGRAZ configuration duty time 150 s (15 nodes) .................................... 207
Figure G.41 BNGRAZ configuration duty time 180 s (15 nodes) .................................... 207
Figure G.42 BNGRAZ configuration duty time 210 s (15 nodes) .................................... 207
Figure G.43 BNGRAZ configuration duty time 240 s (15 nodes) .................................... 207
Figure G.44 BNGRAZ configuration duty time 270 s (15 nodes) .................................... 207
Figure G.45 BNGRAZ configuration duty time 300 s (15 nodes) .................................... 208
Figure G.46 GRPGM vicinity size 30×30 m² (9 nodes) .................................................. 209
Figure G.47 GRPGM vicinity size 35×35 m² (9 nodes) .................................................. 209
Figure G.48 GRPGM vicinity size 40×40 m² (9 nodes) .................................................. 209
Figure G.49 GRPGM vicinity size 45×45 m² (9 nodes) .................................................. 209
Figure G.50 GRPGM vicinity size 50×50 m² (9 nodes) .................................................. 210
Figure G.51 GRPGM vicinity size 55×55 m² (9 nodes) .................................................. 210
Figure G.52 GRPGM vicinity size 60×60 m² (9 nodes) .................................................. 210
Figure G.53 GRPGM vicinity size 30×30 m² (15 nodes) ................................................ 210
Figure G.54 GRPGM vicinity size 35×35 m² (15 nodes) ................................................ 210
Figure G.55 GRPGM vicinity size 40×40 m² (15 nodes) ................................................ 211
Figure G.56 GRPGM vicinity size 45×45 m² (15 nodes) ................................................ 211
Figure G.57 GRPGM vicinity size 50×50 m² (15 nodes) ................................................ 211
Figure G.58 GRPGM vicinity size 55×55 m² (15 nodes) ................................................ 211
Figure G.59 GRPGM vicinity size 60×60 m² (15 nodes) ................................................ 211
Figure G.60 BNGRAZ desired coverage period 120 min ............................................. 212
Figure G.61 BNGRAZ desired coverage period 240 min ............................................. 212
Figure G.62 BNGRAZ desired coverage period 360 min ............................................. 212
Figure G.63 BNGRAZ desired coverage period 420 min ........................................... 212
Figure G.64 BNGRAZ desired coverage period 420 min ........................................... 213
Figure G.65 GRPGM desired coverage period 120 min ............................................ 213
Figure G.66 GRPGM desired coverage period 240 min ............................................ 213
Figure G.67 GRPGM desired coverage period 360 min ............................................ 213
Figure G.68 GRPGM desired coverage period 420 min ............................................ 213
Figure G.69 GRPGM desired coverage period 480 min ............................................ 214
Figure G.70 BNGRAZ MDT = 25% ........................................................................ 214
Figure G.71 BNGRAZ MDT = 25.4% .................................................................... 214
Figure G.72 BNGRAZ MDT = 25.8% .................................................................... 215
Figure G.73 BNGRAZ MDT = 26.2% .................................................................... 215
Figure G.74 BNGRAZ MDT = 26.6% .................................................................... 215
Figure G.75 BNGRAZ MDT = 27% .................................................................... 215
Figure G.76 BNGRAZ MDT = 27.4% .................................................................... 215
List of Tables

Table 1.1 Mobile WSN design considerations (Advantages and Disadvantages of mobility) ......................................................................................................................... 13

Table 5.1 Definition of variables for BNGRAZ ............................................................................................. 91

Table 5.2 Simulation parameters ............................................................................................................. 111

Table 6.1 Definition of variables for BayesMob ..................................................................................... 135

Table E.1 Bn1 \( N_N - N_W \) CPT ........................................................................................................ 185

Table E.2 Bn1 \( C_i \) CPT ..................................................................................................................... 186

Table E.3 Bn2 \( D_i \) CPT ..................................................................................................................... 187

Table E.4 Bn3 \( N_N - N_W \) CPT ........................................................................................................ 188

Table E.5 Bn3 \( O_d \) CPT ..................................................................................................................... 188

Table E.6 \( N_N - N_W \) CPT ............................................................................................................... 189

Table E.7 Bn1 \( C_i \) CPT ..................................................................................................................... 190
Chapter 1

INTRODUCTION

This chapter introduces Wireless Sensor Networks (WSNs), including a brief history and common applications. The chapter also outlines the motivation behind creating mobility strategies and the design considerations related to mobile WSNs, with particular focus on the inherent benefits and implications in terms of performance. The chapter concludes with a statement of the objectives of the work underlying this thesis and an overview of the latter.

1.1 Wireless Sensor Networks

Advances in small, low-power, low-cost micro-electronic and Micro-Electro-Mechanical Sensor (MEMS) technology along with the advances in wireless communication, and digital electronics, have enabled the prospect of deploying large quantities of sensor nodes to form WSNs. A WSN is a collection of sensor nodes spatially distributed over a geographical region of interest. Sensor nodes are equipped with sensing, processing and communication capabilities, which allow them to collect, process, and exchange data relating to one or more variables. The sensor nodes may cooperate with their neighbours (within communication range) to form a temporary sub-network. WSN topologies are generally dynamic and decentralised.

Primitive forms of sensor networks were developed for military applications, such as the Sound Surveillance System (SOSUS) [1]. The latter was used during the Cold War in the early 1950s to detect and track Soviet submarines, with the help of acoustic sensors or hydrophones. Modern research into the field of sensor networks was revived around 1980 by the Defence Advanced Research Projects Agency (DARPA) under the Distributed Sensor Networks (DSN) program. The possibility to extend the research to sensor networks was considered together with some research on supporting components such as operating system and knowledge-based signal processing techniques [2]. In 2001 Kristofer S. J. Pister (University of California) introduced the concept of SMART DUST sensors and initiated the SMART DUST project [3], which was funded by...
DARPA. The SMART DUST project predicted sensor nodes could eventually be shrunk to the size of a grain of rice or even dust. Sensor networks are now generally considered as Wireless Sensor Networks (WSNs) due to the wireless communication capabilities adopted at the level of the node. Research and development into WSNs has been inspired by smaller computing chips, more capable sensing devices, ad-hoc wireless communication, and enhanced battery technology.

The prospect of creating WSNs promises many new application areas. Among these are environmental monitoring, battlefield surveillance, health monitoring, home automation, space exploration, chemical processing, and disaster relief. WSNs would essentially allow environments, habitats, industrial plants, and equipment to be remotely monitored from a terminal possibly situated miles from the phenomena of interest. The spatial separation properties would effectively reduce the risk to human operatives associated with monitoring/collecting data from dangerous geographical regions of interest. In addition to this, the presence of human operatives may contaminate the region of interest. Akyildiz et al [4] presented a survey of WSNs including research trends and outlines the foreseen challenges that need to be addressed.

A general overview of a WSN is shown in Figure 1.1. The geographical region of interest, which is represented by the grey cloud, would be deployed with a large number of sensor nodes. Deployment can be achieved by manual placement (generally...
by a human operative), airdrop (sprayed from a plane which flies over the region), or even sensor nodes physically moving to the region of interest. After deployment the sensor nodes would configure, at which point a number of essential tasks effective to operation of the network would be performed. Among these tasks are localisation, neighbour discovery, synchronisation, and the generation of routing tables. Once configured sensor nodes would enter an active state where they would monitor and collect sensing information relating to some phenomenon under analysis in the region of interest. The sensing types and possible modes of operation vary depending on the application and environment.

Acquisition of sensing data can be event driven, periodic and/or query driven. Under the event driven approach a sensor node collects and transmits sensing data in response to changes in the environment. Alternatively the sensor node can periodically acquire the sensor data. The sensor node may also respond to queries generated by the task manager. Hybrid solutions may also be implemented which adopt more than one data acquisition methods.

Sensor nodes can perform some pre-processing in terms of data conditioning and aggregation before transmitting the information to the task manager via the base station, from a communication point of view. A WSN is effectively an ad-hoc network where peer to peer communication is achieved through multi-hop routing. Sensor nodes forward data packets they receive from neighbouring nodes on to the base station. The type of multi-hop routing depends on the adopted routing protocol. The sensor nodes can enter sleep cycles to conserve energy and periodically wakeup to perform sensing and communication tasks. Sleep cycle patterns are achieved through collaboration with neighbouring nodes and controlled by the Medium Access Control (MAC) layer.

Sensor nodes can incorporate motion capabilities either via onboard hardware or external mobile entities (humans, animals or vehicles). The mobile sensor nodes would require localisation hardware, such as a Global Positioning System (GPS) device attached to the sensor node platform. Localisation is required for navigation and the position information related to collecting and transmitting (routing) the sensing data to the base station. A generic example of the sensor node platform is given in Figure 1.2, Hill et al [5]. Figure 1.3 shows the Berkeley Mica mote, Xbow [6] and the Golem Dust created as part of the Smart Dust project, Berkeley [3].
Sensor nodes are generally resource-constrained in terms of energy, memory, processing and communication (bandwidth, range etc) due to the physical size and self-contained nature of the devices. The constraint in terms of energy will limit the communication and sensing ranges, which determine the maximum Euclidean distance for transmission and accurately detecting a change in the phenomenon being sensed.

Energy is generally provided via onboard battery cells and/or some harvesting mechanism (energy harvesting via solar cells, power docking stations). The energy source must provide sufficient energy for the proposed network lifetime. WSNs are often envisaged as one-time deployable and should be able to operate in remote, possibly hostile, regions where replenishing/replacing the energy source via human
intervention may not be feasible. Efficient and effective utilisation of energy would directly relate to the WSN operational lifetime. Therefore, while traditional communication network protocols aim to achieve high Quality of Service (QoS) provisions, WSN protocols in general focus primarily on power conservation. Energy trade-off mechanisms need to be incorporated with respect to the sensing, processing, communication and possibly mobility to ensure the application criteria are met.

WSNs are envisaged to operate autonomously (depending on the type of application) whilst satisfying the desired application performance criteria (Quality of Service (QoS)). Among the performance measures for WSN are coverage, connectivity, throughput, scalability, lifetime and robustness. The performance will govern the operational lifetime of the network, which is the time period until the network fails to meet the application criteria. WSN have been proposed for applications that require an operational lifetime of weeks, months and possibly years [4].

WSNs are either homogeneous or heterogeneous. A homogeneous WSN consists of identical sensors with equal capacity in terms of sensing, computation, communication, and power. In a heterogeneous system, the WSN is not limited to identical sensors. Some sensor nodes may for instance collect image data whilst other sensors may collect audio signals. Also sensor nodes can have varying processing capabilities, power, mobility, and so on. These networks are generally more application specific and nodes that are less energy constrained would complete the tasks with the highest energy overhead.

Heterogeneous sensor nodes are generally deployed for WSN which adopt a clustering protocol. For example, the network would consist of two types of nodes say type 1 and type 2. The geographical region would first be deployed with a number of type 1 nodes. These would act as the clustering nodes and perform the general sensing and processing tasks. The region would then be deployed with an overlay of type 2 nodes, which act as the cluster-heads, presumably fewer in number but more powerful in terms of communication, processing, and energy. This approach was described by Duarte-Melo et al [7] and was used to extend the lifetime of the network through effective energy balancing.
1.1.1 Sensing Module

The sensing module incorporates at least one type of sensing device which enables the sensor node to effectively interpret its surrounding environment. The type of sensing invariably depends on the application criteria, see [8]. These sensing devices have an associated sensing range, which is defined as the Euclidean distance at which a node can accurately detect a change in the sensed variable. The sensing range considered in this work is assumed to be homogeneous to all sensor nodes.

Analogue sensors generally produce a raw continuous signal that corresponds to the physical phenomena being measured. This signal must be sampled and digitalised by some form of ADC, either on or off board the processing unit, to condition the signal for analysis and transmission. Analogue sensing devices will often require calibration and linearization. The processing unit must incorporate some form of compensation capabilities for sensor readings to be valid. Furthermore scaling can cause issues, where each sensor has different timing and voltage responses.

Digital sensing devices internally generate raw analogue signals, but generate a digital output, via internal ADC. Thus all the required compensation and linearization is generated internally.

These sensing devices communicate their signals to the processing module for possible pre-transmission processing and aggregation, via some standard chip to chip communication protocol. Sensor modules are generally capable of generating thousands of samples per second. However, this would provide large quantities of redundant data, and increase energy consumption unnecessarily. The devices must be tuned to produce adequate data, to satisfy the application criteria, and in a way that maintains low energy consumption requirement.

1.1.2 Processing Module

The processing module coordinates the operation of the sensor node throughout the lifetime of the network. Among the tasks performed by the module are: executing the communication protocols; controlling the radio device; pre-processing (possibly aggregation); interface with sensing devices; power management; execute node configuration protocols; coordinate any possible mobility strategies and coordinate
possible commands/queries from the task manager. In order for the sensing node to satisfy the cost and size expectations, the processing unit would have limited processing and memory capabilities. The memory requirements of the node are generally small; nodes would contain some basic operational program code and minimal storage for sensor data prior to transmission. The data storage facilities may also be required for multi-hop routing, where nodes forward data from neighbouring nodes destined for the base station.

1.1.3 Communication Module

The communication module generally consists of a transceiver which provides the means for transmission of data through the WSN. The WSN is generally connected in a mesh topology, where communication is achieved via multi-hop routing, thus neighbouring nodes relay data packets between source and destination.

Communication is essential to the operation of the WSN and is required to transmit sensor data packets to the task manager (user). It also enables the delivery of query and command messages from the task manager, configuration of the network, and possibly the monitoring of performance attributes.

The communication module carries the largest energy overhead for static WSNs. However, in a mobile WSN the motion overheads may exceed those due to communication. As the communication is one of the largest contributors to the sensor nodes’ Energy Discharge Rate (EDR) it must be coordinated to minimise the energy overheads.

The communication range relates directly to the transmission power; the larger the intensity of the signal the further it travels, Heinzelman et al [9]. This would depend on the deployment environment including possible non line of sight between transmitter and receiver. The relationship between power requirements and distance travelled is a polynomial with an exponent of between three and four. Other factors that determine the communication range include the sensitivity of the receiver circuit, the gain and efficiency of the antenna and the channel encoding mechanisms.

Communication bandwidth is also a common constraint found with wireless communication devices, more so than their wired counterparts. Bandwidth limits the
data throughput capabilities of the network, therefore bandwidth must be utilised effectively to enable data to be delivered with minimal latency. However unlike many other high performance networks, WSNs do not generally require high data throughput and 10 – 100 Kbps is sufficient for many applications. Signal to noise ratio can also affect the data throughput capabilities of the network. Therefore constraint mechanisms should be available to the communication module.

The generic communication protocol stack for WSNs, proposed by Akyildiz et al [4] is shown in Figure 1.4. The communication protocol stack includes the following layers: application, transport, network, Medium Access Control (MAC) and physical. Also the following planes are included: power management, mobility management, task management, and coordination.

![Figure 1.4 WSN communication protocol stack](image)

The application layer includes the application software and may include the following protocols: sensor network management, task assignment, data advertisement, sensor query, and data dissemination. The transport layer deals with the end to end transmission of data through the network and is imperative for reliable communication. The network layer handles routing the data supplied by the transport layer. The sensor nodes will generally operate in noisy environments where node mobility may be possible. Therefore the MAC layer should be power aware and able to minimise collision with neighbourhood broadcasts. Collisions require retransmission and thus
increase communication overheads. The physical layer handles the raw data transmission over the communication medium and handles the modulation, transmission and receiving techniques.

The power, mobility, coordination and fault management planes monitor and coordinate the power, movement and task distribution among the sensor nodes. The above planes and associated protocols ensure that communication is achieved in an energy efficient manner.

1.2 Mobile Wireless Sensor Networks

A mobile WSN is created when a proportion of the sensor nodes (including the base station) are capable of physically changing position within the geographical region of interest. This mobility can be categorised into three types: random, predictable and controlled, see Kansal et al [10]. Hybrids may exist when two or more of the categories are adopted by the WSN. Random motion means that the sensor nodes may move in any direction, at any speed between a maximum and minimum, for an unspecified time period. Predictable motion refers to cases where sensor nodes have a deterministic roaming path. Therefore a sensor node's positions at time $t+1$, $2$, $3$ are known at time $t$. This may be evaluated by the sensor node's position at time $t-1$. Controlled motion in effect, indicates that the sensor nodes' motion characteristics are coordinated in a decentralised or centralised manner perhaps in response to external stimuli.

Mobility models can be either synthetic models or trace files. Synthetic models are based on strategies, algorithms and roaming patterns. Trace files allow application specific mobility to generate mobility models for WSN sensor nodes.

Traditionally mobility within WSNs has been envisaged as an additional management overhead that may impair the performance of the network, Kansal et al [10]. The mobility would generally be due to sensor nodes being attached to external mobile entities, such as animals, humans, vehicles; or nodes may float and follow current flows of rivers and oceans. This would generally relate to random and predictable mobility and would be represented by some appropriate mobility model. Consequently the sensor nodes may not directly influence the motion characteristics and additional protocols may need to be incorporated into the network to ensure that the network's performance satisfies the application criteria. For example the MAC and network layer
may need to incorporate additional functionality to accommodate the sensor nodes' mobility.

Mobility may be achieved by attaching a mobility module to the sensor node platform. This would generally consist of motors and servos which would enable the sensor node to physically move. This is referred to as controlled motion and coordinating the mobility module would be via motion strategies and algorithms executed by the processing unit.

The motion considered in this thesis is controlled whereby the sensor nodes and/or the base station explicitly coordinate the WSN motion characteristics. Mobility would therefore be achieved by incorporating mobility hardware directly onto the sensor node platform. The level of mobility may range from a few sensor nodes being mobile to all the nodes in the network. Other scenarios may exist where the sensor nodes are static whilst the base station/data sink is mobile.

1.2.1 Motivation for Implementing a Mobile WSN

Generally the deployment of static WSNs is random or deterministic. For example, sensor nodes may be air dropped (sprayed from a plane flying over the region of interest) or manually placed by human operatives. The selected deployment strategy would be governed by environmental and application criteria. A WSN which is deployed using a random approach would incorporate a high level of redundancy to ensure the network provides total coverage, even in the event of some node failures. This high redundancy would increase the costs associated with deploying the network. A deterministic approach to deployment, where nodes are manually place by human operatives, may not be feasible when considering remote uninhabited regions. In addition, there is a high risk that the environment may be contaminated or danger to the human operative. There is also the cost implications associated with employing the human operatives.

In these situations the optimum deployment configuration could be achieved by implementing a mobile WSN, Wang et al [11] and [12], which enable the network to adapt to application and environment changes. A possible approach is to deploy the mobile sensor nodes to the outer periphery of the region of interest. The sensor nodes would then move into the region of interest and generate an optimum deployment
configuration. The deployment could be coordinated by the network designer, alternatively it may be autonomous (sensor nodes would cooperate to generate the deployment configuration). This dynamic approach to deployment could reduce the amount of environmental and regional information the network designer requires prior to deploying the WSN.

Mobile WSNs have the ability to adapt to runtime dynamics such as moving to avoid obstacles and environmental noise (for example, vibrations from vehicles passing through the region of interest), Kansal et al [13]. The application criteria may even change during the post-deployment phase, thus requiring the WSN to physically reconfigure via moving part or all of the WSN elements. Also the number of sensor nodes may not be sufficient to cover the region of interest and therefore a mobile WSN, where sensor nodes migrate around the region of interest collecting the sensing data, would provide the desired coverage over time.

Sensor nodes can randomly fail due to malfunction, malicious destruction or energy exhaustion. These failures may create coverage holes or disconnect a subsection of the network thus reducing the operational lifetime of the network, effectively resulting from the network failing to satisfy the application criteria. A static WSN would rely on a high level of redundancy to counteract these failures. However a mobile WSN would carry fault tolerance capabilities where the sensor nodes would physically move to recover from the failure(s) and maintain performance, Ganeriwal et al [14] and Wang et al [15].

Mobility can prevent the deterioration in performance via balancing or reducing the sensor nodes’ energy overheads. Due to the ad-hoc nature of these networks, an edge effect may be observed where the neighbouring nodes of the base station expend more energy due to relaying data packets from nodes situated at the outer periphery. Balancing these energy overheads may be achieved via swapping sensor node positions or repositioning the base station to regions with the highest communication traffic, Rao et al [16].

Mobility may also be used to increase the resolution of sensing, Butler and Rus [17]. This is achieved by the repositioning of sensor nodes to regions with the highest activity. The desired increase in resolution may be event or query driven. Static WSN
would rely on redundancy to achieve the latter. Also sensor nodes may move to reinforce communication paths and thus reduce the communication overheads.

All of the above make the investigation of mobility worthwhile however the challenges and issues discussed below should be considered.

1.2.2 Challenges when Implementing a Mobile WSN

The addition of mobility hardware essentially increases the size, complexity and cost of the sensor node platform. Mobile sensor nodes would have additional hardware such as motors and servos to allow them to physically reposition themselves within a region of interest. This will increase the size of the sensor node platform and increase its exposure to malicious attacks and interception (reduce the sensor nodes' stealth capabilities). The motion hardware must be energy efficient such that it does not use vast amounts of energy during motion. As previously stated, the largest energy overhead for a static network is communication. However for a mobile network the motion overhead would surpass communication.

The energy source for the nodes would in many cases be obtained from an irreplaceable battery cell. Therefore the Energy Discharge Rate (EDR) of the sensor node should be minimised to extend the WSN lifetime. These networks are envisaged to operate autonomously for months to years. The sensor nodes may also include some energy harvesting techniques to replenish energy levels, for example the increased size would allow solar cells to be attached to the sensor nodes. These factors will increase the weight of the sensor node and thus increase the EDR related to motion.

In order for a WSN to be effective, sensor nodes need to know where they are and this is achieved through localisation. When considering a static WSN the deployed nodes would only localise once during post-deployment. However, when mobility is incorporated the sensor nodes would need to localise frequently. The latter is also required for effective navigation. Localisation may be achieved through an onboard GPS (for outdoor environments) or a localisation protocol. GPS would be the most viable option for frequent localisation but it will increase the complexity and cost of the sensor node platform. Localisation protocols such as trilateration and triangulation using landmarks situated in the environment would incur communication overheads and the node would also need to cooperate with their neighbours.
Sensor node motion will more often than not change the network topology. This would effectively alter a sensor node's relative neighbours. This in turn explicitly affects the network layer (routing protocol) and data link layer (MAC protocol). Therefore reactive MAC protocols and routing protocols would possibly need to be adopted when implementing mobility. Alternatively routing tables and synchronisation sequences would need to be frequently updated. This would inherently require cooperation between the sensor nodes, and thus increase the communication overheads. This cooperation would generally be performed during the configuration phase. The configuration duty time would therefore depend on the mobility characteristics.

Mobile sensor nodes must also incorporate some navigational techniques to allow them to successfully move through the environment. The nodes should be capable of handling obstructions and uneven terrain. Therefore tactile sensors may need to be incorporated onto the sensor node platform. Table 1.1 summarises these design considerations by presenting the advantages and disadvantages of implementing a mobile WSN.

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adapt to application and environment</td>
<td>High energy overheads to physically reposition sensor nodes - reduced lifetime</td>
</tr>
<tr>
<td>Adaptable coverage and connectivity</td>
<td>Frequent configuration overheads</td>
</tr>
<tr>
<td>Energy balancing</td>
<td>Increased hardware complexity</td>
</tr>
<tr>
<td>Reduced redundancy</td>
<td>Increased processing power (mobility decisions)</td>
</tr>
<tr>
<td>Possible energy harvesting</td>
<td>Possible exposure of nodes</td>
</tr>
<tr>
<td>Reduced node deployment (recyclable)</td>
<td>High initial purchase costs</td>
</tr>
</tbody>
</table>

Motion within WSNs would be governed under a specified mobility strategy. This would implicitly define the mobility objectives such as enhancing or maintaining the network performance. Sensor node motion characteristics such as timing, speed, and direction would be explicitly coordinated by the mobility algorithm. As such, efficient and effective mobility strategies and algorithms need to be generated to minimise the disadvantages and maximise the benefits associated with adopting a mobile WSN. The mobility algorithms may adopt a centralised or decentralised approach to motion.
A centralised algorithm would generally run on a commander node (essentially assumed to be a mobile base station). The commander node would gather information about the network through monitoring data packets and communication. The information gathered would feed into the mobility algorithm which would return the desired motion characteristics for individual nodes or the entire network. The commander node would then broadcast these motion control signals to the relative sensor nodes. Sensor nodes would incur additional communication overheads when receiving these control signals. In addition the complexity of the commander node(s) would also increase due to the additional processing resources required to store and process sensor node information.

Under a decentralised approach the mobility algorithm would be local to each sensor node. The node would generally base decisions regarding motion on local information (direct neighbourhood information). Storing and gathering information about the entire network would carry large communication overheads. However local information does not provide a comprehensive knowledge of the rest of the network. Therefore this adds a level of uncertainty which must be taken into account to ensure effective mobility. The decentralised algorithm must be robust and able to handle uncertainty whilst ensuring the mobility objectives are achieved. Artificial intelligence techniques may need to be adopted to handle the uncertainties and determine the optimum mobility decisions.

1.3 Applications

Application development for WSNs include military, environmental monitoring, health monitoring, home automation, space exploration, chemical processing, and disaster relief, among many other commercial and industrial areas.

An example of a WSN was featured in the action thriller movie Twister (1996) [18]. The film featured two rival teams of scientists that wanted to earn their place in meteorological history by launching their equipment packs (Dorothy) inside a twister to transmit valuable data about tornado behaviour. The equipment pack consisted of a WSN which was sucked up into the tornado (deployment) and collected meteorological data. The latter was transmitted back to a remote terminal, situated at a safe distance from where the tornado landed. The film depicts a situation where the WSN needed
robust sensor nodes with a relatively short life span (a few hours by opposition to months or years). Interestingly the twister provided the means for mobility, thus removing one level of difficulty.

The sensor network may consist of a variety of sensor types with varying modalities. These sensors could include seismic transducers, low sampling rate magnetic, thermal, visual, infrared, acoustic, and radar, enabling the following variables and conditions to be monitored:

- Temperature,
- Humidity,
- Vehicular movement,
- Lighting conditions,
- Pressure,
- Soil makeup,
- Noise levels,
- The presence or absence of certain kinds of objects,
- Mechanical stress levels on attached objects,
- The current characteristics such as speed, direction, and size of an object.

The following sections discuss examples of WSN applications for military, environment, health, home and commercial areas.

1.3.1 Military Applications

The Military were the first to foresee the potential in deploying WSNs as an integral part of command, control, communication, computing, intelligence, surveillance, reconnaissance and targeting. Some of the military applications include the following:

*Battlefield surveillance* - the WSN could be deployed rapidly to observe the activities of opposing force, to allow attack and possibly retreat strategy to be formulated. Monitoring friendly forces, equipment, resources, and ammunition are also vital to military operation. These networks and more specifically the sensor nodes could be located on vehicles, weapons, and possibly around friendly forces to form a WSN. The commanders and leaders would then have the capability of monitoring the overall status of the military command and personnel.
Battle damage assessment - nodes could be deployed to monitor the effectiveness of an attack on opposing infrastructure. Therefore commanders would know whether additional attacks are required and also precise locations, which would assist in minimizing unnecessary destruction. This could also lead to some form of target discovery and tracking.

Reconnaissance of opposing forces and terrains - these networks could be deployed in critical terrains, and some valuable, detailed and timely intelligence about the opposing forces and environmental terrain could be collected for evaluation and analysis of opposing force’s military status.

WSNs could also be deployed in the case of chemical and biological warfare. These networks would have the capability of being deployed around the contaminated region (ground zero), to produce important warning signs and post fallout analysis of the region. Information about the casualty numbers and exclusions zones could also be generated and analyzed throughout the attack period; also analysis of weather and propagation patterns would allow predictions to be made about future exclusion regions.

1.3.2 Environmental Applications

WSNs have the potential to collect environmental information, such as tracking the movement of birds or small animals or insects (for example, bird observation on Great Duck Island Mainwaring et al [19], Maine, USA). Applications could also include monitoring environmental phenomena such as the pollution, chemical and biological effects on crops and livestock; marine, soil and atmospheric contexts. For example ocean water monitoring ARGO project [20] uses a sensor network to observe the temperature, salinity, and current profile of the upper ocean. Other possible applications include forest fire detection where nodes could be deployed over high risk regions to monitor possible changes in temperature. They could also assist fire fighting teams to extinguish the blaze and prevent it spreading.

Flood detection is another prime example of WSN use. The ALERT system [21] deployed in the US is a system which monitors rainfall, water level, and weather via sensors which supply information to a centralised database system for flood warnings and predictions. Precision agriculture is also a foreseen application where wireless sensors would collectively monitor the pesticides level in drinking water, the level of
soil erosion, and level of air pollution in real time. Other applications that would fall into this category include collecting information for meteorological or geophysical research, bio-complexity mapping of the environment and planetary explorations.

1.3.3 Health and Home Applications

Health and home have also been foreseen to benefit from the concept of WSNs and bring many application possibilities. Some of these applications include providing interfaces and support to individuals with disabilities, integrated patient monitoring within the hospital or for out-patients. These networks would have the potential to provide diagnostics, drug administration, monitoring of patient and doctor movements around hospitals.

Tele-monitoring of human physiological processes status, the network would allow physiological data to be collected over a long period of time, and would provide a significant benefit to medical exploration. A “Health Smart Home” has been designed at the Faculty of Medicine in Grenoble France to validate the feasibility of such applications see [22]. The project proposes a smart fall sensor system, which monitors the behaviour of elderly people and mentally disabled patients and identifies periods of distress, for example determine if a patient has fallen. The sensor node size would enable them to be integrated into a patience lifestyle whilst giving them their independence and enabling doctors to monitor their status.

Home automation would allow the end user / home owner to manage home devices locally and remotely more easily. Sensor nodes could be buried into vacuum cleaners, ovens, security systems, refrigerators and DVD recorders, allowing them to interact with each other and the user via the internet or satellites. For example users could monitor the contents of their refrigerators at work before going shopping, alternatively switch appliances on, such as the oven and hot water cylinder before they arrive home.

1.3.4 Other Commercial Applications

WSN have many potential commercial applications. These could include monitoring material fatigue, managing inventory, quality control, robot control and guidance in automatic manufacturing environments and monitoring risk processes in chemical
plants. These networks could be embedded into plant machinery to evaluate possible
time to failure. The system as a whole could source replacement components and
service engineers before the plants starts to malfunction, thus reducing the possibility of
downtime.

WSNs can be configured for security applications, where sensor nodes would act as
intrusion detectors. The latter could be deployed at ease without any wiring
infrastructure, which would be appealing for fast and effective coverage of an area.
This could also include monitoring car thefts with the nodes being deployed to detect
and identify threats within a geographical region, thus informing car owners to take
extra precautions. Tracking vehicles once they have been stolen would also be possible
under this concept.

Building management system applications, where the commercial building would be
completely autonomous in terms of controlling temperature and humidity and
monitoring security. Large quantities of nodes which collect sensing data would be
deployed within the building. The user (building management system) would use the
data collected to control plant and inform personnel of maintenance requirements. For
example temperature sensing data would be used to control heating and cooling plant.

In terms of the utility industries, nodes could be deployed to collect meter reading for
gas, water, and electricity. This could be extended to leak detection in the case of gas
and water. Nodes could monitor and detect abnormal changes in pressure and cracks,
thus isolating possible leakage points within the service infrastructure. Leaks are a big
problem within service distribution and supply companies are normally aware of their
presence, but pinpointing their precise location may incur high costs. Sensor nodes
could be deployed at pipe intersections and outlets or nodes could flow through the pipe
dine and measure the pipes integrity (the services may need to be isolated for safety
purposes).

1.4 Objective and Overview of the Thesis
The aim of the research presented herein was to investigate mobile WSNs and create
new mobility strategies and algorithms to improve their performance. The hypothesis
put forward is that incorporating Artificial Intelligence (AI) at the level of the sensor
nodes will improve the mobile WSN's performance and adaptability, in the face of uncertainty associated with using minimal information (typically from neighbours).

The research applies Bayesian network based techniques to enable sensor nodes to reason about the impact of their motion characteristics and consequently make 'intelligent' motion decisions. This provides a decentralised mechanism where sensor nodes generate performance predictions and motion characteristics based on local neighbour information at the cost of minimal additional communication overheads. The work has resulted in algorithms which have been implemented and validated through simulation and comparison with existing techniques wherever possible. A literature survey on WSNs and mobile WSNs including relevant mobility strategies is presented in Chapter 2. Chapter 3 summarises the mobile WSN modelling parameters and presents a new WSN simulator. The simulator was developed using Matlab™ and provides a tool for evaluating the performance of mobile WSNs. The performance criteria considered herein include coverage, connectivity, and network lifetime.

Mechanisms are created to support the performance evaluation of the new mobility strategies. Among these is a new Coverage Against Time performance measure, presented in Chapter 4, which evaluates the coverage with respect to time. Traditionally coverage is measured at an instantaneous point in time and returns the fraction of the total region collectively covered by the sensor node. The new Coverage Against Time measure evaluates the accumulation of coverage with respect to time and also the coverage service period for points within the region of interest.

A biologically inspired mobility strategy is presented in Chapter 5 which emulates the behaviour of herbivores grazing pastures. The geographical region of interest for a WSN becomes analogous to pasture, and the information collected by the nodes analogous to forage. A smaller number of mobile nodes are deployed instead of a large number of static nodes to cover a region of interest. Under this scenario coverage is achieved by the sensor nodes collectively migrating around the region of interest and coverage being achieved over time.

Three mobility algorithms are presented which coordinate the migration of the WSN. A fixed path approach was adopted with a pre-specified roaming for each sensor node. A new centralised reference point based algorithm called Grazing Reference Point Group Mobility (GRPGM) is presented. Under the latter the WSN motion is coordinated by a
commander node broadcasting reference points to all the sensor nodes. The reference points are derived from reasoning about WSN performance via information received from all the sensor nodes.

The third mobility algorithm presented is called Bayesian Network GRAZing (BNGRAZ) and is a new decentralised approach to grazing that implement a Bayesian network for motion decisions. The Bayesian network allows sensor nodes to predict the probability of maintaining or increasing coverage and connectivity given that they move in a particular direction. These predictions are based on the position of local neighbours and information gathered through forwarding data packets. These predictions enable a sensor node to reason about its optimum motion characteristics with a view to achieving the desired performance.

In addition, the fixed path, GRPGM and BNGRAZ algorithms are compared to the Random Way-Point (RWP) mobility model, discussed in section 2.4.1. RWP entity based mobility model was proposed to model movement of individuals that form an ad-hoc network.

The decentralised approach to grazing required the creation of a new Coverage Approximation (CA) algorithm, discussed in section 5.4.3.2. The CA algorithm enables a sensor node to approximate the WSN coverage from local neighbourhood information and data packets routed through the sensor node. The information is used to maintain a coverage map of the region of interest enabling a sensor node to estimate which areas have not been covered by the WSN. The coverage map forms part of the evidence fed into the Bayesian network to predict the probability of an increase in coverage (discovering areas which have not been covered by the WSN).

A new decentralised Bayesian network based coverage maintenance mobility algorithm (called BayesMob) is presented in Chapter 6 that imparts self-healing properties to the WSN. Sensor nodes may randomly fail due to energy exhaustion, malicious destruction and or malfunction ultimately affecting the network’s performance (in terms of coverage, connectivity and lifetime). BayesMob predicts the probability of an increase in coverage given that a sensor node moves in a particular direction. These predictions enable the sensor node to coordinate its mobility characteristic to increase coverage and thus enabling the network to recover from the failures.

Chapter 7 presents conclusions and recommendations for future work.
Chapter 2
LITERATURE REVIEW

2.1 Introduction
This chapter presents a summary of previously published work forming a background for the work presented here; namely, the creation and investigation of the effectiveness of various mobility strategies and algorithms applied to mobile WSNs for maintaining and enhancing the network performance.

More specifically the literature review covers the following areas:

- **Network simulators** – capable of simulating WSNs.
- **WSN performance** – Quality of Service (QoS) requirements and performance aspects. The performance aspects considered include coverage, redundancy, communication and operational lifetime (sensor nodes’ Energy Discharge Rate).
- **Mobile WSNs** – in which various motion coordination strategies and algorithms are reviewed.
- **Artificial Intelligence (AI)** – general definition of intelligence and AI, outline of trends in the engineering of intelligence, the application of AI in WSN, and Bayesian networks.

2.2 WSN Simulators
The need for and usefulness of simulation and simulation environments needs little introduction. When designing complex and expensive systems it is always beneficial to try solutions before implementing them. WSNs and their operation are affected by enough parameters to warrant the use of simulations to investigate the vast solution space associated with anyone problem related to them. A survey of the literature revealed that there are several simulators available for analysing the performance of wired and wireless networks. However because of many constraints imposed on sensor networks, such as energy limitations, decentralised collaboration and fault tolerance,
algorithms for sensor networks tend to be quite complex. Therefore simulators proposed for WSNs need to address issues unfamiliar to traditional network simulators. The following discusses simulators developed specifically for WSN, and simulator that have incorporated features that may help to address the simulation of WSN.

2.2.1 SENSE_2.0

SENSE [23] is a WSN simulator developed by Gilbert (Gang) Chen at the Centre of Pervasive Computing and Networking at Rensselaer Polytechnic Institute. SENSE was developed specifically for the simulation of WSNs. SENSE is built on top of COST, a discrete event simulation tool. COST adopts a component orientated approach which enables COST models to be highly reusable. COST is written in C++ with a heavy use of templates. A COST model built for one simulation can be easily embedded into other simulations.

SENSE features a fast sequential simulation engine and an efficient memory management library designed exclusively for wireless networks. The majority of COST and SENSE has been rewritten using CompC++, resulting in clearer and shortened code, reduced compilation time and improved simulation speed. CompC++ is a language extension to standard C++ that introduces the fifth programming paradigm, component-orientated programming.

The dominant features of SENSE are its ability to address factors such as extensibility, reusability, and scalability, and to take into account the needs of different users. The drawback of the simulator is that it runs via a shell and does not include a graphical user interface. The components associated with the simulation engine are limited in number, and therefore users need a good understanding of CompC++ to develop simulation components specific to their own applications. Finally the simulator is not well supported and simulation examples are limited.

2.2.2 The Network Simulator (Ns2)

The Network simulator (Ns2) [24] is a discrete event simulator targeted at networking research. Ns2 provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. Ns2 began as a variant
of the REAL network simulator [25] in 1989. In 1995 Ns2 development was supported by DARPA through the VINT project [26] at LBL, Xerox PARC, UCB, and USC/ISI.

Ns2 is perhaps the most widely used network simulator. It has been extended to include some basic facilities to simulate WSN. However, there are no dominant protocols or algorithms and it is unlikely there will be any, because a WSN is often tailored for a particular application with specific features, and it is unlikely that a single algorithm can always be the optimal one under changing circumstances. In addition, one of the problems of Ns2 is its object-oriented design approach which introduces much unnecessary interdependency between modules. Such interdependency sometimes makes the addition of new protocol models extremely difficult, only mastered by those who have intimate familiarity with the simulator.

Other publicly available network simulators, such as JavaSim [27], and SSFNet [28] attempt to address problems that were left unsolved by Ns2. JavaSim developers realised the drawback of object-oriented design and tried to solve this problem by building a component-orientated architecture. However, they chose Java as the simulation language inevitably sacrificing the efficiency of the simulation. SSFNet designers were more concerned about parallel simulation. These simulators are not superior to Ns2 in terms of design and extensibility.

2.2.3 Glomosim

Global Mobile Information Systems Simulation Library (Glomosim) [29] is a library-based sequential and parallel simulator for wireless networks, developed by the University of California Los Angeles (UCLA). It was developed using the parallel discrete-event simulation capability provided by Parsec [30], a C-based parallel simulation language. Glomosim adopts a layer approach which is similar to the OSI seven layer network architecture. Standard Application Programming Interfaces (APIs) are used between the different simulator’s layers, which allow the rapid integration of models developed at different layers by alternative users/researchers. The user/application designers are able to program new protocols, modules and mobility models and add them to the library. This makes Glomosim extensible and possibly suitable for simulating mobile WSNs.
2.3 QoS and Performance Evaluation

The performance of any system/process would generally be evaluated under the umbrella term Quality of Service (QoS) which refers to the quality as perceived by the user/application. In the networking community the QoS is generally not related to a single performance measure, but a collection of metrics which evaluate the overall network performance. WSNs are no exception and the QoS performance measures are generally application specific thus the relevance of different measures will vary across applications. Kansal et al [13] suggested that the performance of a WSN may be best judged by the quality of application specific information returned. Chen et al [31] analysed the QoS support in WSNs, and outline the challenges in providing the specified QoS criteria. A generic QoS criterion for all applications is difficult to achieve. However, the following aspects should always be considered when attempting to formulate QoS criteria: coverage, connectivity, lifetime, real-time constraints, robustness, eavesdropping, data reliability, capacity, scalability and redundancy. The first three aspects (coverage, connectivity and lifetime) were used throughout this work to evaluate mobile WSN and are discussed in sections 2.3.1 to 2.3.3. For the sake of completeness, the following gives a brief discussion of the other performance aspects.

Real-time constraints - are important in some applications where not only the data relating to the phenomena is important, but also the time in which the data is translated by the user. This implies that latency between sensing the phenomena of interest and the data being received and the interpretation at the user interface should be minimal.

Robustness - defines the ability of the network to remain operational even if certain well-defined failures occur. Failure could imply single or multiple nodes in the network becoming energy exhausted, or nodes becoming disconnected due to malicious destruction or malfunction. Whatever the cause of failure, the remaining nodes should operate at some defined level to satisfy the application criteria. Tamper resistance also falls into the robustness category. It defines the ability of the network to remain operational even when single or multiple nodes are subject to deliberate attacks from external agents attempting to prevent network operation.

Eavesdropping – defines the ability of the network to prevent external entities from extracting and altering data transmitted through the network. This would generally be
related to the WSN security features which are generally application specific. Hwang et al [32] investigated the inherent tradeoffs involved between energy, memory, and security robustness in WSNs. The authors defined a metric (security leakage factor) to quantify the security robustness in the system. Eavesdropping may be prevented by making the WSN unobtrusive by incorporating stealth capabilities. This would enable a sensor node to operate in hostile environments without being detected by external entities.

**Data reliability** - is also an important factor in the performance of WSN. The base station/user should have the ability (via communication protocols) to distinguish between valid and invalid data. The data received from the sensor nodes should be accurate with minimal discrepancies. Packet loss can be generalised under this measure, and would typically give an indication of received packets against transmitted ones. Zhao et al [33] analysed the packet delivery performance in WSNs, from the physical and MAC layer perspective. The authors conducted experiments to evaluate the packet delivery rate for three possible application environments: an indoor office building, a habitat with moderate foliage, and an open car park. The findings quantify the prevalence of gray areas within the communication range of sensor radios and indicate significant asymmetry in realistic environments. The limited network Bandwidth must also be considered due to sensor nodes communicating over a wireless medium. This relates to the WSN data throughput capabilities.

**Capacity** - directly relates to the sensor nodes data throughput. As a whole a network will have a limited throughput capability due to the wireless nature of the network. Capacity would directly relate to the Medium Access scheme adopted. Therefore in order to maximise the throughput capabilities of the networks the sensor nodes should have the ability to successfully transmit data packets over the medium, in a way that minimises latency and collision. Gupta et al [34] analysed the throughput capacity of wireless networks and outlined the required conditions for successfully transmitting a data packet over a number of sub-channels. As a result an upper bound on the throughput capacity of WSNs was defined.

**Scalability** – indicates the ability of the WSN to gracefully handle either an increase in work or network size, Chen et al [31]. For example, it can refer to the capability of the WSN to increase total throughput under an increased load when resources (typically hardware) are added. Scalability, as a property of systems, is generally difficult to
define and in any particular case it is necessary to define the specific requirements for scalability on those dimensions which are deemed important. If a WSN’s performance improves after adding sensor nodes, proportionally to the capacity added, is said to be a scalable system. Alternatively if the network fails it does not scale.

**Redundancy** - WSNs are characterised by high redundancy in terms of the number of sensor nodes deployed and sensor data. However, while the redundancy does help loosen the reliability/robustness requirement of data delivery, it unnecessarily spends much precious energy. Mobility, data fusion, or data aggregation, is a solution to maintain robustness by decreasing the redundancy, but these mechanisms also introduce latency and can add additional strain on resources.

The above only gives a general overview of possible application specific QoS performance criteria. Three performance measures which are applicable to all applications are sensing coverage, connectivity and network lifetime. Due to the applicability of these measures they have been used throughout this work to evaluate the performance of mobile WSN. The following sections give a detailed discussion of these measures.

### 2.3.1 Coverage

Coverage is related to the collective sensing regions of the sensor nodes that form the WSN. When sensor nodes cooperate to form a WSN their individual sensing regions are combined to form the WSN’s sensing coverage region. Meguerdichian *et al* [35] defined coverage as the fraction of the total intended area actually covered by the WSN. Therefore coverage is traditionally calculated by determining the percentage of the geographical region of interest covered by the WSN’s sensing region.

Coverage may also be evaluated by considering the degree of coverage which defines how well the region of interest is covered. The actual degree of coverage would be determined by the sensing accuracy and redundancy. The former defines the reliability of the sensing data received by the user/application software. The latter (redundancy) relates to multiple sensor nodes (their sensing regions) with the same modalities covering the same physical point. Coverage may vary across the region of interest for example sensor nodes may be deployed more densely at points where frequent variable changes or activities are expected and sparse where minimal changes are anticipated.
The degree of coverage may also influence the information processing algorithms. High degrees of coverage would provide a robust WSN and redundancy may be exploited to extend the network lifetime by switching redundant sensor nodes to power saving sleep modes. The latter will invariably depend on the application criteria and associated cost implications.

Kansal et al [13] measured coverage by evaluating the area which is visible to the sensor nodes compared to the free area left in the square after the area occupied by the obstacles themselves is subtracted.

Depending on the deployment scheme, there would generally be more nodes available than required for simple coverage of the space. Therefore coverage is achieved by overpopulating the geographical region of interest. Static WSNs are a prime example of this, where density becomes a design dimension. However WSNs that incorporate some level of mobility would relieve the need for high density. Also nodes would be dynamic and could adapt to application and environmental issues.

Wang et al [36] presented the design and analysis of novel protocols that can dynamically configure a WSN to achieve guaranteed degrees of coverage and connectivity. The work differed from existing connectivity or coverage protocols in the following ways: they present a Coverage Configuration Protocol (CCP) that can provide different degrees of coverage requested by the application criterion. A geometric analysis of the relationship between coverage and connectivity is also presented.

Under the CCP nodes can be in one of three states: sleep, listen or active. When deployed all nodes are in the active state. If the deployed region contains redundant nodes, the latter will enter the sleep state, without affecting the degree of coverage. The sleeping node would periodically enter a listen state to evaluate its eligibility to enter the active state. Jiang et al [37] presented a similar density control algorithm by coordinating the sleep and active states of the sensor nodes.

2.3.2 Connectivity

Connectivity relates to the WSN topology (physical locations of sensor nodes) and facilitates the communication between neighbouring nodes and the successful
transmission of sensing data to the task manager via the base station (possibly commander node). A sensor node is connected if it can successfully communicate with the base station possibly via multi-hop routing. Multi-hop routing is necessary because sensor nodes have a limited communication range and therefore connectivity may be achieved by connections with neighbouring sensor nodes. Connectivity is represented as a fraction of connected sensor nodes against the total deployed number. The definition of connectivity implies that a sensor node is capable of communicating with all other connected nodes. A further investigation of connectivity and literature related to mobile WSNs and connectivity is given Section 2.4.3.4.

2.3.3 Lifetime

Critical to any WSN applications is the actual lifetime. With many foreseen applications, nodes should have the ability to be deployed into a remote area and operate in an unattended manner for months or years. However the limiting factor for network lifetime is the energy storage facilities of the nodes. Nodes that populate the network are generally powered by individual batteries located on the nodes structure. As the nodes' physical size reduces to meet the applications' QoS requirements so does the battery size and thus energy storage capacity. Each node should therefore be designed to manage its individual energy, in order to prolong lifetime. Another important consideration is energy balancing. Nodes should operate in a way that balances the Energy Discharge Rate (EDR) evenly between all nodes. This is to prevent uneven energy discharge resulting in coverage holes and connectivity loss due to early energy exhaustion.

The most significant factor in determining lifetime is the radio power consumption and motion power overheads. This power consumption can be decreased by adjusting the communication output power, transmission duty cycle, and mobility characteristics. However, these limitations can jeopardise other QoS measures, such as connectivity, packet loss and capacity.

In many deployments it is not the average node lifetime that is important, but rather the minimum lifetime. Lifetime predictions can be based on a number of scenarios, for instance, the lifetime of any node in the network. The time until a specified percentage of the nodes become energy exhausted (for example the network would satisfy the
application criteria until 10% of the nodes fail) could be the criterion to satisfy. Alternatively the lifetime criterion could be based on the time until collective node failure causes the network operation to fail. Given the above criteria the EDR of the sensor nodes should be balanced such that all nodes fail at approximately the same time.

Bhardwaj et al [38] derived a bound on the maximum network lifetime that any collaborative protocol can ever hope to achieve. Zhang et al [39] also explored the fundamental limits of sensor network lifetime that all density algorithms can possibly achieve. The work derived the asymptotic lower bound on node density (k-coverage) required to ensure full coverage for the duration of $k$ times the lifetime of a single sensor ($k$ sensor nodes cover each point in the region). In addition two upper bounds on lifetime in a finite region with a finite density of nodes are defined. Firstly, an upper bound of lifetime for a special family of algorithms in which the entire region is completely covered initially, and k-coverage is gradually reduced until lifetime drops below a specified threshold. Secondly, an upper bound of lifetime that holds universally for any possible density algorithm.

Carlos et al [40] presented work on generating the lower bounds on the energy cost of sensing nodes communicating, with a view to collecting information required for configuration protocols and sensor node coordination. The work focused on the energy cost of exchanging information to satisfy the cooperation and the configuration protocols and ignored the energy cost of moving agents (static scenarios only). The authors proposed a model of the energy overhead for nodes exchanging information in the WSN.

The network lifetime in the case of static WSNs would be directly influenced by the transmission scheme, network density, and transceiver parameters (Optimal transmission power, and cell size). Xue et al [41] investigated the lifetime of a WSN which is assumed to be static, with $n$ randomly distributed sensor nodes communicating to a base station. The authors stated that the lifetime can be extended by incorporating motion capabilities. The latter enables the WSN to balance the sensor nodes’ EDR. A location aware hybrid transmission scheme that balances the network energy consumption is presented. Under this scheme the energy balancing is achieved by distributing the relaying overhead related to communicating with the base station over $n$ possible variations in relaying nodes.
A widely employed energy saving technique is to place nodes into hibernation during periods of reduced operation. Chiasserini et al [42] created a Markov model of a WSN where nodes enter into sleep/hibernation periods. The work investigated the WSN performance while adopting this model, and evaluated energy consumption, network capacity (throughput), and data delivery delay. The model enables the user to investigate the trade-offs that exist between these performance metrics and the sensors entering sleep periods.

Xu et al [43], [44] proposed two protocols to coordinate the sensor nodes sleep cycles. The protocols identify redundant sensor nodes and turn off their radio to reduce communication overheads (energy associated with constantly listening to the communication medium). The proposed protocols are Geographical Adaptive Fidelity (GAF) and Cluster-based Energy Conservation (CEC). The GAF self-configures redundant nodes into small groups based on their locations and uses decentralised distributed algorithms to control sensor node sleep duty cycles to extend network operational lifetime. The nodes are either in one of three states: sleeping, discovery, active. In the discovery state the nodes evaluate their neighbour's state and determine its redundancy status. If redundant it enters a sleeping state, otherwise it enters the active state. The node then remains in that state for a specified time period, from which it re-enters the discovery state and repeats the cycle.

The CEC follows the same principle as GAF, but eliminates its dependency on location information and uniform radio propagation. This protocol becomes applicable in circumstances where location information is not available. CEC organises nodes into overlapping clusters that are interconnected.

Simulations show that GAF can substantially conserve energy (40% to 60% less energy than an unmodified ad-hoc routing protocol), which allows the network operational lifetime to increase in proportion to node density. The CEC protocol shows similar traits although the protocol eliminates the dependency on global location information and its assumption about radio range.
2.4 Mobile WSNs

The previous section has given a general overview of WSN performance aspects. The mobility algorithms developed as part of this work attempt to improve the coverage, connectivity and lifetime of mobile WSNs. This section reviews literature related to mobile WSNs and mobility algorithms aimed at improving coverage, connectivity and lifetime.

2.4.1 Mobility Models

A concise review of current mobility models available for the simulation of ad-hoc networks is presented by Camp et al [45]. The authors described several mobility models that represent mobile nodes whose movements are independent of other nodes that form the network (i.e. entity mobility models). They also outlined an alternative mobility strategy where node movements are dependent on all nodes that form the network (i.e. group mobility models). These group mobility models can be used to model a group of sensor nodes that move through the region of interest. For example, sensor nodes attached to a convoy of trucks on the highway or a platoon of marching soldiers. These entity and group mobility models can be used to represent random and predictable mobility within WSNs, such as nodes being attached to external mobile entities.

The authors discuss the following entity mobility models:

- Random Walk Mobility Model, which is based on random destination (assumed to be uniformly distributed) in the region of interest and a speed which is uniformly distributed between \([\text{minspeed}, \text{maxspeed}]\).

- Random WayPoint (RWP) Mobility Model, Johnson et al [46], which is similar to the Random Walk model, but RWP includes a pause period between nodes changing direction and speed.

- Random Direction Mobility Model, which forces nodes to travel to the edge of the geographical region before nodes change direction and speed.

- Gauss-Markov Mobility Model, which uses a tuning parameter to vary the degree of randomness when selecting the node’s speed and direction.
• Probabilistic version of the Random Walk Mobility Model, which utilises a set of probabilities to determine the next position of the node. This produces probabilistic, rather than purely random movements, which may yield more realistic behaviour. This is loosely based on the way people complete their daily tasks as they tend to continue moving in a semi-constant forward direction. Rarely do we suddenly turn around and retrace our steps, or rely on random movements to reach our target location.

As well as the entity mobility models, the following group mobility models are also outlined:

• Exponential Correlated Random Mobility Model, which uses motion functions, to determine the individual nodes direction and speed.

• Column Mobility Model, which forces nodes to form a line when uniformly moving forward in a particular direction.

• Nomadic Community Mobility Model, which forces nodes to move together from one location to another.

• Pursue Mobility Model, which forces nodes to follow a given target. This could be an external entity or even a node which is classified as the leading node.

• Reference Point Group Mobility (RPGM) Model, which is a group mobility model that bases the nodes’ movements on the logical centre of a reference point.

2.4.2 Accommodating the Physical Relocation of Sensor Nodes Associated with Mobile WSNs

Sensor node cooperation is part of the network configuration which is vital to the efficient operation of mobile WSNs. During configuration, sensor nodes cooperate with neighbours to achieve a number of essential tasks. Among these are synchronisation, neighbour discovery, routing table generation, localisation, navigation, and possibly MAC configuration. Jung et al [47] described experiments conducted in evaluating cooperation using autonomous mobile robots to perform a cleaning task. The experiments were designed to evaluate the effect of cooperation on the task criterion.

32
The experiments range from using emergent cooperation with no communication to explicit cooperation and communication.

Pham et al [48] investigated the mobility of WSNs from a Medium Access Control (MAC) perspective. They present a new adaptive mobility-aware MAC (MS-MAC) protocol. The protocol uses any changes in received signal level as an indication of mobility and triggers when necessary, the mobility handling mechanism. The mobility handling mechanism generates active zones around the mobile nodes and prolongs the configuration period for node synchronisation. The MS-MAC works similarly to the S-MAC (Ye et al [49]) which conserves energy when nodes are stationary. However, the medium access scheme may also switch to work similarly to the IEEE 802.11 [50] for extremely mobile ad-hoc scenarios. S-MAC is basically a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) MAC protocol, based on IEEE 802.11. MS-MAC introduces periodic coordinated sleep/wakeup duty cycles, thus extending the battery lifetime of the sensor nodes.

An additional consideration associated with mobile WSN is localisation errors. Son et al [51] studied the effect of inaccurate location information caused by node mobility under a rich set of motion scenarios and models. They identified two main problem areas named LLNK and LOOP; that are caused by mobility-induced location errors. The LLNK problem relates to the link connection problem with neighbouring nodes, and the LOOP problem is related to inaccurate location information of destination nodes caused by their mobility.

Samar et al [52] investigated the communication links' behaviour for a node in a random mobile environment. The authors generated analytical expressions characterising the link properties related to creation, lifetime and expiration. These link properties can be used for the efficient design of routing and MAC protocols, or to evaluate existing protocols, also when designing mobile models for the evaluation of routing and MAC protocols, or performance aspects of the WSN.

Traynor et al [53] presented a mobility management protocol for group migration of sensor nodes. The authors suggest that in order for WSNs to provide the specified coverage or track moving targets, mobility should be incorporated into the network. As nodes migrate across the geographical region they would encounter access points or data sinks, where they would perform a number of handshaking exchanges that should
allow data to be delivered nearly seamlessly. They suggested that the time associated with re-establishing connectivity, the frequency of dropped packets and the contention for the air interference, all increase significantly for mobile groups. The proposed protocol reduces the above by allowing a single node to perform handoffs on behalf of all group members. This “gateway” node eliminates the need for multiple handoff messages by obscuring group membership to external parties.

2.4.3 Controlled Mobility

As previously stated, controlled mobility relates to the controlled and coordinated motion of network elements. Traditional systems have considered mobility as an extra overhead to which the network must adapt, possibly at the expense of a loss in performance. Kansal et al [10] envisioned the use of controlled mobility as a new design dimension to enhance sustainability in WSNs. The authors proposed a new method where controlled mobility is used by the network itself to enhance functionality. They identified the advantages of controlled mobility and outlined two research challenges. The first is designing sensor network protocols which exploit mobile components effectively and the second is solving navigational problems associated with the mobile elements (essentially when and where nodes should physically relocate). Also they discussed a case study where controlled mobility is used namely, Networked InfoMechanical Systems. The mobility strategies and algorithms created as part of the work presented herein concentrate on the second research challenge.

Networked InfoMechanical Systems (NIMS) introduce a new actuation capability for embedded networked sensing. By exploiting a constrained actuation method based on rapidly deployable infrastructure, NIMS suspend a network of wireless mobile and fixed sensor nodes in three-dimensional space. This permits run-time adaptation with variable sensing locations, which enables the network to handle environmental influences such as the presence of obstacles. NIMS are based on the Infostation model proposed by Goodman et al [54] and Iacono et al [55]. In an Infostation model, users can connect to the network in the vicinity of ports (or Infostations), which are geographically distributed throughout the area of network coverage.

Pon et al [56], [57] described the NIMS architecture proposed for WSN, including the hardware platform and software architecture. The authors suggested that this platform
will address sensing uncertainty, adaptive sampling requirements, and operate in large three-dimensional environments with energy harvesting capabilities.

Rahimi et al [58] described the NIMS methodology that permits the intensive and extensive exploration of environments with mobile, robotic sensor systems, and state that the system has promising potential. They developed a Nested Stratified Sampling method for NIMS as a “divide and conquer” algorithm. Under the algorithm the complete surveillance area being observed, is stratified into a number of layers. Then the environmental variables are sampled in each stratum with a number of samples proportional to the area of the strata.

Qiu et al [59] proposed a novel sensor movement strategy in which a commander controls a cluster of mobile sensors to monitor a square area at a certain distance ahead of him in his direction of movement. The mobility strategy was proposed for battlefield surveillance where a commander requires accurate information about a specific region of the battlefield. As the speed and direction of the commander changes, the new positions of the sensor nodes are generated by the proposed movement control algorithm. The motion coordination is centralised where the commander explicitly controls the speed and direction of the sensor nodes. Therefore during relocation the commander is responsible for executing the motion algorithm and broadcasting the motion control signals to the sensor nodes. Adopting this movement strategy would therefore carry significant communication overheads. However the commander has explicit control over the network behaviour and essentially performance.

2.4.3.1 Mobile Base Station/Gateway/Sink

An alternative approach to implementing a mobile WSN is to assume that the sensor nodes are static whilst the base station or gateway is mobile. The following discussion relates to references which present strategies and algorithms for coordinating the base station or gateway motion and data communication for enhanced network performance in terms of energy, delay and throughput.

Akkaya et al [60] presented base station motion coordination strategies and discussed techniques which ensure seamless data communication. The authors considered both constrained and unconstrained network traffic. When the network traffic is unconstrained the base station motion is coordinated to reduce its neighbours’ (sensor
nodes) energy overheads associated with forwarding data packets. Alternatively, when the network traffic is delay-constrained the motion coordination algorithm periodically checks the deadline miss rate for real-time packets and triggers a relocation stimulus for the base station/gateway if the miss rate exceeds a certain threshold. The base station/gateway would effectively move closer to the sensor nodes which forward the largest number of real-time packets. Butler [61] considered a similar strategy by using mobile robots as gateways into a WSN.

Chakrabarti et al [62] proposed a data collection scheme which relies on predictable base station/gateway mobility. For example the base station(s) could be attached to public transport vehicles (buses, shuttles and trains) which periodically pass through the WSN and collect pollution data relating to a city centre. The mobile entities are referred to as mobile observers. As the mobile observers pass through the city the attached base station would collect pollution data from each sensor node/cluster, perform possible data aggregation and effectively forward sensing data to a remote user (maybe when it reaches the station). This approach reduces the communication energy overheads associated with forwarding the data packets through the network to the base station and thus increases the operational lifetime of the network. The approach assumes that all sensor nodes are static and therefore the largest energy overhead is communication.

The authors also presented a queuing formulation to model data collection by the mobile observer over the region of interest. This queuing formulation captures the uncertainty due to random placement of sensor nodes deployed across the region of interest. The authors showed that to achieve a predefined outage, defined as the fraction of nodes that fail to send their data, the predictability of the observer’s motion can lead to large power savings over a network with no mobility. Also they proposed a simple observer driven communication protocol. This protocol pulls data from the nodes by waking the nodes up when they are in close proximity.

Shah et al [63] investigated a similar concept but called the mobile observers, data MULEs (Mobile Ubiquitous LAN Extensions). The MULEs would pick up data from sensor nodes when they are in close proximity, buffer it, and drop off the data to wired access points. The work assumes that the WSN is sparsely deployed across a geographical region of interest. The main advantage of this approach is the reduction in communication overheads due to the nodes only having to transmit data over short distances, thus maximising the lifetime of the WSN. However the main disadvantage is
the latency of data transmission due to the data MULEs physically moving to collect the data. Therefore this concept would not be feasible for delay-constrained data traffic (real time applications).

Lu et al [64] proposed a similar approach to the mobile gateway concept except that they assumed that mobile users collect information from the WSN. For example, consider a WSN deployed across a geographical region of interest, the sensing data would be collected via a mobile user passing through the sensor field carrying a compatible PDA device. This approach carries several advantages and also adds additional constraints on the sensor node memory. The authors presented MobiQuery, a spatio-temporal query service that allows mobile users to periodically gather information from their surrounding environment. They also presented a novel just in time pre-fetching algorithm that enables MobiQuery to maintain robust spatio-temporal guarantees even when nodes operate under extremely low duty cycles.

2.4.3.2 Mobile Deployment Strategies

Deployment is the process of physically distributing the WSN (sensor nodes) across a geographical region of interest. The initial deployment configuration may implicitly govern the network’s performance. The deployment configuration could be optimised via generating a heterogeneous WSN, where a proportion of the sensor nodes contain motion capabilities. After the initial deployment configuration, the mobile sensor nodes would move from densely deployed regions to areas where sensors nodes are sparse. For example Wang et al [11] presented a protocol for optimising the coverage of a WSN during the deployment phase. The motion coordination is based on a bidding-protocol. The static sensor nodes detect coverage holes locally using Voronoi diagrams and bid for coverage from mobile sensor nodes. The mobile sensor nodes then choose a location with respect to the bids it receives which are based on the size of the coverage holes detected.

Wang et al [12] also presented a study on the design and evaluation of distributed self-deployment protocols for mobile sensors. Voronoi diagrams are used to discover coverage holes in the WSN, which are caused by non-uniform deployment. Three movement-assisted sensor deployment protocols are proposed, which calculate the target positions of nodes with a view to maximise performance. These protocols are
VEC (VECtor-based), VOR (VORonoi-based), and Minimax. The protocols are based on the principle of moving sensors from densely deployed regions to areas with sparse coverage.

The VEC algorithm is motivated by the attributes of electro-magnetic particles: when two magnetic particles are too close to each other, a repelling force pushes them apart. The sensor nodes are analogous to magnetic particles. Compared to the VEC, the VOR algorithm is a pull-based algorithm which pulls sensors to their local maximum coverage holes. In the VOR, if a sensor node detects the existence of a coverage hole, it will move towards its farthest Voronoi vertex, with a view to average the distance between all neighbours. A Voronoi vertex is an intersection of three or more segments, each equidistant from a pair of sensor nodes.

The Minimax algorithm is similar to the VOR, where sensor nodes repair coverage holes by moving closer to the farthest Voronoi vertex, but it does not move as far as the VOR to avoid situations where closest becomes the farthest vertex.

The three algorithms are evaluated under the following performance criteria: deployment quality and cost. The deployment quality is a measure of the sensor coverage and the time to reach this coverage value. Cost has two sub components. Firstly, the energy associated with physically repositioning the sensor nodes and secondly, the number of sensor nodes required to achieve the desired coverage. Simulations showed that the proposed algorithms require fewer sensor nodes than the random deployment of static nodes and that coverage is maximised within a short deployment period with minimum movements.

2.4.3.3 Controlled Mobility to Extend Lifetime

If sensor nodes are required to move then the right choice of mobility strategy will extend the WSN lifetime. Sensor nodes mobility may be coordinated to search for energy and recharge and possibly deliver energy to immobile energy exhausted nodes. The latter is an energy harvesting technique and was explored in Rahimi et al [65] whilst generating a simple analytical framework. The authors proposed that energy harvesting is a possible solution to uneven sensor node EDRs, caused by varying energy overheads related to forwarding data packets and mobility. Current approaches
to energy management mainly focus on low power architecture and low power network design at different communication layers. These include:

- Low power hardware architectures.
- Low power software techniques.
- Limiting transmission range and power control at the physical layer to bound device consumption, Ramanathan et al [66].
- Low power MAC mainly by increasing MAC layer sleep time of the nodes, Ye et al [67].
- Dynamic configuration of nodes with extra deployment in geographical regions for sleep cycles in higher time granularity, Cerpa et al [68].
- Geographic and power aware routing to bound network traffic, Yu et al [69].
- Data Aggregation to increase the good throughput of the network and suppress unnecessary data traffic, Madden et al [70] and Intanagonwiwat et al [71].

The energy harvesting techniques explored by Rahimi et al [65] would inherently carry overheads, in terms of communication and physically moving the sensor node to collect and deliver energy. For example, delivering the energy harvested from solar cells would carry a larger overhead than collection, unless the sensor nodes move in response to the sun position (move out of shadows).

The majority of the energy aware routing protocols proposed for WSNs adopted a multi-hop routing strategy, in order to minimise the total transmission power. Problems may arise with this strategy as nodes located in the vicinity of the base station or sink carry the largest energy overhead due to forwarding data packets from the periphery of the WSN, to the base station. In addition, the focus on optimising the transmission energy tends to increase the level of data packet relaying and thus increases the data transmission latency, which can be critical to real-time applications.

Rao et al [16] presented an alternative distributed mobility algorithm aimed at extending the network lifetime. The mechanism was inspired by observing the natural grouping behaviour of Emperor penguin communities in the Antarctic regions. During the winter season a group of Emperor penguins form a closely huddled group, with a
view of improving the collective heat insulation. Individual penguins within the group exhibit a local mobility pattern, which results in each individual spending an equal amount of time at the periphery of the huddle, where the heat loss is the maximum. This strategy ensures that the heat loss of the group is evenly distributed across all individuals in the group and thus enables the survival of the group. The work draws a parallel between the heat loss of a penguin at the huddle periphery and the routing energy burden of a sensor node located near the base station.

The distributed mobility algorithm coordinates motion based on the node’s EDR, energy reserve and also its neighbours’ energy vitals (local neighbourhood information only). A sensor node would compare its own energy EDR and level against that of its neighbours. If neighbouring nodes then have a greater predicted lifetime the node would induce a swap. The swap procedure presented by Rao et al [16] is the process of two nodes swapping their physical positions in the network. This strategy ensures that the energy overhead related to relaying packets to the base station is balanced throughout the nodes in the network.

The swap protocol was evaluated, and compared to the performance upper bound derived from a centralised version of the algorithm. The results showed that using the distributed algorithm for controlled node mobility it is possible to significantly extend the network lifetime, whilst considering the energy overhead related to the node physically swapping positions. The results also showed that the lifetime benefits hold in situations when the energy cost of physical node mobility is modelled up to four orders of magnitude larger than the energy cost for packet communication.

2.4.3.4 Controlled Mobility to Improve the Performance of Communication

A mobile WSN will have a dynamic topology due to the physical relocation of sensor nodes. The communication links between sensor nodes will change and thus connectivity will vary. Mobility may also be used as a control primitive to increase or maintain the reliability of communication, through sensor nodes moving to establish connections or break existing links.

Goldenberg et al [72] provided an extensive evaluation on the feasibility of using mobility as a control primitive. The paper presented a mobility control scheme aimed at improving the communication performance of WSNs. This scheme is completely
distributed where nodes only require local topology information for the motion coordination. The scheme is self-adaptive and is able to transparently encompass several modes of operation which each improve the power efficiency for uni-cast, multiple uni-cast, and many-to-one (concast) data packet transmissions.

The mobility control scheme presented aims to reduce the communication energy overheads and latency related to transmitting data packets between a source and destination pair. The communication between the latter may be achieved via multi-hop routing where sensor nodes relay (forward) data packets through the network. The mobility control scheme coordinates the physical relocation of the relay nodes to reduce the communication distance. A relay node computes the average position between its relaying neighbours and moves to the position over a defined number of mobility rounds. Under the scheme, the source and destination nodes do not relocate. The authors argued that if the quantity of data is large and frequent enough, expending energy on physically repositioning the relay nodes to an optimum position for improved communication performance would be justified.

Helmy [73] presented a novel architecture for query resolution in large-scale mobile sensor networks. Nodes maintain information about neighbouring nodes within a zone specified by a number of hops. Information is also retained about sensor nodes outside the zone which are referred to as contacts. During a query, instead of flooding the network to find the answer, only the contacts defined above are queried for information they hold about their associated zone. The contacts may in turn query their own contacts, and so on, until the answer is found. Contacts are initially chosen from nearby neighbours. As they move away from the neighbourhood they discover new neighbours. A salient feature of the architecture is that it takes advantage of mobility to increase the efficiency of query resolution.

2.4.3.5 **Controlled Mobility to Maintain/Increase Coverage**

A mobile WSN can be coordinated to maintain or increase the sensing coverage via the sensor nodes physically repositioning to regions of sparse coverage. The latter may be due to non-uniform deployment, environmental influences and/or sensor node failure. The coverage criterion would be specified by the application performance requirement and is vital to operational lifetime (useful life) of the network. Sensor nodes may also
relocate to provide varying degrees of sensing resolution (coverage) across the region of interest. The following literature addresses the above concepts.

Ganeriwal et al [14] proposed a self aware motion coordination algorithm for fault repair which they refer to as Coverage Fidelity (CoFi). The algorithm uses mobility as an adaptive actuation facility for automated repair of the network with the sole objective of salvaging the lost coverage due to node failures.

The CoFi algorithm consists of four phases, Initialisation phase, Panic request phase, Panic reply phase and the Decision phase. In the initialisation phase, every node learns about its sensing neighbours and calculates its exclusive coverage region (region which only it covers) which would normally be achieved during the WSN configuration. In the panic request phase a dying node requests for the update of the WSN topology and broadcasts its exclusive coverage region (coverage hole) to neighbouring sensor nodes. This would be followed by the panic reply phase, where the sensing neighbours respond to the panic request, stating whether it can cover the coverage hole whilst still covering its exclusive coverage region, the distance it must move and its energy reserve. Under the CoFi algorithm sensor nodes will not move to repair a coverage hole at the cost of overall coverage (a sensor node will not move to increase coverage if it loses some level of coverage). Finally in the decision phase the dying sensor node transmits a message to the neighbour which should physically relocate to repair the coverage hole. The decision is based on evaluating the neighbours’ panic requests and determining the neighbour with the largest energy reserve after relocation.

The algorithm only addresses failures due to energy exhaustion and thus a node knows the time at which it will fail. If a node dies due to malfunction or malicious destruction the algorithm will not repair the coverage holes. Simulations compared CoFi with a static WSN and results showed the fault tolerance associated with the former algorithm. The CoFi algorithm was implemented as part of the work and its performance was compared to the new BayesMob self-healing algorithm presented in Chapter 6.

Wang et al [15] also explored the concept of using mobility to relocate sensors to deal with sensor failure or respond to new events. The authors defined the problem of sensor relocation and proposed a two phase sensor relocation solution: first the redundant sensor nodes are identified. Second the redundant nodes are then coordinated to the target locations. A Grid-Quorum algorithm is proposed which quickly identifies the
redundant sensor nodes with low message complexity. Cascaded movements are then utilised to relocate the redundant sensor to regions with coverage holes. Simulation results are presented which show that the sensor node relocation is achieved in a timely, efficient and energy balanced manner.

Butler et al [17] presented a motion coordination strategy that allows redundant sensor nodes (due to high density) to physically relocate to regions that require a higher sensing resolution. The latter may be desirable if events occur more frequently in a particular region. Two distributed algorithms are proposed, called the history-free technique and history technique. These algorithms coordinate the mobile sensor nodes, such that the distribution of the sensor node group tends towards the distribution of the sensed event. The two algorithms trade-off the amount of computation and memory required to coordinate the physical relocation of the sensor node with the accuracy of the sensors positions.

Also relevant to coverage is environmental influences, such as the presence of obstacles and noise located in the geographical region of interest. These environmental influences introduce sensing uncertainty and may degrade coverage. The environmental influences may be circumvented via the physical relocation of sensor nodes.

Kansal et al [13] proposed that mobility could be used to overcome the effects of unpredictable environmental influences, thus enabling the WSN to adapt to run time dynamics. The authors evaluated the constraints associated with mobility in terms of hardware complexity and the energy overheads related to precise localisation, navigation, and physically moving the nodes. They concluded that complex mobility is not feasible for compact, densely deployed and widespread wireless sensor networks. Kansal et al [13] also presented a low complexity single dimension mobility strategy, which has low energy actuation primitives. The node would move along a single dimension with a view of optimising coverage due to environmental influences such as the presence of obstacles. The mobility strategy is evaluated through extensive simulations and real world experiments. The results show that the low complexity actuation gives significant advantages in terms of WSN coverage.

Kansal et al [74] outlined the design and evaluation of a network of cameras with limited mobility (pan, tilt, and zoom), and used the limited motion for improving sensing performance. The authors considered the issues associated with controlling
motion in a distributed manner. Motion has a high resource overhead, but the work considered a limited form of mobility, such as a camera panning, tilting or zooming. An architecture which allows each node to learn the medium and phenomenon characteristics is presented. Thereby leading to a quantitative metric for sensing performance which is concretely tied to real sensor and medium characteristics, rather than assuming a simple range based model. The problem of determining the desirable network configuration is expressed as an optimisation of this metric. A distributed optimization algorithm which computes a desirable network configuration and adapts it to environmental changes is presented. The relationship of the proposed algorithm to simulated annealing and incremental sub-gradient descent based methods was also discussed.

Based on the hypothesis that incorporating Artificial Intelligence (AI) at the level of the sensor node will improve the mobile WSN’s performance and adaptability, the following section gives a review of AI. The section also reviews literature related to adopting AI techniques to improve the performance of WSNs.

2.5 Artificial Intelligence

2.5.1 Intelligence – the general consensus

In general intelligence can be considered as an umbrella term used to describe a property of the mind that encompasses many related abilities, such as the capacities to reason, plan, solve problems, think abstractly, comprehend ideas, use language, and learn. There are several ways to define intelligence. In some cases, intelligence may include traits such as creativity, personality, character, knowledge, or wisdom. However, most psychologists prefer not to include these traits in the definition of intelligence. At least two major "consensus" definitions of intelligence have been proposed. First, from "Intelligence: Knowns and Unknowns" [75] and [76], a report from a task force convened by the American Psychological Association in 1995:

‘Individuals differ from one another in their ability to understand complex ideas, to adapt effectively to the environment, to learn from experience, to engage in various forms of reasoning, to overcome obstacles by taking thought. Although
these individual differences can be substantial, they are never entirely consistent: a given person's intellectual performance will vary on different occasions, in different domains, as judged by different criteria. Concepts of "intelligence" are attempts to clarify and organize this complex set of phenomena. Although considerable clarity has been achieved in some areas, no such conceptualization has yet answered all the important questions and none commands universal assent. Indeed, when two dozen prominent theorists were recently asked to define intelligence, they gave two dozen somewhat different definitions.'

Another definition of intelligence comes from "Mainstream Science on Intelligence" [77], which was signed by 52 intelligence researchers in 1994:

'A very general mental capability that, among other things, involves the ability to reason, plan, solve problems, think abstractly, comprehend complex ideas, learn quickly and learn from experience. It is not merely book learning, a narrow academic skill, or test-taking smarts. Rather, it reflects a broader and deeper capability for comprehending our surroundings—"catching on", "making sense" of things, or "figuring out" what to do.'

2.5.2 Artificial Intelligence — definition and trends

Artificial Intelligence (AI) is a science and engineering discipline that attempts to create intelligent machines, especially intelligent computer programs with the ability to automate tasks [78]. Examples include control, planning and scheduling, the ability to answer diagnostic and consumer questions, handwriting, speech and facial recognition. The first work in AI was done by Warren McCulloch and Walter Pitts (1943), where they proposed a model of artificial neurons in which each neuron is characterised as being "on" or "off". In the early 1950s, Claude Shannon (1950) and Alan Turing (1953) were writing chess programs to von Neumann-style conventional computers. In 1951 two graduate students in Princeton University Mathematics department, Marvin Minsky and Dean Edmonds, built the first neural network computer. This early work became the foundation for modern AI research.

Artificial intelligence (AI) can be divided roughly into two categories these are Conventional AI and Computational Intelligence (CI). Conventional AI mainly uses techniques classified as machine learning, characterised by formalism and statistical
analysis. Methods include: Expert Systems, Case-based Reasoning, Bayesian Networks and Behaviour-based AI. Expert systems are generated in software with two basic components: a knowledge base and an inference engine. The systems mimic an expert's reasoning process. Case-based Reasoning, for instance, is a system that uses a process of solving new problems based on reviewing the solutions of past problems. Behaviour-based AI is based on a modular decomposition of intelligence, where the latter is controlled by a set of independent semi-autonomous modules. A description of Bayesian network is given in Section 2.5.4.

Computational intelligence involves iterative development or learning. Learning is achieved by adapting parameters and connections with a view to optimise the performance of the underlying system. Techniques include: Neural Networks, Fuzzy Systems, and Evolutionary Computation. A Neural Network is an interconnected group of artificial or biological neurons. Connections between the neurons carry weights which are adjusted, through parameter training, to achieve the desired output. Fuzzy Systems are derived from fuzzy set theory which deals with reasoning that is approximate rather than precisely deduced from classical predicate logic. Evolutionary Computation is a biologically inspired technique which includes populations, mutations and survival of the fittest. Evolutionary Computation divides into evolitional algorithms and Swarm Intelligence (SI).

SI is based on the collective behaviour of decentralised self organised systems [79]. Typically SI systems are made up of a population of simple agents interacting locally with one another and with their environment. These agents would follow simple rules, and although there is no centralised control structure dictating how individual agents should behave, local, and to a certain degree random, interactions between such agents lead to the emergence of "intelligent" global behaviour, unknown to the individual agents. Natural examples of SI include ant colonies, bird flocking, herbivores grazing and fish schooling.

Mobile WSNs which adopt the decentralised mobility algorithms created as part of this work, namely BNGRAZ and BayesMob, could be considered as a SI system. Under these mobility algorithms the sensor nodes' motion is based on interacting with neighbouring sensor nodes and there is no centralised motion control mechanism. This motion essentially leads to global behaviour in the sense that the WSN migrates to improve/sustain coverage and connectivity. However, unlike the SI systems which
explicitly determine the fitness of each solution within the problem space using computation, sensor nodes evaluate the fitness of a motion solution by calculating the probability of the WSN performance improving or remaining unchanged.

2.5.3 AI and WSN

Generally the use of AI within WSNs has been historically disregarded due to data processing and energy constraints Yang et al [80]. However in recent years some research has focused on using AI techniques to impart autonomy to the network. The premise is that a network can monitor its operating environment and topology and deal with changes that may affect it as well as changes in the underlying application. These networks are envisaged to operate with minimal human intervention. AI techniques have been applied to deployment, topology control, network mobility, routing, and MAC protocols. AI techniques have also been used for data aggregation. For example, Kulakov et al [81] presented two different data aggregation architectures, with algorithms which use wavelets for initial data-processing of the sensory inputs and artificial neural-networks which use unsupervised learning methods for categorization of the sensory inputs.

Hummel et al [82] presented a study of the applicability of AI technologies, in particular software agent and neural networks to the following challenges in mobile wireless networking: Mobility modelling, mobile wireless networking while integrating sensors and heterogeneous networks, and mobile information access.

Barbancho et al [83] explored the application of neural networks within the networking layer to improve multi-hop routing and path discovery. The authors presented a neural network based routing algorithm. In addition they compared the performance of the algorithm with two well known routing paradigms, Directed Diffusion and Energy-Aware Routing.

Miao et al [84] investigated the self-deployment of heterogeneous mobile sensors using biologically-inspired principles and methodologies. The work assumed that the initial deployment is random. Two alternative deployment strategies aimed at static and mobile WSNs are presented. The aim of the proposed deployment strategies is to achieve an optimum network configuration with respect to the network topology and region of interest. The paper first presents an optimal deployment pattern for
heterogeneous WSNs. The pattern is based on a technique inspired by the retina mosaic pattern widely observed in both human and many animal visual systems. The heterogeneous sensor nodes are manually organised into the mosaic deployment patterns to yield maximum coverage with minimum network overheads. The second deployment strategy presented is a Swarm Intelligence (SI)-based sensor movement strategy with the assistance of local communication (communication between neighbouring sensor nodes), through which the sensor nodes self-organise themselves to reach the optimal deployment configuration. Simulation results compared the SI-based deployment strategy with a random deployment strategy and showed that the SI-based strategy yields a faster convergence time and smaller energy overheads.

Yang et al [80] presented a swarm intelligence based surveillance protocol for WSNs with mobile supervisors. Supervisors are data sinks which query sensor nodes under their control and in response collect sensing data. The results showed that surveillance protocol handles data communication between static sensor nodes and mobile supervisors such that network and individual sensor node lifetime is maximised.

Sugawara et al [85] presented an investigation into the emergence of swarm intelligence through robots (mobile sensor nodes) performing the task of foraging. Foraging is a type of parallel search strategy, and is normally engaged in by animals and insects when searching for food. The work assumed the foraging task is analogous to that of social insect such as ants and bees. The investigation showed that the efficiency of the swarm depends on the interaction duration between each robot and that the optimum interaction duration depends on the task at hand.

2.5.4 Bayesian Networks

Bayesian probability theory generally referred to as Bayes’ theory was introduced by Thomas Bayes (1702 – 1761). It is a brand of statistics at variance with traditional statisticians or ‘frequentists’ which interprets the probability of an event occurring as the limit of its relative frequency in a large number of trails [86]. Bayesian interprets the concept of mathematical probability as a measure of state of knowledge or subjective degree of belief and is centred on Bayes’s formula (2.1). The latter relates the conditional and marginal probabilities of two events (A and B).
\[
P(A | B) = \frac{P(B | A)P(A)}{P(B)}
\]  \hspace{1cm} (2.1)

- **\(P(A)\)** is the prior probability or marginal probability of \(A\). It is "prior" in the sense that it does not take into account any information about \(B\).

- **\(P(A | B)\)** is the conditional probability of \(A\), given \(B\). It is also called the posterior probability because it is derived from or depends upon the specified value of \(B\).

- **\(P(B | A)\)** is the conditional probability of \(B\) given \(A\) and is referred to as the likelihood function.

- **\(P(B)\)** is the prior or marginal probability of \(B\), and acts as a normalizing constant.

Intuitively, Bayes's theorem in this form describes a way of updating the belief in observing 'A' having observed 'B'. This theory can also be extended to create a Bayesian network, see [87], [88].

A Bayesian network is a graphical structure which can be used to represent and reason about an uncertain domain. These networks are directed acyclic graphs. The Bayesian network nodes represent a set of random variables associated with the domain/system. Nodes can represent any kind of variable, be it a measured parameter, a latent variable or a hypothesis. They are not restricted to representing random variables, which represent another "Bayesian" aspect of a Bayesian network. The nodes are connected by directed arcs (or links) which encode the conditional interdependencies between the variables. Arcs must be connected in a way which prohibits nodes returning to themselves by simply following directed arcs. An example of a Bayesian network is shown in Figure 2.1.

Suppose that there are two events which could cause a sensor node to fail: either the node becomes energy exhausted or it malfunctions. Also, suppose that the malfunction has a direct effect on whether the node becomes energy exhausted (namely a malfunction causes the node to exhaust all energy). This situation can be modelled with the Bayesian network shown in Figure 2.1. All three variables have two possible values \(T\) (for true) and \(F\) (for false).

The joint probability mass function is given in equation (2.2).

\[
P(F, E, M) = P(F | E, M)P(E | M)P(M)
\]  \hspace{1cm} (2.2)
where the names of the variables have been abbreviated to $F = \text{Node Failure}$, $E = \text{Energy Exhausted}$ and $M = \text{Malfunction}$.

The Bayesian network can help answer questions like "What is the probability that the node has malfunction, given that the node has failed?" This would be calculated by using equation (2.3) which uses the conditional probability formula and summing over all nuisance variables (marginalising).

$$P(M = T | F = T) = \frac{P(F = T, M = T)}{P(F = T)} = \sum_{E \in \{T, F\}} P(F = T, E, M = T)$$

$$\sum_{E, M \in \{T, F\}} P(F = T, E, M)$$

(2.3)

When generating a Bayesian network the variables of interest are identified by answering the question: what are the nodes to represent and what is their range of values? Assuming nodes only take discrete values, they should be both mutually exclusive and exhaustive. This means that any variable must take on exactly one of these values at any one time. Common types of discrete nodes include:

![Figure 2.1 Example of a Bayesian network]
- Boolean nodes, which represent propositions by taking the binary values true and false.

- Ordered values. For example, in the case of a node in a mobile WSN the direction might represent a sensor node's neighbour direction and take the values {north, south, east, west}.

- Integral values. For example, a node called Relay might represent the number of times a data packet has been forwarded in the network, ranging from 0 to 20.

The structure of a Bayesian network qualitatively captures the relationships (interdependences) between variables. For example, two nodes should be connected directly if one affects or causes the other one to change, with the arc indicating the direction of the effect. If there is an arc from node A to node B, A is called a parent of B, and B is a child of A. The set of parent nodes of a node $X_i$ is denoted by $\text{parents}(X_i)$. The joint probability distribution associated with the node can be written as equation (2.4) which is the product of the local distributions of each node and its parents.

$$P(X_1, ..., X_n) = \prod_{i=1}^{n} P(X_i | \text{parents}(X_i)) \quad (2.4)$$

If node $X_i$ has no parents, its local probability distribution is said to be unconditional, otherwise it is conditional. If the value of a node is observed, then the node is said to be an evidence node.

The relationships between connected nodes are quantified by conditional probability distributions. Assuming discrete variables, these distributions take the form of Conditional Probability Tables (CPT). The CPT for each node is generated by evaluating all the possible combinations of values of its parent nodes. For example, the probability associated with each of the child node's value(s) will be specified by taking into account each instantiation of its parent node(s). This would generally be achieved by statistical data and/or knowledge from a domain expert.

Reasoning about the domain of interest is achieved by conditioning upon new information (also referred to as evidence) which is performed by the flow of information through the network. The process of conditioning is also referred to as
probability propagation, inference, or belief updating. The types of reasoning within Bayesian networks include:

- **Predictive reasoning** – reasoning from causes to effects and follows the direction of the network arcs.
- **Diagnostic reasoning** – reasoning from effects (symptoms) to causes and occurs in the opposite direction of the network arcs.
- **Intercausal reasoning** – reasoning about mutual causes of a common effect.

In addition, hybrids of the above types of reasoning may be used, to accommodate situations when reasoning does not fit neatly into the types above. This is generally referred to as combined reasoning.

2.6 Summary

The literature review has investigated mobile WSNs, WSN performance measures and commercial simulation environments/software for WSNs. Mobility models, strategies and algorithms for WSNs, which improve performance, are investigated. A brief overview of AI is outlined, including the current research trends and the application of AI within WSNs. This demonstrates that, to the best of the author’s knowledge, Bayesian networks have not been applied to the WSN motion coordination problem. In addition the Grazing mobility strategy and associated algorithms developed as part of this work has not been proposed for WSN. The literature search has also shown that the novel Coverage Against Time has not been previously used to evaluate the WSN coverage. The value of this literature review is to demonstrate the feasibility of adopting a mobile WSN and provide additional evidence of performance benefits.
3.1 Introduction

This chapter presents a new WSN simulator which was developed as part of the work presented here and the motivation behind its creation. Included is a detailed discussion of the simulator engine and architecture. The simulator considers the following aspects of a WSN: sensing, processing, communication, energy, mobility, localisation, configuration and the performance metrics considered in this work. A new mechanism, called the Contacts Information Table (CIT), for storing neighbour and non-neighbour information, local to each sensor node is outlined. The CIT was created as part of the work presented here.

3.2 A New WSN Simulation Environment (WSN Simulator)

3.2.1 Motivation

The new WSN simulator was developed to investigate, develop and test mobility strategies and algorithms for mobile WSN. The simulators discussed in the literature review (Chapter 2) are generic in their approach to simulating wireless networks, with the exception of SENSE which is applicable to WSNs. Having reviewed (and tested) existing simulators, it became clear that these simulators focus on the communication aspects of wireless networks. Therefore, additional functions and protocols would need to be generated to tailor the simulators to evaluate the mobility and performance characteristics. In addition, these simulators are generally freeware and have limited user support. Hence, incorporating additional functionality requires a comprehensive knowledge of the simulator's source code and programming language. The motivation for developing a new simulation environment is that it can be tailored to WSNs and more specifically to evaluating the mobility and performance aspects. The proposed WSN simulation environment (WSN Simulator) can also be utilised by application designers as a WSN mobility and performance evaluation tool.
The WSN Simulator was developed in Matlab mainly because of the support available and relative ease of use. The package offers powerful visualisation which speeded up the development process. Matlab also offers matrix-based programming, advanced mathematical functions, and includes various toolboxes to handle complex functions. The following outlines the development and operation of the WSN simulator and the accompanying models.

3.2.2 Simulator Engine & Code Structure

The WSN simulator comprises of the following parts/functions:

- Graphical User Interface (GUI), focussing on ease of use and visual representation of WSN operation. The simulator is based around the GUI which is initialised by `GUI_MK1.m`.

- Node and simulation environment initialisation, this task is handled by `node_gen.m`. The function initialises all variables associated with the simulation.

- Main simulation engine, the simulator uses concurrency to emulate multiple entities (nodes) in real time using a single processor. The main simulation engine is handled by `concurrent.m` which divides the processor time among the sensor nodes and simulation related overheads.

- Data communication, the simulator emulates sensor node communication by passing packet headers and calculates the energy overheads. This task is handled by `data_com_energy.m`.

- Mobility, the simulator includes a number of mobility models and algorithms.

- Configuration/re-configuration, sensor node configuration is coordinated by `cal_config_energy.m`. The function also calculates the energy overheads related to configuration. Part of the configuration process consists of neighbour discovery which is handled by `node_neighbour.m`.

- Performance evaluation, the simulator is capable of evaluating coverage, connectivity and lifetime which are calculated by `coverage_cal.m`, `sim_con_scan.m` (includes an additional neighbour calculation performed by `sim_neigh_cal.m`) and `individ_node_life.m` respectively. The WSN simulator
GUI also shows the predicted WSN lifetime which is calculated by `WSN_predict_life.m`.

An illustration of the interrelationship between these WSN simulator parts/functions is shown in Figure 3.1. A flow diagram of the simulation process is shown in Figure 3.2 and a flow diagram for the simulation engine (`concurrent.m`) is shown in Figure 3.3. The functions included in both of these flow diagrams are presented throughout this chapter. The WSN simulator source code is given in Appendix B.

![Figure 3.1 The interrelationship between the parts/functions of the WSN simulator](image)
Figure 3.2 Flow diagram of the simulator's initialisation and run procedure
CHAPTER 3 - WSN SIMULATOR DEVELOPMENT

Processor slot for Sensor Node (SN = 1)

Run mobility strategy/algorithim functions for SN

Configuration (cal_config_energy.m) for SN

Data communication (data_com_energy.m) for SN

Calculate node lifetime (individ_node_life.m) for SN

SN == N (number of nodes)

WSN Predicted Lifetime (WSN_predict_life.m)

Simulator neighbour discovery (sim_neigh_cal.m)

Connectivity calculation (sim_conn_scan.m)

Coverage calculation (coverage_cal.m)

Generate replay data (data_movie.m)

STOP Button

STOP

Figure 3.3 Flow diagram for concurrent.m
3.2.2.1 GUI_MKI Graphical Set up

The front end of the WSN simulator is the Graphical User Interface (GUI). The latter includes a dynamic plot of the geographical region of interest where the WSN is displayed. It also includes a series of buttons to control the simulator and call boxes to set the desired simulated WSN parameters (application criteria). The simulator is initiated via the `GUI_WSN.m` function. A graphical view of the WSN simulator is shown in Figure 3.4.

![Figure 3.4 The GUI of the WSN Simulator](image)

The geographical region of interest defaults to a square area with the size set in the GUI. The following parameters can be set within the GUI: communication range, sensing range, WSN size (number of sensor nodes), mobility level, motion strategy/model, coverage resolution, desired coverage period and the desired performance plots.

The operation of the WSN simulator is controlled via the following buttons:
• **Deploy** - calls the `node_gen.m` function which initialises the WSN simulator parameters and generates the sensor nodes and associated variables;

• **Start** – calls the `concurrent.m` function which starts the simulation engine;

• **Stop** – pauses/stops the simulation engine;

• **Plot** – generates the supported performance plots selected by the user;

• **Start & Stop Replay** - The WSN simulator also supports the facility to replay the simulation at an increased speed. The replay is started via the replay button which calls `replay.m` and stop via the replay stop button which calls `replay_stop.m`.

The following simulation parameters can be set via the GUI:

• **Geographical region of interest size – area under investigation by the WSN.**

• **Mobility model** – Mobility model/motion coordination algorithm.

• **Mobility level** – percentage of sensor nodes which have motion capabilities.

• **Desired performance results/plots** – coverage, connectivity etc.

• **Sensing range** – the maximum Euclidean distance at which a node can accurately detect a change in the sensed variable.

• **Communication range** – the maximum Euclidean distance at which a node can successfully communicate with a neighbouring node.

• **Number of sensor nodes**, deployed to the region of interest.

• **Desired total coverage value** – percentage of the region of interest covered, see Chapter 4.

• **Desired coverage period** – frequency at which total coverage is achieved, see Chapter 4.

• **Coverage resolution** – the resolution at which coverage is measured, see 3.2.3.1.
3.2.2.2 Node and Environment Initialisation

The simulation initialisation procedure is triggered by the deploy button, which calls the `node_gen.m` function. The latter handles the initialisation of the simulation environment and the associated sensor nodes. This includes generating the structures and variables that represent each sensor node. The variables stored by the simulator are divided into two categories; information available to sensor nodes and simulated related variables (information only known by the simulation environment). This allows the simulator to emulate a real WSN where the sensor nodes would have limited information. Each sensor node is represented by a structure element called 'meteo'. Among the information stored in the latter are: node identification, position, speed, direction, neighbouring node data, local performance measures, reference map of the geographical region, and the new Contacts Information Table (CIT). The structure of the CIT is outlined in section 3.2.2.3.4. The WSN simulator enables the user to set the initial power reserve (joules) for each simulated sensor node via the `node_gen.m`.

Simulation related variables generated include: message handling structures, coverage measure structure, information related to the sensor nodes and GUI plot objects. The user can select a random or manual deployment. The former generates the nodes' positions via a uniform distribution. The latter enables the user to set the individual positions of each sensor node. The function also executes the initial sensor node configuration and performance evaluation.

3.2.2.3 Concurrent Main Simulator Engine

The main simulator processes are handled by `concurrent.m`. The latter is called when the start button is depressed. `concurrent.m` shares the processor time between all sensor nodes and the simulation related overheads, such as updating the GUI and evaluating the network's performance. Concurrency is simulated by allocating each node a predefined time slot. Within the latter, all the necessary tasks relating to the sensor nodes' operation are performed. The processor time slots will depend on the processing overheads related to each sensor node. The default value is 200 ms. However; the user is free to set this value to suit the processing requirements. The default time slot was selected by running Matlab's profiler which returns the sensor nodes processing requirements (processing times). The simulators processing time for
servicing the entire WSN once can be calculated from equation (3.1). This is also shown in Figure 3.5.

\[
\text{WSN processing time} = (\text{number of nodes} \times \text{processor time slot}) + \text{simulator overheads}
\] (3.1)

The simulator sequentially allocates sensor nodes their processing time slot. Hence the WSN processing time is deterministic (the time until a sensor node is allocated its next time slot is equal to the WSN processing time). The WSN processing time is taken into consideration when each node executes its associated tasks. Therefore, the WSN simulator compensates for the use of a single processor and effectively emulates a real time system. For example, if the sensor node moves at a speed of \(V\), the distance a node should move within its allocated processing slot is calculated from equation (3.2).

\[
\text{Distance} = V \times \text{WSN processing time}
\] (3.2)

Figure 3.5 Concurrency within WSN simulator using time division of processor
The sensor nodes' tasks performed during their processing time slot include: motion coordination and navigation, configuration and data transmission. The simulation related runtime overheads include:

- Calculating performance data.
- Displaying performance data.
- Storing performance data.
- Executing administration tasks such as updating the GUI and storing the data associated with replaying the simulation.

All these tasks are executed by calling their associated functions.

The following assumptions are made:

- The WSN simulator enables a user to set the sensing range via the simulator's GUI. The sensing region of the nodes was assumed to be a circular disc centred at the nodes' positions and with a radius equal to the sensing range, shown in Figure 3.4. The WSN simulator assumes a Boolean sensing model (the coverage for any given point in the region of interest is either true or false) and therefore the circular disc represents the area covered by each sensor node.

- The WSN simulator does not include any allowance for the processing and data storage characteristics. The energy overhead associated with processing is also omitted as it is minimal in comparison to communication and motion overhead.

- The WSN simulator enables the user to set the desired communication range via the simulator's GUI which adjusts the energy overheads. The assumed communication region of the nodes is homogeneous and is represented by a circular disc centred at the positions of the nodes and with a radius equal to the communication range. The WSN simulator also assumes a Boolean communication model and therefore nodes that reside within each other's communication range can communicate effectively. Additional communication models may be added to the simulator to enable a realistic representation of communication. However, this is outside the scope of the work presented here.

- The simulator does not evaluate performance such as propagation delay, fading, throughput etc. But the simulator does evaluate the energy associated with
communication. These performance measures are not vital to the success of the research.

- Localisation protocols are omitted from the WSN simulator as it is beyond the scope of the research. The WSN simulator assumes that each sensor node is aware of its own position through either the configuration messages (using localisation protocol) or a GPS (Global Positioning System) attached to the sensor node platform. The implications associated with these approaches have not been investigated. Localisation is an integral part of the WSN. Precise node positions are essential to navigation, routing, and sensing. The data packets should contain location information associated with the sensing data.

3.2.2.3.1 Data Communication

The data communication within the WSN simulator is handled by the `data_com_energy.m` function. The data packet generation and transmission is periodic. However, users may adapt the source code to support query and event driven communication. The transport layer protocols have been omitted from the current version of the WSN Simulator. A greedy path routing protocol was adopted. Under the greedy path routing protocol [4], a sensor node, whenever possible, transmits its data packets and packets received from neighbouring nodes directly to the base station, providing the latter is within its communication range. Alternatively, the sensor node forwards the data packets to the neighbour which is closest to the base station. The simulator assumes a perfect MAC layer.

The `data_com_energy.m` function does not pass the actual data packets between source and destination. Only the packet header is passed between the sensor nodes (source and relaying nodes) and base station (destination). The header contains source and destination information. The latter includes a timestamp, source identification and position. The header also includes routing information and a relay counter associated with the number of nodes which have forwarded the data packet. When a sensor node wishes to transmit a data packet it loads the packet header into the base station’s or neighbouring sensor node’s communication buffer. The neighbouring node, which receives the forwarded packet, will depend on the greedy path routing protocol. This process is repeated between neighbouring nodes until the packet header resides in the
base station buffer. The base station may carry-out some performance evaluation and analysis before clearing it from its buffer.

The communication energy overhead related to transmitting data packets between the source and destination is emulated via the exchange of the packet header. The simulator calculates the energy associated with transmitting and receiving the packet header (data packet) and reduces the sensor nodes energy reserve respectively.

The following communication energy model was adopted from Heinzelman et al [9].

Energy spent in transmission (joules) = \( e_d b d^\alpha + e_r b \) \hspace{1cm} (3.3)

Energy spent in reception (joules) = \( e_r b \) \hspace{1cm} (3.4)

Energy spent sensing (joules) = \( e_s b \) \hspace{1cm} (3.5)

Where \( e_d \) is the energy dissipated per bit per metre\(^2\) for transmission and has been set to 100×10\(^{-12}\) J, \( e_r \) is the energy spent by transmission circuitry per bit and has been set to 50×10\(^{-9}\) J, \( e_r \) is the energy spent by reception circuitry per bit and has been set to 50×10\(^{-9}\) J, \( e_s \) is the energy spent per bit and has been set to 50×10\(^{-9}\) J, \( b \) is the number of bits to transmit or receive, \( d \) is the distance from transmitter to receiver and \( \alpha \) is a constant \( \geq 2 \) which depends on the attenuation the signal will suffer in that environment.

The adopted energy model uses \( \alpha = 3 \). The distance is equal to the communication range set by the user. The data packet size including the header and footer (and possible error check facility) has been set to 64 bytes for all simulations herein. The WSN simulator provides facilities to set these parameters to suit the simulated communication hardware and communication protocols.

3.2.2.3.2 Mobility

The WSN simulator incorporates a number of mobility models, strategies and algorithms which the user can select via the GUI. These include the following entity mobility models (node movements are independent of other nodes):

- Random Walk Mobility (RWM) – **RWM.m**
- Random Way Point (RWP) – **RWP.m**
• Random Gauss Markov (RGM) – RGM.m

Group mobility models (node movements are dependent of other nodes) included are:

• Pursue mobility model – pursue_mobi.ity.m

• Reference Point Group Mobility (RPGM) – RPGM.m

These mobility models are outlined in Camp et al [45]. The new mobility strategies and algorithms presented in Chapters 5 and 6 are also available, these include:

• BNGRAZ – BNGRAZ.m

• GRPGM – GRPGM.m

• BayesMob – BayesMob.m

The user also has the facility to implement and test additional mobility models and algorithms. The proportion of sensor nodes which are mobile is referred to as the mobility level. The WSN simulator GUI enables the user to set the mobility level by specifying a value between the range of zero and one, completely static and completely mobile respectively.

The assumed motion energy model is based on the X4e robot platform [89], [90]. The latter consists of four HS-311 servos (continuous running). Each servo consumes 4.4 J m⁻¹ and therefore the entire motion energy for the platform is 17.8 J m⁻¹. These energy values are derived from the following calculations:

Circumference of wheel = \( C = 232 \text{ mm} \)

Servo current drain = \( i = 1080 \text{ mA/rotation} \)

Speed 360 degrees / 0.9 s

Servo voltage = \( v = 6 \text{ v} \)

\[
\text{Energy per rotation (joules)} = i \times v \times t \\
\text{Energy per metre (joules)} = \frac{1000}{C} \times i \times v \times t
\]

The position of all sensor nodes with relation to the geographical region of interest is displayed via the WSN simulator GUI. When a sensor node moves, the new coordinates are updated via the plot_data.m function.
3.2.2.3.3 WSN Configuration/Re-configuration

Configuration and re-configuration are vital to the effective operation of the network. The WSN would generally be re-configured when the topology changes due to node failures, node movement and/or deployment of additional sensor nodes. As such, re-configuration can be periodic and/or event driven. Network configuration is achieved by communication between neighbouring nodes. Therefore the configuration carries a communication overhead. The WSN simulator assumes a periodic configuration model, and enables the user to set the duty time. During configuration, a sensor node will transmit one configuration packet and also receive a configuration packet from each of its neighbouring sensor nodes. The configuration packet consists of 32 bytes. The communication energy overheads for configuration are derived from (3.3), (3.4) and (3.5) given in 3.2.2.3.1. The configuration tasks and overheads are handled by the `cal_config_energy.m` function.

3.2.2.3.4 Neighbour Discovery & CIT

Sensor nodes that are located within each other’s communication range are defined as neighbours. Neighbour discovery and gathering of associated information is achieved via the network configuration. Neighbour information may consist of identification, position, motion characteristics and energy. This information may assist with the following tasks: generating routing tables, localisation, navigation and synchronisation (time and possibly hibernation cycles). Neighbour discovery within the WSN simulator is coordinated via the `node_neighbour.m` function.

A new mechanism, called the Contacts Information Table (CIT), for storing neighbour and non-neighbour information was created as part of the work presented here. Each CIT entry (neighbour or non-neighbour information) is referred to as a contact. A sensor node’s CIT is updated during the neighbour discovery process (`node_neighbour.m`). The CIT also stores non-neighbouring sensor nodes’ information gathered from forwarding data packets. The latter information is extracted from the data packets’ header. Figure 3.6 shows an example of the CIT.
Figure 3.6 Contacts Information Table (CIT), local to sensor nodes

The CIT stores the following parameters associated with each contact entry: identification code; timestamp; relay count; motion speed; motion direction; energy and coordinates. The WSN simulator assumes that each sensor node has a unique identification code. The timestamp stores the time associated with the latest communication with the contact (last time the contact entry was updated). The relay count is associated with non-neighbour contacts and stores the number of sensor nodes which have forwarded the data packet prior to the sensor node receiving the packet. The motion characteristics (represented by the speed and direction) and the energy reserve associated with neighbouring nodes are also stored in the CIT.

The contacts’ position coordinates are represented by two Gaussian distributions (along \(x\) and \(y\)). Alternative distributions may be added if required. The mean (\(\mu\)) fields associated with these distributions store the coordinates (position within the region of interest). The latter remains constant until the next configuration time (communication with the contact). The Standard Deviations (SDs or \(\sigma\)) represent the uncertainty associated with the contacts’ coordinates (information). In addition, the assumed Gaussian distributions provide a mechanism for quantifying the interdependencies between sensor nodes.

When the contacts’ entries are updated the SD represent the uncertainty associated with the localisation (contacts positions). The default SD, when the contacts are updated, is 1 m. This would represent a sharp distribution with minimal variation around the mean. Between updating the contacts’ information, the standard deviations are periodically incremented with respect to time and the contacts’ last known speed. The speed of non-neighbour contacts can be approximated by the maximum feasible node speed.
Therefore the variation around the mean would increase, thus representing increased uncertainty associated with the contacts’ position. The SD is calculated using:

$$\sigma = \sigma + \nu \cdot t_{update}, \quad \sigma = \sigma_x = \sigma_y$$ (3.8)

where \( \nu \) is the contact’s speed and \( t_{update} \) is the SD update duty cycle time.

### 3.2.3 Performance Evaluation

This section outlines the WSN simulator’s performance evaluation capabilities and associated techniques and functions. The WSN simulator is currently capable of evaluating the following performance measures: coverage (sensing), connectivity and network lifetime. A definition of these measures with respect to WSNs is included as well as the modes of operation of the associated performance evaluation protocols and functions.

#### 3.2.3.1 Coverage

The assumed coverage measure in this work is the fraction of the total geographical region collectively covered by the WSN (sensor nodes’ coverage regions). The coverage is calculated by `coverage_cal.m`. Coverage is calculated using a matrix of pixel points which represent the region of interest. This is referred to as the coverage map. The map is generated by overlaying a grid on the region of interest and each intersection of a horizontal and vertical grid line represents a pixel point. The values associated with these pixel points (matrix cells) represent the coverage status. Figure 3.7 shows a graphical representation of the coverage map. The coverage resolution is related to the distance between the horizontal and vertical lines and thus the quantity of pixel points. The coverage map resolution can be set in the simulator’s GUI.

This method of using grid line intersections has been implemented in order to minimise the computational overhead associated with calculating the precise area covered by the WSN. This becomes computationally heavy when dealing with multiple sensor nodes and intersecting coverage regions (the assumed sensor node coverage region is a circular disc with a radius equal to the sensing range). Sekhar et al [91] adopted a
similar coverage evaluation method except the authors use grid squares to represent the region of interest. The coverage is calculated by approximating the fraction of the grid square covered. The approximations are based on the vertices covered.

![Geographical region of interest with a grid overlay to represent coverage map](image)

Figure 3.7 Generating the coverage map by the grid intersections (pixel points) using the user specified resolution

The size of the matrix which stores the pixel points is calculated using (3.9).

\[ Md_c = 1 + \frac{S_x}{R}, \quad Md_r = 1 + \frac{S_y}{R} \]  

(3.9)

If we assume that the geographical region is rectangular or square, \(Md_c\) gives the number of columns + 1 and \(Md_r\) gives the number of rows + 1. Therefore the total number of pixel points that define the coverage map equals \(Md_c \times Md_r\). \(S_x\) is the length of the region along the x axis and \(S_y\) is the length of the region along the y axis. \(R\) is the desired coverage resolution and specifies the distance between the pixel points.

Each pixel point within the matrix stores a series of values which enable the coverage to be effectively calculated. The values include a timestamp, the service interval time, service counter, coverage weight and a current coverage status value. The timestamp stores the time at which the pixel point was last covered. The service interval time
stores the time period between consecutive coverage periods. The service counter stores the number of times the pixel point has been covered within the current coverage period. These values relate to the new WSN Coverage Against Time measure developed as part of the work presented and outlined in Chapter 4. Each pixel point is associated with a coverage weight \( W_{\text{point}}(r,c) \) which ranges between zero and one and represents the application criteria coverage requirement. The coverage weight values relate to the grazing mobility strategy, see Section 5.2. The current coverage status stores a value greater than zero when the pixel point is covered. The value indicates the number of individual sensor nodes (a node’s sensing region) that cover the pixel point. This value can be manipulated to yield a measure of redundancy.

The function `coverage_cal.m` first resets the current coverage status for each pixel point. It then sequentially scans through all the sensor nodes positions and resulting coverage regions and updates the status of the pixel points located within each of the sensor nodes’ coverage regions. The sum of pixel points with a current coverage status value greater than zero yields the quantity of pixel points covered. The WSN coverage value is calculated by (3.10).

\[
\text{Coverage}_{\text{WSN}} = \frac{\sum P_{\text{covered}}}{P_{\text{total}}} \times 100
\]  

(3.10)

\( P_{\text{covered}} \) is the total number of pixel points covered and \( P_{\text{total}} \) is the total number of pixel points that represent the region of interest. The coverage value is displayed on the GUI and also saved to a text file. The WSN simulator gives the user the option to generate a coverage plot from the data stored in the text file once the simulation process is complete. All simulation performance plots are generated by the `plot_perform.m` function.

### 3.2.3.2 Connectivity

Connectivity is evaluated by the `sim_conn_scan.m` function see Figure 3.8 for the function flow chart. The function has access to each sensor nodes' connected status variable. This variable stores a one if the sensor node is connected and a zero otherwise. Initially the connected status of all sensor nodes is reset to zero. The function then
determines which nodes are one hop neighbours (directly connected) of the base station and changes their connected status. Then the function sequentially scans through the sensor nodes (those which are active, with sufficient energy) and evaluates their connected status. If a sensor node is connected it checks all of its neighbours' connection status and updates as required. The function stores the identification of the updated neighbour with the lowest identification value. Providing the updated neighbour's identification is lower than the current nodes identification, the scan is restarted at the updated neighbour's identification. Alternatively, the scan continues from its current point. The scan terminates when all permutation of the sensor nodes and neighbours' connected status has been realised.
Before a reliable evaluation of connectivity can be realised the simulator independently performs a neighbour discovery for each sensor node by the `sim_neigh_calc.m` function. The function determines which sensor nodes are located within each others' communication range. This is similar to `node_neighbour.m`, except the sensor node
neighbour information is stored within the simulators information set, whereas `node_neighbour.m` stores the information within each sensor nodes structure. The WSN connectivity is calculated using (3.11).

\[
Connectivity_{WSN} = \sum \frac{Connected_{Nodes}}{Total_{Nodes}} \times 100
\]  

(3.11)

Where \( Connected_{Nodes} \) is the number of sensor nodes which are connected and \( Total_{Nodes} \) is the total number of deployed sensor nodes. The connectivity value is displayed on the GUI and also saved to a text file. The WSN simulator gives the user the option to generate a connectivity plot from the data stored in the text file, once the simulation process is complete. All simulation performance plots are generated by the `plot_perform.m` function.

### 3.2.3.3 Lifetime

All sensor nodes generated by the WSN simulator monitor their individual EDR by executing the `individ_node_life.m` function. The function effectively calculates the operational energy overheads and deducts these from the node’s energy reserve. The operational overheads include motion and communication costs see energy models given in Section 3.2.2.3. In addition, the function also predicts the sensor nodes lifetime based on the sensor nodes EDR.

The WSN lifetime is predicted via the `WSN_predict_life.m`. This function evaluates the individual nodes lifetime predictions (`individ_node_life.m`) and predicts the WSN lifetime. The WSN lifetime (including prediction) can be set either to the time at which the first sensor node fails (from the start) or that at which the commander node fails. The function also terminates the simulation when either one of these criterion are met.

### 3.3 Summary

This Chapter has describe the Matlab based WSN simulator developed as part of the work presented. The simulator has enabled the author to evaluate the performance of mobile WSN. Hence it can be used as a tool (by network designers) to evaluate the
performance of a network operating under a particular mobility algorithm/strategy. The performance measures which can be evaluated include: coverage, connectivity and network lifetime. The simulation environment incorporates a number of mobility models (in addition to those outlined in Chapters 5 and 6). The communication, sensing, processing and energy models adopted by the WSN have been discussed, including the theoretical analysis behind each sensor node module and the WSN as a global entity. The simulator is currently unable to evaluate communication related performance measure such as fading, scattering, and propagation delay etc. This is outside the scope of this research area. The WSN simulator and models presented here have been used throughout the research, to evaluate the performance of the new mobility strategies and algorithms presented in Chapters 5 and 6. In addition, a new mechanism created as part of this work, called the CIT, for storing neighbour and non-neighbour information was outlined.
4.1 Introduction

This chapter gives a detailed overview of the Coverage Against Time performance metric, developed as part of this work. This was presented in the IADAT-ten2006 conference paper, see Appendix A. The motivation behind generating this new coverage measure and the associated performance metrics are outlined below.

4.2 Motivation

The WSN coverage is a performance measure used to evaluate the area of the region of interest collectively covered by the sensor nodes. When considering a static WSN, sensor nodes positions remain constant throughout the lifetime of the network within the region of interest. Therefore, the application designer would ensure that the number of sensor nodes and deployment configuration satisfies the coverage criteria, referred to as total coverage. In the case of a mobile WSN, the topology will dynamically change over time for the obvious reason. This will invariably affect the performance of the WSN, and more specifically the coverage. The coverage in the general sense will only return a measure of coverage at a specific instant in time, hence referred to as the instantaneous coverage.

In the case of a mobile WSN, total instantaneous coverage can only be achieved if the number of sensor nodes is sufficient to cover the entire region of interest. If the number of nodes is insufficient, total coverage can only be achieved by the sensor nodes physically repositioning (migrating) within the region of interest. Therefore, total coverage can only be achieved over time. When considering the latter the instantaneous coverage would not give a true representation of the WSN coverage. Mobility adds an additional dimension to coverage as the latter would vary with time. Therefore an alternative coverage measure referred to as Coverage Against Time has been created to effectively evaluate coverage when considering a mobile WSN.
4.3 Concept of Coverage Against Time

The Coverage Against Time measure adds an additional dimension to coverage by evaluating the coverage with respect to time. The measure returns three metrics associated with Coverage Against Time. These are the accumulated coverage, coverage period and the coverage service interval. These metrics are a direct consequence of the total coverage not being achieved via the number of sensor nodes and the deployment configuration. Hence the WSN must physically reposition the sensor nodes to achieve the desired level of coverage.

The coverage period represents the time taken for the WSN to converge to total coverage. This is the time between initial deployment and total coverage and also between two consecutive total coverage instants. The application designer would generally define a desired coverage period. If at time \( t \), the coverage equals \( x \) then at \( t + \text{coverage period} \), the coverage would equal total coverage. If the region of interest is represented by pixel points (see 3.2.3.1, measuring coverage) the total coverage is achieved when all pixel points (or a specified percentage) have been covered by the WSN. The accumulated coverage does not evaluate the number of pixel points covered at an instant in time but evaluates which pixel points have been covered within a time period. Therefore, if at time \( t \) the accumulated coverage equals \( x \) then at time \( t + 1 \) the accumulated coverage equals \( x + \text{additional pixel points covered} \). Any change in coverage status for the pixel points covered at time \( t \) does not explicitly impact the accumulated coverage. Therefore, assuming the sensor nodes are physically repositioning to areas within the region of interest, which have not been covered within that coverage period, the accumulated coverage value will increase. Once the accumulated coverage value equals total coverage, the accumulated coverage will reset to the current instantaneous coverage value. Figure 4.1 shows a graphical representation of the coverage period.
The coverage service interval is the time period between a pixel point (defined in section 3.2.3.1) being serviced (covered) in one coverage period and re-serviced in the subsequent coverage period. The service interval does not take into account the covered period but only the period between coverage, see Figure 4.2. Continuous coverage and re-coverage of an area within the same coverage period will reduce the service interval.
The coverage *service interval* was included to account for the *coverage period* not giving a true representation of the *service interval* for each pixel point. If a pixel point is covered at the beginning of one coverage period, it may not be re-covered until the end of the subsequent coverage period. Figure 4.3 shows a graphical representation of the relationship between the *coverage period* and *service interval*.

![Diagram of Accumulated Coverage value vs. Time](image)

**Figure 4.3 Variation between Coverage Period and Service Interval**

4.4 Performance Metrics

The pixel point *service intervals* are used to generate a service interval distribution. An example of the distribution is shown in Figure 4.4. These distributions enable the application designer to evaluate the likelihood of the service interval being contained within a specified boundary. The Coverage Against Time measure generates a tool for
CHAPTER 4 - COVERAGE AGAINST TIME

predicting the performance of a mobile WSN at the design stage. Note that the application designer would need a WSN simulator to explore this avenue.

The pixel point service interval distributions have a peak at the first bin. This indicates that some of the pixel points have a service interval less than the bin size. Also some service intervals may equal zero. This is due to pixel points covered at initial deployment and the start of a new coverage period. The number of sensor nodes, deployment configuration and spread of the WSN will influence the size of this peak. The remaining pixel point service interval times will depend on number of sensor nodes and motion control algorithm.

The maximum service interval time will depend on the specified total coverage which can be set to less than 100% via the WSN simulator’s GUI. The maximum service interval time when the total coverage equals 100% is determined using equation (4.1).

\[
\text{maximum service interval time} = (2 \times \text{coverage period}) - \text{pixel point coverage time}
\] (4.1)

The maximum would be observed if a pixel point is covered at the beginning of one coverage interval and the end of the subsequent period. When considering a total coverage value which is less than 100% the maximum service interval time will invariably be determined by the motion control algorithm. In effect a pixel point could...
be serviced in coverage period \(x\) and only re-serviced in period \(x + 2\). This would be observed because every pixel point would not have to be covered to achieve the total coverage value and thus a new coverage period. The number of pixel points this will affect is determined by the size of the region of interest and also the specified total coverage. Therefore the proportion of the distribution that reflects the total coverage value should only be considered when evaluating the pixel point service interval distribution.

All the metrics generated by the Coverage Against Time performance measure enable the application designer to evaluate the expected and measure the post-implementation coverage with respect to the deployment characteristics and motion control algorithm. Figure 4.5 shows an example of typical coverage periods and mean and max services interval returned from the Coverage Against Time measure whilst the WSN’s mobility is coordinated using the new BNGRAZ algorithms (these plots are linked).

![Example of Coverage Periods](image)

![Examples of Mean and Max Service Intervals](image)

**Figure 4.5 Examples of coverage periods and mean, max service intervals**

### 4.5 Summary

This Chapter has described a new performance measure for WSNs which was developed as part of the work presented. This is referred to as Coverage Against Time. This measure evaluates the WSN coverage with respect to time. The Coverage Against Time measure returns two metrics, the coverage period and service interval. These metrics have enabled the author to evaluate the performance of mobile WSNs, in particular the mobility strategies and algorithms presented in Chapters 5 and 6.
Chapter 5

GRAZING MOBILITY STRATEGY

5.1 Introduction

This chapter outlines the concept of grazing in the context of mobile WSNs. The resulting mobility strategy enables a WSN to provide total coverage of a region of interest when the number of sensor nodes alone is not sufficient to cover the region. Three motion coordination algorithms namely, Fixed Path, Grazing Reference Point Group Mobility (GRPGM) and Bayesian Network GRAZing (BNGRAZ) which implement grazing were created. The Fixed Path algorithm adopts a deterministic motion pattern along a fixed pre-defined path, set by the network designer. GRPGM is a centralised mobility algorithm, where the commander node (mobile base station) broadcasts control signals through the network to control the mobility of all other sensor nodes. Finally, BNGRAZ is a decentralised algorithm, based on Bayesian networks, which enables a node to reason about its own motion characteristics with respect to performance (connectivity and coverage). As part of this algorithm a new decentralised Coverage Approximation (CA) algorithm, which enables a sensor node to approximate the WSN coverage via local neighbourhood information (one-hop neighbours) was devised and is presented herein.

In addition, the performance of the WSN whilst adopting these mobility algorithms was compared to the Random WayPoint (RWP) mobility model under which sensor nodes move through the region of interest according to a random distribution, as presented by Johnson et al [23].

5.2 Principle of WSN Grazing

WSN grazing is a motion strategy for mobile WSNs in which nodes are made to behave like herbivores grazing pastures. The geographical region of interest for a WSN becomes analogous to the pasture, and the information collected by the sensor nodes analogous to grass. Grazing is defined as the consumption of one organism (grass/information) by another organism (grazing mammal/sensor node) (Lyon and...
Machen [92]). Grazing can also be classified as a type of foraging. The technical meaning of foraging within the science of behavioural ecology refers to predator-prey interactions, [85]. Grazing differs from predation-prey interactions in that the organism being eaten is not killed [93], and differs from the feeding of parasites in that the two organisms do not stay together very long, nor is the grazer so limited in what it can eat.

The grazing strategy provides an alternative data collection methodology for WSNs. Unlike a static WSN which rely on a large deployment of sensor nodes, the grazing strategy considers the deployment of a smaller number of mobile sensor nodes which do not provide total coverage at any instant. Total coverage is continuously obtained via the WSN migrating (sensor nodes physically changing position) around the region of interest. From any instant t, total coverage is achieved at time $t + WSN$ migration time (time taken by the WSN to completely cover the region of interest). This motion means that the WSN can continuously adapt to environmental and application changes.

Under the grazing strategy, the analogy between grass and information provides the WSN with the means to evaluate the need for coverage. This is particularly relevant in situations where the coverage provided by the WSN can only be achieved through migration (typically a small number of mobile nodes migrate around the region of interest collecting data). Using this analogy to grazing, Figure 5.1 depicts the correlation between the pasture height and the coverage weight – the latter ranges from zero to one. A value of one indicates that data has recently been collected from that area (analogous to grazing mammals having completely exhausted the grass in that area of the pasture). A value of zero indicates that data has not been collected from that area for a time period greater than or equal to the desired coverage period (the grass is long and ready for grazing). When referring to the coverage calculation presented in section 3.2.3.1, these values represent the pixel point coverage weights ($W_{p}^{\text{pixel}}(r,c)$). During the coverage period the coverage weight value is linearly decremented at a rate specified by the desired coverage period. For example, if a pixel point is covered at time $t$ it will have a weight of one. Then assuming that the pixel point has not been recovered, the value at time $t +$ desired coverage period will equal zero. All values between these limits would represent the coverage status. The way in which the coverage weight is used to aid migration is described below.
5.3 Design Considerations

The implementation of the grazing strategy requires additional design considerations. Among the latter are navigation, localisation, configuration, and resource constraints. The proposed application and environment would also govern the feasibility of implementing a mobile WSN which adopts the grazing strategy.

The grazing strategy states that the WSN should continuously migrate around the region of interest to provide total coverage. As a result, sensor nodes must physically relocate within the region of interest. This motion adds additional strain to the limited resources of the nodes and possibly impairs their performance. For example, the topology changes may lead to increased data transfer latency due to collisions resulting from a non-perfect MAC layer. When mobility is implemented motion becomes the largest energy overhead, unlike static WSNs where communication is the largest energy overhead. Driving onboard motors and servos can severely reduce the lifetime of the WSN. These energy overheads may be reduced by attaching the sensor nodes to external mobile entities. For example within an urban environment, the sensor nodes could be attached to buses which follow set routes through the region of interest. Goldenberg et al [72] suggested that a more fanciful example is a system of simple living organisms such as insects which are outfitted with radio transmitters and whose motion is controlled by a neuro-electronic interface.

Figure 5.1 Grass height relationships to the coverage weight value
The high energy overheads associated with a mobile WSN may also be alleviated by adopting energy harvesting techniques which collect energy from the environment. For example, sensor nodes may carry onboard solar cells, or recharge via docking stations situated within the region of interest. Adopting these techniques will increase the feasibility of implementing the grazing strategy.

Mobile WSNs also require an increased configuration rate (how often a sensor communicates with its neighbours) to ensure sufficient information is available to the sensor nodes. This is due to topology and environment changes resulting from the sensor nodes physically relocating within the region of interest. Typical tasks which are performed during configuration are localisation, navigation, generating routing tables, and neighbour discovery. These tasks are vital to the successful coordination of the grazing strategy. The configuration process carries overheads in terms of energy and bandwidth and therefore the duty time should be extended to ensure that the overheads are kept to a minimum. The extension of the configuration duty time induces a level of uncertainty into the WSN and therefore mobility algorithms capable of handling this uncertainty are required. The following section presents mobility algorithms created as part of this work.

5.4 Mobility Algorithms

Three mobility algorithms for mobile WSNs have been created as part of this work, namely Fixed Path mobility, centralised Grazing Reference Point Group Mobility (GRPGM), and the decentralised Bayesian Network based algorithm (BNGRAZ). Section 5.4.4 presents a comparison between these algorithms is given in terms of the networks performance and also the design considerations discussed above.

5.4.1 Fixed Path Mobility Algorithm

The fixed path approach to mobility assumes that the user or application designer predetermines the roaming path of the WSN. This assumes that the user has prior knowledge of the environment in which the WSN is to be deployed. Once the WSN is deployed each sensor node continuously follows its defined roaming path, until failure occurs or a new path is defined.
Under this approach, if a point within the region is covered at instant \( t \) then the time until its next service will be \( t + \text{time until a sensor node re-covers that point} \). The time taken for the sensor nodes to migrate around the region is fixed; and data collection is deterministic.

Problems arise when the user has limited knowledge of the environment under observation and so an optimum roaming path may be difficult to achieve. Many of the proposed applications include deploying the network to unmanned regions. Therefore, the lack of information about the region can jeopardise the efficiency of the defined roaming path. The fixed path approach is not fault tolerant, as node failure can (and usually does) create coverage holes, unless a high level of redundancy is used, which is not cost effective. If a single node fails, the useful life of the WSN is jeopardized. The fixed path WSN is also not capable of adapting to any application and environment changes unless a high level of user control is adopted.

The fixed roaming path adopted here moves the nodes around the region on a fixed radius from the centre of the region of interest, see Algorithm 5.1.

### Algorithm 5.1: Fixed path algorithm.
Assume manual deployment */deploy each node to the desired \( x \) and \( y \) coordinates/*

\[
\text{fixed}_\text{radius} = \sqrt{(x_{\text{centre}} - x)^2 + (y_{\text{centre}} - y)^2}
\]

\( d_{\text{max}} = \text{maximum distance from centre of region to perimeter} - \text{sensing range} \)

While(1) */ continuously execute /*

move around the region on a fixed \( \text{radius} \) from the centre in the anti-clockwise direction

\[
\text{node speed} = \frac{\text{speed} \times \text{fixed}_\text{radius}}{d_{\text{max}}}
\]

Figure 5.2 shows a typical roaming path for this solution. Nodes at the outer periphery move faster and further than the centre nodes. The net effect is that nodes at the outer periphery have a higher EDR and exhaust their energy quicker. A consequence of this approach is that different areas of the region vary in their coverage time and service interval.
Alternative roaming paths could be defined that improve on the performance of the presented approach. These could include a roaming path that balances the energy overhead associated with node mobility, ensuring that all nodes move an equal distance. Also the user may specify that every point within the region should have an equal service interval. Hybrids that include the pursuit-mobility model discussed in Camp et al [45] may also be generated, in which the user only defines the roaming path of the leader node (pursued). The other sensor nodes then become pursuers. However the system is not adaptive and thus the roaming path may never ensure total coverage.

5.4.2 Grazing Reference Point Group Mobility (GRPGM) Algorithm

The GRPGM behaviour is based on the Reference Point Group Mobility (RPGM) model Hong et al [94]. It is a centralized approach where the commander node explicitly controls the migration of the sensor nodes. The RPGM model was proposed to emulate the group mobility behaviour of individual users that collaborate to form an ad-hoc network. RPGM generates reference points within the region of interest using a uniform distribution. Each mobile node then selects a random position within the vicinity of the reference point. The vicinity area would generally be user defined. All nodes then move to their position at a defined speed and wait for a new reference point to be generated.
The GRPGM algorithm differs from the RPGM model in that the reference points are generated by the commander node to ensure the sensor nodes cover the region of interest. Whereas RPGM generates the reference points randomly and does not consider the coverage requirement. It is assumed that the commander node operates in a similar fashion to the base station. However the commander node also carries mobility capabilities. Therefore all data generated by the sensor nodes would be routed to the commander node before possible data aggregation occurs and being forwarded to the sink/user. This allows the commander node to monitor the position of all transmitting nodes by capturing the data packet header.

The commander node then generates a map of the geographical region. This represents the collective coverage of the WSN, and previously covered areas. The commander node then calculates the areas within the region that requires coverage and generates a new reference point. The latter is then broadcast to all nodes in the network where they select a position which is uniformly distributed within a specified vicinity of the reference point. The value that determines the size of the vicinity area will be directly related to the quantity of sensor nodes deployed. The position of the reference point and vicinity area is the same for each node, ensuring that all nodes are contained in the same vicinity of the reference point. The latter is adopted to minimize the likelihood of sensor nodes losing connectivity. All the sensor nodes would then migrate to their randomly generated position. Once all nodes reach their position, the commander node calculates and broadcasts a new reference point.

The sensor nodes periodically generate data packets destined for the commander node, which extracts each node’s location information from the packet header, and updates its local coverage map of the region. The commander node’s coverage map is generated and updated using a similar procedure to that described in Section 3.2.3.1. However, the pixel point values associated with the commander node’s map only store a coverage weight, \( W_{\text{point}}(r, c) \) see Section 5.2.

The reference points are determined by dividing the region of interest into equally sized sub-regions. The number and size of the latter will depend on the region of interest size. The available reference points are based on the centre coordinates of each sub-region. The commander node uses its coverage map and equation (5.1) to assign a coverage weight for each of the sub-regions.
$W_{\text{sub-region}}(i) = \frac{\sum W_{\text{point}}(r,c)}{T(i)}$ (5.1)

$W_{\text{sub-region}}$ is effectively the average pixel point coverage weight for each sub-region $i$ and ranges between zero and one. A weight of one indicates that the sub-region is completely covered by the WSN. The commander node follows Algorithm 5.2 to determine the new reference point.

**Algorithm 5.2: Procedure for determining the next reference point.**

1. Set current sub region to the region in which the commander node resides
2. /*Search for the lowest faction of covered pixel points in the sub-regions directly surrounding current sub-region.*/
3. for all one hop sub regions that surround the current sub region
   1. Determine the minimum sub region coverage weight (Using equation 5.1 to calculate the sub-region coverage weight).
4. if minimum coverage weight of the surrounding sub regions is less than the current sub region coverage
   1. Set new reference point x coordinate to minimum sub region column index multiplied by sub region size minus half of the sub region size.
   2. Set new reference point y coordinate to minimum sub region row index multiplied by sub region size minus half of the sub region size.
5. else
   1. New reference point equal old reference point

One consequence of adopting the GRPGM is that the nodes select random positions within the vicinity of the reference point. The sensor nodes may position themselves such that redundancy is high, effectively due to all nodes positioning themselves on top of each other, this may result in poor performance. Nodes may also become disconnected from the network. The above may result in the WSN migrating through the region of interest a number of times before total coverage is achieved.

The GRPGM algorithm carries additional communication overheads, due to the broadcast of reference points. The broadcast of the reference points was achieved via a flooding routing protocol. Therefore sensor nodes may be required to relay the reference point to neighbouring nodes. The centralized nature of GRPGM eliminates
the need for sensor nodes to determine their optimum direction, and therefore reduces the strain on the limited computational resources available to the sensor node.

5.4.3 BNGRAZ Algorithm

The Bayesian Network GRAZing (BNGRAZ) algorithm adopts a distributed approach to motion control. BNGRAZ uses Bayesian networks (section 2.5.4) to reason about motion with respect to performance. Bayesian networks were adopted to enable sensor nodes to base their move on statistical data. As such sensor nodes can reason about their deployed environment and handle the uncertainty related to their local topology. Computational AI techniques would not be suitable for mobile WSNs due to the limited processing, memory and energy constraints.

The Bayesian networks were used for predictive reasoning. They were configured to predict the probability of a change in performance given that a sensor node moves in a particular direction. The predictions are made by the sensor node and are based on a possible move in one of the cardinal directions north, south, east, or west. The performance aspects under consideration are connectivity and coverage (Coverage Against Time).

The BNGRAZ only considers discrete variables so the relationships between connected nodes are represented by a Conditional Probability Table (CPT), see Appendix E. Therefore, for each distinct instantiation of the parent node the value that the child node will take is specified via the CPT. Values within BNGRAZ have been specified with a view to achieving the desired response from the Bayesian networks.

The structures of the Bayesian networks were designed using the Netica development software from Norsys [95]. The latter provided a mechanism for experimenting with alternative BN structures and CPT values. The BN structures were created by first evaluating the hypothesis (posterior). Secondly the conditional interdependencies between the latter and variables (specifically information available to the sensor node which incur minimal overhead to acquire) within the WSN were then evaluated. Typically these variables are acquired from local neighbours. The variables were interconnected with directed arcs which encoded the conditional interdependencies.
The methodology for determining the CPT values for each node was expert knowledge gained through trial and error experiments. The Netica software enabled the parent node values to be set, whilst observing the child node’s status. The value the child node takes is determined from its parent values and also the CPT. Setting the parent node values provided a mechanism for tuning the CPT values to achieve the desired response from the Bayesian networks. All Bayesian networks were implemented into Netica to evaluate their performance before being implemented into the associated motion algorithms which were developed in the Matlab simulation tool.

BNGRAZ incorporates three discrete Bayesian networks that predict the probability of the WSN performance increasing or remaining unchanged given the WSN topology and sensor node’s motion direction. The variables used in BNGRAZ are outlined in Table 5.1. Bayesian network 1 (Bn1) determines the probability that connectivity will decrease if the sensor node moves in one of the cardinal directions \( (C_i = T) \). Bayesian network 2 (Bn2) determines the probability that an un-serviced area will be discovered in cardinal direction \( i \) \( (D_i = T) \). Bayesian network 3 (Bn3) aggregates \( C_i \) and \( D_i \) to determine the probability of the optimum direction \( O_d \) of motion which will maintain or increase the WSN performance from a connectivity and coverage perspective. The structure for BNGRAZ is shown in Figure 5.3. Figure 5.4 shows the structure of Bn3.
Table 5.1 Definition of variables for BNGRAZ

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>True</td>
</tr>
<tr>
<td>F</td>
<td>False</td>
</tr>
<tr>
<td>N</td>
<td>Cardinal direction north (45 to 135 degrees)</td>
</tr>
<tr>
<td>S</td>
<td>Cardinal direction south (225 to 315 degrees)</td>
</tr>
<tr>
<td>E</td>
<td>Cardinal direction east (315 to 45 degrees)</td>
</tr>
<tr>
<td>W</td>
<td>Cardinal direction west (135 to 215 degrees)</td>
</tr>
<tr>
<td>i</td>
<td>Cardinal direction indices (N, S, E, W)</td>
</tr>
<tr>
<td>C</td>
<td>Sensor node communication range</td>
</tr>
<tr>
<td>C_1</td>
<td>Connectivity decrease in cardinal direction i ∈ {True, False}</td>
</tr>
<tr>
<td>D_i</td>
<td>Discover un-serviced area (area that has not been covered by the WSN) in cardinal direction i ∈ {True, False}</td>
</tr>
<tr>
<td>N_i</td>
<td>Need to move in cardinal direction i ∈ {True, False}</td>
</tr>
<tr>
<td>O_i</td>
<td>Optimum direction i ∈ {N, S, E, W}</td>
</tr>
<tr>
<td>K_i</td>
<td>Neighbour lying in cardinal direction i ∈ {True, False}</td>
</tr>
<tr>
<td>L</td>
<td>Neighbour distance &lt; C_i ∈ {True, False}</td>
</tr>
<tr>
<td>A_i</td>
<td>At least one neighbour is lying in cardinal direction i ∈ {True, False}</td>
</tr>
<tr>
<td>d_i</td>
<td>At least one neighbour’s distance &lt; C_i in cardinal direction i ∈ {True, False}</td>
</tr>
<tr>
<td>U_i</td>
<td>Un-serviced area (area that has not been covered by the WSN) in cardinal direction i ∈ {True, False}</td>
</tr>
<tr>
<td>θ_n</td>
<td>Motion angle of sensor node i ∈ {N, S, E, W}</td>
</tr>
</tbody>
</table>

Bn1
Probability of connectivity decreasing

Bn2
Probability of discovering un-serviced area (coverage)

Bn3
Probability of maintaining or increasing WSN performance

Sensor node move in the direction that yields the maximum performance

Figure 5.3 BNGRAZ structure
The joint probability function for Bn3 is given in (5.2). The probability of \( O_d = i \) (optimum direction of motion = N, S, E or W) is calculated by marginalizing (5.2), which is shown in (5.3).

\[
P(O_d = i) = \sum_{N_N, N_S, N_E, N_W} P(N_N)P(N_S)P(N_E)P(N_W) \times P(O_d = i | N_N, N_S, N_E, N_W)
\]

where \( P(N_N) \) to \( P(N_W) \) are calculated using (5.4) and (5.5):

\[
P(N_i, C_i, D_i) = P(C_i)P(D_i)P(N_i | C_i, D_i)
\]

Marginalize equation (5.4) to calculate \( P(N_i = T) \)

\[
P(N_i = T) = \sum_{C_i, D_i \in \{F, P\}} P(C_i)P(D_i)P(N_i = T | C_i, D_i)
\]

### 5.4.3.1 Bayesian network 1 (Bn1): Probability of a Decrease in Connectivity

Given information concerning neighbouring nodes Bn1 calculates the conditional probability of a decrease in connectivity if a sensor node moves in one of the cardinal
directions. The former translates into, for instance, the probability of a neighbour lying in each of the cardinal directions and the probability that the closest neighbour's distance in each direction is less than the communication range \((C_r)\) away. Connectivity is defined as the ability of a sensor node to communicate effectively with the commander node and thus with the user. The structure of Bn1 is shown in Figure 5.5.

\[
A_N \quad d_i \quad \cdots \cdots \quad A_W \quad d_i
\]

\[
N_N \quad N_S \quad N_E \quad N_W
\]

\[
\Theta_m \rightarrow C_i
\]

**Figure 5.5 Connectivity decrease Bayesian network 1 structure**

The joint probability function for Bn1 is given in (5.6). The probability there is not a decrease in connectivity given that the node moves in direction \(i\) \(P(C_i = F | \theta_m = i)\) is calculated by using the joint mass probability function and marginalization, which is given in (5.7). The probability there is not a decrease in connectivity, given that the sensor node moves in one of the cardinal directions is dependent on the direction and distance from neighbouring nodes.

\[
P(C_i, \theta_m, N_N, N_S, N_E, N_W) = P(\theta_m)P(N_N)P(N_S)P(N_E)P(N_W)P(C_i | \theta_m, N_N, N_S, N_E, N_W) (5.6)
\]

Using the joint probability mass function and marginalizing,

\[
P(C_i = F | \theta_m = i) = \frac{P(C_i = F, \theta_m = i)}{P(\theta_m = i)} = \sum_{N_N, N_S, N_E, N_W \in \{F\}} P(N_N)P(N_S)P(N_E)P(N_W)P(\theta_m = i) \times P(C_i = F | \theta_m = i, N_N, N_S, N_E, N_W) (5.7)
\]
where \( P(N_{N}) \) to \( P(N_{w}) \) are calculated using (5.8).

\[
P(N_{i} = T) = \sum_{A_{i}, d_{i} \in \{T, F\}} P(A_{i}) P(d_{i}) P(N_{i} = T \mid A_{i}, d_{i}) \quad (5.8)
\]

The belief in the evidence \( A_{i} \) (at least one neighbour is lying in cardinal direction \( i \)) and \( d_{i} \) (at least one neighbour's distance in cardinal direction \( i \) is less than the communication range) is calculated from the \( P(K_{i} = T) \) (probability that a neighbour is lying in cardinal direction \( i \)) and \( P(L = T) \) (probability that a neighbour's distance is less than the communication range). \( P(K_{i} = T) \) and \( P(L = T) \) are calculated from the Contacts Information Table (CIT) which is local to each sensor node, see sections 5.4.3.1.1 and 5.4.3.1.2. The format, generation, and update of this table are outlined in Section 3.2.2.3.4. A sensor node's motion is dependent on its neighbours' position and motion characteristics. Taking into consideration a neighbour's position coordinates only, for a given performance prediction, the prediction will be incorrect if the neighbour physically relocates during the configuration duty time. The Gaussian distributions associated with neighbours' positions in the CIT enable a sensor node to manage the uncertainty associated with neighbour movements during configuration duty time. The distributions are also a way of quantifying the interdependence between a node's motion and that of its neighbours and vice versa as each neighbour maintains a similar distribution for the sensor node under consideration.

The following two sections outline the calculation of \( P(L = T) \) and \( P(K_{i} = T) \). These probabilities are calculated from the distance and angle distributions associated with a node and its neighbour. The distance and angle distributions are calculated from the \( x \) and \( y \) Gaussian distributions which represent a neighbour's coordinates.

### 5.4.3.1.1 Probability that the Distance between a Node and its Neighbour is Less than the Communication Range

The Gaussian distributions associated with each of the contacts' coordinates are used to calculate the probability that the distance between a node and its neighbours is less than the communication range \( (P(L(j) = T) \) where \( j = 1 \ldots n \), and \( n \) equals the total number of neighbours from the CIT). If a neighbour's coordinates, \( x \) and \( y \) values, are assumed to be independent Gaussian random variables with nonzero means, then the probability
density function (p.d.f.) \( f_s(z) \) for the distance between a node and its neighbour is represented by a Rician distribution. Where \( z(j) \) equals the Euclidean distance between a node and its neighbour and is calculated from \( z(j) = \sqrt{x(j)^2 + y(j)^2} \), \( x(j) \) equals the distance along the x axis and \( y(j) \) equals the distance along the y axis. The equation for the Rician distribution is given in (5.9).

\[
f_s(z) = \frac{ze^{-(z^2+\mu^2)/2\sigma^2}}{\sigma^2} I_0 \left( \frac{z\mu}{\sigma^2} \right)
\]

where

\[
\mu = \sqrt{(\mu_x(j) - x_{SN})^2 + (\mu_y(j) - y_{SN})^2}, \quad \theta = \tan^{-1} \left( \frac{\mu_y(j) - y_{SN}}{\mu_x(j) - x_{SN}} \right), \quad \sigma = \sigma_x = \sigma_y
\]

\[
\mu_x = \mu \cos \phi, \mu_y = \mu \sin \phi
\]

and

\[
I_0(\eta) = \frac{1}{2\pi} \int_0^{2\pi} e^{\eta \cos(\theta-\phi)} d\theta = \frac{1}{\pi} \int_0^{\pi} e^{\eta \cos \theta} d\theta
\]

(5.10)

\( x_{SN} \) and \( y_{SN} \) are the coordinates of the sensor node and \( \mu_x \) and \( \mu_y \) are the mean neighbour coordinates from the CIT. Equation (5.10) is the modified Bessel function of the first kind and zeroth order. The probability that the distance between a node and its neighbours is less than the communication range (\( P(L(j) = T) \)) is approximated by numerically integrating the Rician distribution between zero and the (communication range) \( C_r \). The SD (\( \sigma \)) should be higher than one or the Bessel function tends to infinity, resulting in an undefined distribution \( f_s(z) \).

### 5.4.3.1.2 Probability of a Neighbour Lying in Each of the Cardinal Directions

The probability of a neighbour lying in each of the cardinal directions (\( P(K_i(j) = T) \)) where \( j = 1 \ldots n \), and \( n \) equals the total number of neighbours from the CIT), would be calculated from integrating the p.d.f. \( f_\theta(\Theta) \) (where \( \Theta \) represents the angle from the node at which a neighbour is situated), between 45 and 135 degrees for north, 135 and 225 degrees for west, 225 and 315 degrees for south and 315 and 45 degrees for east. Where
θ(j) equals the angle and is calculated from \( \theta(j) = \tan^{-1}\left(\frac{y(j)}{x(j)}\right) \). \( x(j) \) equals the distance between a node and its neighbour along the x axis and \( y(j) \) equals the distance along the y axis. Since \( x \) and \( y \) have been assumed to be independent normal random variables with non-zero means, the angle p.d.f. \( f_\theta(\theta) \) is intractable (to the best of the author’s knowledge and research). Please see Appendix C for workings which validate this statement.

The angle p.d.f. is approximated by a normal distribution, after modelling the angle distribution from two joint random normal distributions which essentially represent a neighbour’s \( x \) and \( y \) coordinates. The mean and SD (SD = \( x \) SD = \( y \) SD) values of the \( x \) and \( y \) distributions were varied to evaluate the resultant angle distribution. The source code and a sample of the angle distributions generated from this analysis are given in Appendix F.

The angle distribution was modelled by first generating two normal distributions for \( x \) and \( y \), each consisting of 500 samples. The angle population consisting of 250,000 samples was calculated using (5.11) for each combination of the \( x \) and \( y \) distribution samples.

\[
\theta = \tan^{-1}\left(\frac{y}{x}\right)
\]  
(5.11)

The angle population was used to calculate the mean and SD for the generated distribution using (5.12) and (5.13).

\[
\bar{\theta} = \frac{1}{N} \sum_{i=1}^{N} \theta_i
\]  
(5.12)

\[
\sigma_\theta = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\theta_i - \bar{\theta})^2}
\]  
(5.13)

A selection of the angle distributions generated is given in Appendix B for a distance of 14.14 m \( (distance = \sqrt{\mu_x^2 + \mu_y^2}) \). Figure 5.6 and Figure 5.7 show the angle distribution for an \( x \) (or \( y \)) SD of 1 and 31 m. These distributions are generated from the angle population via the Matlab distribution fit tool. The distributions show that whilst the \( x \)
(or y) SD is less than a third of the distance the angle distribution fits a normal distribution. As the SD is increased the angle distribution begins to spread between ±180°. Therefore the normal distribution fit becomes poor, and the angle distribution tends towards a uniform distribution. This is shown in Figure 5.7. However under the operating conditions of the system the normal distribution provides an adequate approximation of the angle distribution. The maximum x (or y) SD for a neighbour within the CIT is 30 m (after which point the contact is removed from the CIT) and the sensor nodes aim to maintain a mean distance from its neighbours between 15 and 20 m (assuming the communication range is set to 20 m).

Figure 5.6 Angle distribution with normal fit (x (or y) SD = 1 m)
Figure 5.7 Angle distribution with normal fit (x (or y) SD = 31 m)

Figure 5.8 shows the angle SD relationship to the x (or y) SD, for a randomly selected set of distance values (distance between a node and its neighbour). Observations from Figure F.28 in Appendix F show that when x (or y) SD are equal to the distance, the resultant angle SD is approximately 63°, see Figure 5.8. Also when the x (or y) SD are equal to four times the mean distance, the resultant angle SD is approximately 94°. The maximum angle SD observed from the trials is 100°. Using the Matlab fitting tool, the angle SD (a function of x (or y) SD) can be modelled as the response of a first order system to a step input of magnitude 10 where the distance (distance = \(\sqrt{(\mu_x(j) - x_{SN})^2 + (\mu_y(j) - y_{SN})^2}\), \(\mu_x(j)\) and \(\mu_y(j)\) = mean neighbour coordinates from CIT and \(x_{SN}\) and \(y_{SN}\) = sensor node coordinates) can be substituted for \(\tau\) (the time
constant) and the SD (x or y SD) is the independent variable (see equation (5.14)) for expression of angle SD in the Laplace domain.

\[
\sigma_\theta(s) = \frac{10}{s(s + \frac{1}{\tau})}
\]  

(5.14)

where \(\tau = \text{distance}\) and \(\text{time} = \sigma_x (x\ or\ y\ SD)\)

Applying the inverse Laplace Transform to equation (5.14) yields:

\[
\sigma_\theta(\sigma_x) = 100 - 100e^{\frac{-\sigma_x}{\tau}}; \quad \sigma_x = \sigma_y 
\]  

(5.15)

Figure 5.9 shows the actual angle SD with respect to the x (or y) SD and also the modelled angle SD generated from equation (5.15). There is some variation between the actual and modelled SD due to the limited number of samples when generating the x and y jointly normal distributions.

Algorithm 5.3 was developed to calculate the probability of a neighbour lying in each of the cardinal directions \(P(K_i(j) = T)\). Each sensor node executes the algorithm for each of the neighbours stored in the CIT.
Algorithm 5.3: Approximate $P(K_i(j) = T)$

$X_{SN}, Y_{SN}$ = sensor node’s coordinates, and $\sigma = \sigma_x = \sigma_y$

$\text{distance} = \sqrt{(\mu_x(j) - x_{SN})^2 + (\mu_y(j) - y_{SN})^2}$ = distance between the node and its neighbour.

Assume $f_{\theta}(\theta)$ is represented by a Gaussian distribution with

$$\sigma_{\theta}(\sigma_{xy}) = 100 - 100e^{-\frac{\sigma_y}{\text{distance}}}$$

$$\mu_{\theta} = \tan^{-1}\left(\frac{\mu_y}{\mu_x}\right)$$

/*Probability of direction is approximated by integrating $f_{\theta}(\theta)$ between
45 to 135 and 405 to 495 degrees for north,
135 to 225 and -225 to -135 degrees for west,
225 to 315 and -135 to -45 degrees for south,
315 to 360 and 0 to 45 and -45 to 0 and 360 to 405 degrees for east */

5.4.3.1.3 Determining the Evidence for Bn1

The probabilities generated so far only consider individual neighbours. In order to generate the evidence for Bn1 (the probability of a decrease in connectivity given a move in particular cardinal direction) these probabilities ($P(K_i(j) = T)$ and $P(L(j) = T)$ need to be combined. This returns $P(A_i = T)$ (probability that at least one neighbour is lying in cardinal direction $i$) and $P(d_i = T)$ (probability at least one neighbour’s distance in cardinal direction $i$ is less than the communication range). The individual probabilities for neighbours’ directions and distances are combined using equations (5.16) and (5.17).

$$P(A_i = T) = 1 - \prod_{j=1}^{n} P(K_i(j) = i)$$  \hspace{1cm} (5.16)

$$P(d_i = T) = 1 - \prod_{j=1}^{n} P(L(j) \times P(K_i(j) = i)$$  \hspace{1cm} (5.17)
The $P(L(j) = T)$ is scaled with respect to the belief that the neighbour is in direction $i$ ($P(K_{i}(j) = T)$). This ensures that $P(d_{i} = T)$ is dependent on the probability that the neighbour is in cardinal direction $i$ (under this assumption the $P(d_{i} = T)$ may never equal one).

5.4.3.2 Coverage Approximation (CA) Algorithm for Bn2

In order to determine the probability that a sensor node will discover an uncovered area within the region of interest, i.e. areas which have not been covered by the WSN for a period of time longer than the desired coverage period, a second Bayesian network (Bn2) was created, see Figure 5.15. The evidence required for this prediction is obtained from a coverage map maintained locally by each sensor node. The coverage map is generated and updated using a procedure similar to the one described in Section 3.2.3.1. However in the case of Bn2, the pixel point values associated with the sensor node’s map only store a coverage weight ($W_{poin}(r,c)$), see Section 5.2. The resolution was arbitrarily set to 10 m, to keep the computation and memory load to a manageable level.

The sensor nodes must therefore maintain their coverage map to ensure sufficient evidence is available for Bn2. The former requires WSN coverage information to ensure this is achieved. Depending on the WSN size, this information would not be available to a sensor node. Usually each sensor node is aware of its own roaming path and areas which it has covered, as well as which areas its neighbours have covered. However, the sensor node would not know which areas have been covered by the entire WSN. This lack of global knowledge adds uncertainty related to determining the optimum direction that yields the maximum likelihood of discovering an un-covered area.

The WSN coverage information can be obtained via the base station periodically broadcasting the information or the sensor nodes periodically querying all sensor nodes to obtain their positions. These methods would enable the sensor nodes to maintain an accurate coverage map. However they would carry significant communication overheads (bandwidth, energy). An alternative solution involves sensor nodes using local neighbourhood information to approximate the WSN coverage, by way of the
novel Coverage Approximation (CA) algorithm created as part of the work presented here.

The CA algorithm produces a decentralised approximation of the WSN coverage so as to enable each sensor node to generate and maintain a coverage map. The approximation is derived from a sensor node’s neighbours’ positions, the position of sensor nodes that relay data packets through the node, and some general assumptions about the WSN topology. The contact (neighbour and other sensor nodes) information can be obtained from the sensor node’s CIT discussed in section 3.2.2.3.4. In addition, the CA algorithm must know the total number of active sensor nodes that form the WSN (WSN size). The accuracy of the approximation is dependent on the quantity and quality of the contact information. The contact information related to relaying data packets is not guaranteed, as it is dependent on the network topology and routing protocol. In terms of scalability, the size of the WSN will affect the accuracy of the coverage approximation see section 5.4.3.2.1 below.

The CA algorithm assumes that the WSN retains a mesh topology and that each sensor node is an equal distance away from each of its neighbours. This implies that sensor nodes have at least one neighbour in each of the cardinal directions (north, south, east, west), with the exclusion of sensor nodes that reside on the outer periphery of the WSN. The CA update algorithm is presented in Algorithm 5.4, see Appendix B for the CA algorithm source code.

The CA algorithm first approximates the size of the coverage area, which it assumes is a circular region. This is achieved by determining the Euclidean distance between the node and each neighbour and calculating the mean distance between the node under consideration and its neighbours (mean neighbour distance). The CA algorithm then approximates the coverage area by assuming the WSN is a grid and that the spacing between grid intersections is equal to the mean neighbour distance. The number of grid intersections is equal to the total number of sensor nodes. The area covered by the grid is then transposed into an equivalent circular region.

The CA algorithm then attempts to determine the position of the approximated WSN coverage region in relation to the sensor node concerned. This position is determined by evaluating the node’s neighbours’ positions (direction only) and that of contacts sending/relaying information. The information gathered in this way is used by the node
to position the centre of the approximated coverage area. If a sensor node has a neighbour in each of the cardinal directions, it assumes that it must be approximately in the centre of the WSN topology. Therefore the centre of the approximated area is assumed to be the sensor node’s own coordinates. However, if the node has additional contact information it will adjust the centre of the WSN to ensure that the contact is contained within the approximated area, effectively shifting the WSN’s position towards the furthermost contact. If the sensor node is positioned on the edge of the WSN, it shifts the approximated area centre in the direction of the neighbouring nodes. Again, if additional neighbour information is available, the shift is adjusted to ensure that the contacts are retained within the approximated coverage area. The shift distance will be equal to the approximated circle radius minus the sensing range. The approximated coverage area is used to update the local coverage map.

Validation results for the CA algorithm are presented below in section 5.4.3.2.1. The CA algorithm was run in conjunction with the configuration procedure, when the local knowledge of the WSN was updated.
Algorithm 5.4: WSN CA (Coverage Approximation)

*/approximate the coverage area of the WSN/**

Calculate the mean distance from a node and its neighbour.

\[
\text{mean neighbour distance} = \mu_{\text{distance}} = \frac{\sum_{j=1}^{n} \sqrt{(\mu_x(j) - x_{SN})^2 + (\mu_y(j) - y_{SN})^2}}{n}
\]

where \( x_{SN} \) and \( y_{SN} \) are the sensor node coordinates and \( j = 1 \ldots n \), \( n \) equals the total number of neighbours from the CIT with mean coordinates \( \mu_x(j) \) and \( \mu_y(j) \).

Approximate an assumed circular coverage area of the WSN using the mean distance and quantity of sensor nodes.

\[
\text{Coverage}_{\text{area}} = \pi R_c^2 = \left( \mu_x(\sqrt{i_s} - 1) + 2S_s \right)^2
\]

where \( i_s = \text{total number sensor nodes}, S_s = \text{sensing range}. \)

Determine the radius of the approximated coverage area.

\[
\text{Coverage}_{\text{radius}} = R_c = \sqrt{\left( \frac{\text{Coverage}_{\text{area}}}{\pi} \right)}
\]

*/rules to determine the centre position of the approximated circular WSN coverage area/**

Calculate the direction of all neighbouring nodes.

\[
\text{direction} = \theta = \tan^{-1} \left( \frac{\mu_y(j) - y_{SN}}{\mu_x(j) - x_{SN}} \right)
\]

Determine the cardinal direction (north = 45 to 135°, south = 225 to 315°, east = -45 to 45°, and west = 135 to 225°) of neighbouring nodes based on their direction.

**If** neighbouring nodes lie in all cardinal direction and no other routing information is available

Position approximated WSN coverage area centre at nodes coordinates and add to coverage map.

**else**

**if** only neighbouring node information is available and in one or more of the cardinal directions there are no neighbours

Shift approximated WSN coverage centre by WSN coverage radius minus sensing range. To the centre of all other cardinal directions that contain neighbours.

**else if** other routing information is available from contact nodes that rely on the node forwarding data packets to the commander node.

Shift equals distance from contact plus sensing range minus WSN approximated coverage radius. Shift in the direction of contact node.

**end**
5.4.3.2.1 CA Algorithm Overheads, Accuracy and Scalability

The overheads associated with the CA algorithm are minimal due to the decentralised nature of the algorithm. Obtaining the information required to generate the coverage approximation carries no additional overhead, as it is already stored in the CIT. Therefore, additional communication is not necessary to update the coverage map. The base station needs to inform all nodes when a failure occurs, so that the algorithm can accommodate all active nodes in the approximation. A small processing overhead will also be incurred by executing the CA algorithm. However, the reduced overheads do come at the cost of accuracy.

In terms of accuracy, the coverage approximation essentially has two associated sources of error; the size of the coverage area, and the position of the coverage region in relation to the sensor node. The error related to the size of the coverage area is dependent on the size of the WSN, number and position of neighbouring nodes, sensing range, and configuration duty time. The algorithm assumes that the distance between all nodes is equal, and is assumed to be the mean distance from all neighbours. As the number of nodes in the network increases, this assumption becomes less accurate and therefore the associated error increases. This is also dependant on the sensing range. A larger sensing range will generally yield a bigger error. A greater number of neighbouring nodes will increase the accuracy of the mean neighbour distance which should reduce the respective coverage error. The configuration duty time will also influence the uncertainty related to the stored positions of the neighbours. The longer a node remains without communicating with neighbours, the larger the uncertainty and accordingly the error. The assumption that the WSN coverage area is circular clearly introduces additional error due to areas being assumed as covered when they are not and vice versa. Although not evaluated separately, the impact of this error is taken into account when simulating the CA algorithm.

The error related to the position of the approximated coverage area is dependent on the WSN size and also the sensor node's position within the network. Appendix D characterises the maximum coverage error and shows that its magnitude is determined by the distance error. The distance error is the distance between the centre of the actual and approximated coverage regions. The calculation for the maximum distance error is given in (5.18).
\[ d_{\text{error}} = \mu_d - (R_c - S_r) - \frac{\mu_d (i_n - 3)}{2} = \mu_d - \left( \frac{\mu_d \left( \sqrt{i_n} - 1 \right) + 2S_r}{\pi} - S_r \right) - \frac{\mu_d (i_n - 3)}{2} \] (5.18)

\( R_c \) is the radius of the approximated WSN coverage region, \( i_n \) equals the number of sensor nodes that form the WSN, \( S_r \) is the sensing region, and \( \mu_d \) is the mean neighbour distance.

The analysis demonstrates that the maximum distance error is dependent on the sensing range, mean neighbour distance and most importantly the number of sensor nodes that form the WSN. In addition, it shows that if geographical region of interest size is fixed, the CA algorithm is not scalable and the respective coverage error increases with respect to the WSN size. A definition of scalability is given in section 2.3. The scalability issues are shown in Figure 5.10. It shows that the maximum \( d_{\text{error}} \) increases with the number of sensor nodes due to the approximated coverage radius (\( R_c \)) increasing. The \( S_r \) was set to 10 m and the \( \mu_d \) was set to 15 m.

The analysis assumes that the maximum distance error calculation occurs when all the sensor nodes are positioned in a line and the sensor node at one end runs the CA algorithm. Therefore, the latter positions the approximated region in the direction of its neighbour such that the sensor node is within the region.

![Figure 5.10 CA algorithm scalability plot](image-url)
The performance of the CA algorithm was evaluated by plotting the approximated coverage area alongside the actual WSN coverage and also plotting the resulting coverage approximation error. This coverage error is generated by evaluating each sensor nodes coverage map pixel point values against the actual coverage. If a pixel point within the nodes' coverage map is incorrectly plotted as covered, or uncovered, it results in a coverage error. The coverage error is represented as a percentage of pixel points incorrectly plotted against the total number of pixel points.

Figure 5.11, Figure 5.12, Figure 5.13 and Figure 5.14 show the approximated WSN coverage area and error for a deployment of 10 and 15 nodes. Initially the nodes were randomly deployed to a 20 × 20 m² area at the centre of the region of interest and their motion was coordinated by BNGRAZ. Under the latter the sensor nodes deploy until the distance between all sensor nodes is approximately equal to the communication range. Therefore sensor nodes physically relocate to maximise coverage (instantaneous and coverage against time). Figure 5.11 and Figure 5.12 show that, for a deployment of 10 and 15 nodes the approximated coverage area for each sensor node has a maximum error of approximately ±10% in both cases. However, the bulk of error comes from the shift in the position of the coverage region, as explained in the previous paragraph. Figure 5.13 and Figure 5.14 show that the associated coverage error does not exceed 20% for a deployment of 10 nodes and 40% for 15 nodes. These results support the statement that the CA algorithm is not scalable and the magnitude of the error depends on the position of the sensor node within the WSN. Nodes which are neither at the centre or periphery of the WSN will incur the largest coverage approximation error.
Despite the scalability issues and errors described above, the CA algorithm is used to approximate the WSN coverage; an alternative to the sensor nodes estimating the WSN coverage is to broadcast the coverage information through the network which carries high communication overheads. These errors assist the operation of BNGRAZ and help ensure that the sensor nodes successfully migrate around the region of interest. The motion coordination for BNGRAZ is decentralised and so sensor nodes do not collaborate with each other to decide the optimum motion direction. If all sensor nodes
had identical coverage maps, the motion direction which yields uncovered areas would vary depending on the nodes' position within network. Hence the sensor nodes may all attempt to move in different directions. This would be prevented by sensor nodes connectivity maintenance requirements (stay connected to neighbours), essentially resulting in the sensor nodes oscillating around their current position. If this occurs, the network would not achieve the desired coverage. The discrepancies in the sensor nodes' coverage maps help to minimise the motion oscillation.

5.4.3.3 Bayesian network 2 (Bn2): Probability of Discovering Un-serviced Area

The second Bayesian network (Bn2) calculates the probability of the sensor node discovering some area within the geographical region of interest that requires coverage by the WSN, given that the sensor node moves in one of the cardinal directions and given the coverage map (generated via the CA algorithm). This is referred to as discovering un-serviced area. The Bn2 structure is shown in Figure 5.15. The joint probability mass function for Bn2 is given in (5.19).

![Figure 5.15 Discovering uncovered area Bayesian network 2 structure](image)

The four parent nodes $U_N, U_S, U_E, U_W$ are evidence and represent the fraction of un-serviced area in each of the cardinal directions. The un-serviced area values are calculated by evaluating the local coverage map generated and maintained by the CA algorithm as discussed above in section 5.4.3.2.
The values assigned to $U_N, U_S, U_E, U_W$ are calculated by referencing the sensor node’s position with respect to the coverage map. A measure of ‘uncovered-ness’ in each of the cardinal directions is determined by evaluating the fraction of uncovered pixel points in each cardinal direction. This is calculated by taking the conjugate of the average pixel point weight. Figure 5.16 shows how the fraction of uncovered points is calculated for each cardinal direction. The region of interest is divided into four areas with respect to the sensor node’s position. The fraction of uncovered pixel points in areas 1 and 2 yield uncovered region in the north direction. Areas 1 and 3 yield the uncovered region in the west direction, 2 and 4 for the east direction, and 3 and 4 for the south direction.

The probability of discovering un-serviced area, given that the node move in cardinal direction $i$ ($P(D_i = T | \theta_m = i)$), is calculated by determining the joint probability mass function for $B_{n2}$ shown in (5.19) and marginalizing, which yields (5.20).

\[
P(D_{i} = T | \theta_m = i) = \frac{P(D_{i} = T, \theta_m = i)}{P(\theta_m = i)}
\]

\[
\sum_{\theta_m \in \{T,F\}} P(U_N)P(U_S)P(U_E)P(U_W)P(\theta_m = i) \times P(D_{i} = T | \theta_m = i, U_N, U_S, U_E, U_W)
\]

Figure 5.16 Fraction of un-covered pixel points in each cardinal direction
The probabilities generated by Bn1 and Bn2 are aggregated by Bn3 to determine the probability of maintaining/improving performance given that the sensor node moves in a particular direction. The output from Bn3 is used when selecting the motion status and direction of the sensor node. The latter process is outlined below.

5.4.3.4 Selecting the Motion Status and Direction

Each sensor node evaluates the probability of performance remaining the same or increasing \( P(O_d) \) in each cardinal direction, thereby establishing the general direction of motion. The predictions generated from the Bayesian networks only consider motion in each of the cardinal directions. Therefore, if the motion coordination is based purely on these predictions, the node would only move in the cardinal direction. The final bearing in the resulting cardinal direction (ranging between ±45 degrees) is determined by the difference between \( P(O_d) \) in adjacent cardinal directions. If the largest \( P(O_d) \) is greater than a specified Motion Decision Threshold (MDT), the node moves in the direction determined by Algorithm 5.5, otherwise it will remain static. The MDT defines the motion status of the sensor node by setting the sensitivity of the BNGRAZ algorithm through the responsiveness of sensor nodes to areas which require coverage. If this value is too low, the sensor nodes will continuously move and thus exhaust their limited energy reserve. Alternatively, setting the threshold too high will prevent the sensor node from moving.

**Algorithm 5.5**: Selecting motion direction

Calculate the difference between adjacent cardinal directions probabilities \( P(C_i ± 1) \)

if difference < 2% and difference > -2%

 motion direction = direction which yields maximum probability

else

 motion direction = direction which yields maximum probability + difference between adjacent cardinal direction probabilities

end
5.4.4 Simulation & Results

This section evaluates the performance of the three mobility algorithms, referred to as grazing algorithms. These mobility algorithms were compared to the RWP mobility model. In this work the speed was taken to be constant, to ensure consistency when comparing it to the other mobility algorithms. The RWP mobility model was implemented due to its simplicity and also because the sensor nodes have a direct roaming path between the source and destination coordinates. The model can also be tuned via the pause period time. The pause period can be set to reflect the desired coverage period (set by the network designer). This would reduce unwarranted motion and increase the operation lifetime of the WSN. This random model provided a mechanism to evaluate the performance of a mobile WSN where mobility is achieved by sensor nodes being attached to external mobile entities that move randomly through the region of interest.

The performance measures under consideration are connectivity, coverage, Coverage Against Time and lifetime. The mobile WSNs were simulated using the WSN simulator outlined in Chapter 3. The area of the geographical region of interest was set to 100 × 100 m². Sensor nodes (including the commander node) are homogeneous with equal capacity in terms of communication, processing, energy and sensing. The mobile sensor node platform is based on the X4e rover robots [89]. Table 5.2 details the settings of the simulation parameters.

The total coverage value outlined in Chapter 4 was set to 95% (total coverage is achieved when 95% of the region of interest is covered by the WSN). The sensor nodes are randomly deployed under a uniform distribution to a 20×20 m² area at the centre of the region of interest. The size of the sub-region which is applicable to GRPGM was set to 25×25 m². The pause time for RWP was set to 5 s.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Rationale</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication range</td>
<td>20 m</td>
<td>Typical ranges observed in external environments with Berkeley motes</td>
</tr>
<tr>
<td>Sensing range</td>
<td>10 m</td>
<td>Sensing range is half the communication range</td>
</tr>
<tr>
<td>Mobility cost</td>
<td>17.8758 J/m</td>
<td>X4e robot platforms [89]</td>
</tr>
<tr>
<td>Total initial energy</td>
<td>87480 J</td>
<td>Capacity of 6 v alkaline battery</td>
</tr>
</tbody>
</table>
The evaluation of the mobility algorithms outlined in previous sections is limited to the following parameters:

- Sensor node speed.
- Size of the mobile WSN (number of sensor nodes that form the network).
- The desired coverage period (generally set by the network designer).
- Size of the vicinity around the reference point (GRPGM only).
- Configuration duty time.
- The motion decision threshold value (BNGRAZ only).

These will be discussed in sections 5.4.4.1 to 5.4.4.6. The simulation results show the impact of varying each of the parameters in terms of coverage, connectivity and lifetime. The analysis of the mobile WSN performance in response to changes in the above parameters provides network designers with a design tool. As such, a network designer could use the desired WSN performance results and this analysis in conjunction to determine the value or range of values for each of the above parameters.

The following assumptions have been made:

- Mobile sensor nodes carry onboard GPS devices, therefore the nodes are location aware.
- The configuration only communicates location information to neighbouring node, such that they can update their CIT.
- Lifetime is the time until the commander node (mobile base station) fails. When considering the fixed path approach, it is assumed that the commander node is situated at the periphery of the network.
- The Medium Access Control (MAC) protocol is perfect. Therefore the practical implications associated with communication were not considered. Having said that, in a practical implementation the repeat mechanism built into the MAC protocol would ensure successful transmissions at the cost of negligible packet delay. The latter is of the order of hundreds of milliseconds (upper estimate) which would be negligible when compared with the duty time of the system at hand.
• The Sensor node speed under RWP is homogeneous. Therefore the random speed selection between min and max is omitted for comparison purposes.

Connectivity is a measure of a sensor nodes ability to communicate with the commander node (base station), possibly using multi-hop routing. The former is expressed as an average over the lifetime of the WSN. A decrease in connectivity may occur if sensor nodes fail or move outside the communication range of their neighbours. Coverage (instantaneous) represents the fraction of the total region covered at any point in time. The former is averaged over the lifetime of the WSN. The Coverage Against Time measure is used to generate a service interval distribution, see Chapter 4. The results show the mean service interval time and SDs for the service interval distribution associated with each simulation. The service interval distributions are given in Appendix G.

The assumed lifetime definition for this work is the time until the commander node fails. The latter may also be referred to as the time until the network fails to satisfy the performance criteria. In addition, the percentage of the total energy remaining after the failure of the commander node (lifetime) is evaluated. The total energy represents the sum of the sensor node’s individual energy capacity. The remaining energy may be defined as wasted energy. Therefore mechanisms may be incorporated to utilise this remaining energy, such as reassigning the responsibilities of the commander node to other sensor nodes. The remaining energy plots can also be used to evaluate energy balancing capabilities of each approach to mobility, i.e. balancing the EDR of each sensor node.

5.4.4.1 Sensor Node Speed

The sensor node speed controls the rate at which a sensor node physically changes position within the region of interest. This will invariably affect the rate at which the network topology changes and may alter the uncertainty associated with localisation and neighbouring node information between configuration instants.

The sensor node speed was varied from 0.0125 m/s to 0.1 m/s and the performance results in terms of coverage, connectivity and lifetime are given below in Figure 5.17 to
Figure 5.22. The number of nodes deployed was 15. The configuration duty time was set to 60 s and the size of the vicinity around the reference point was set to 50×50 m². The desired coverage period was set to 120 min and the MDT for BNGRAZ was arbitrarily set to 26%.

Figure 5.17 plots connectivity against the sensor node speed for each mobility algorithm. The sensor node speed does not affect the fixed path algorithm and a small variation in connectivity is observed with RWP and GRPGM. However the plot shows that for BNGRAZ, the increased node speed will reduce connectivity. A 20% decrease in connectivity is observed for a speed increase from 0.0125 to 0.1 m/s for the given parameters. Similar conclusions for coverage can be drawn from Figure 5.18 where coverage is evaluated against sensor node speed, i.e. the increased node speed reduces coverage (instantaneous) for the BNGRAZ algorithm. This is due to the increased uncertainty related to the neighbour information stored in the CIT. Under BNGRAZ, the sensor nodes use this information to reason about their motion directions. Therefore, there is an increased likelihood that a neighbour or the sensor node will move outside each others’ communication range.

The mean service interval time and SD for the variation in speed is given in Figure 5.19 and Figure 5.20. These plots show that the increased node speed reduces the service interval time and coverage period. This is because the sensor node can migrate faster around the network. The fixed path algorithm provides the fastest migration and so the best service interval and coverage period times. The mean service interval time for
GRPGM is the greatest for all node speeds. Compared to fixed path, at a speed of 0.0125 m/s the mean service interval time is 200% higher, whereas RWP and BNGRAZ is only 50% higher. This is due to the increased pause time whilst all sensor move to their random position around the reference point. As the speed increases, the times become comparable. The means for RWP and BNGRAZ are similar. Comparing the SDs given in Figure 5.20 show that the spread of the service interval times around the mean for RWP is the highest. However, the mean service interval time at a speed of 0.0125 m/s is half the value of GRPGM. The BNGRAZ algorithm provides the second best service interval distribution for all sensor node speeds.

Figure 5.19 Mean service interval vs. sensor node speed

Figure 5.20 Service interval SD vs. sensor node speed

Figure 5.21 plots the network lifetime against the sensor node speed. As the speed is increased, the network lifetime reduces. This is due to motion energy overheads increasing with speed. The network operating under GRPGM has the longest lifetime, at least 30% longer than the other algorithms (when compared to BNGRAZ). The fixed path and RWP has the shortest lifetime. As the speed is increased the variation in network lifetime becomes smaller. The percentage of the total remaining energy against the node speed is given in Figure 5.22. The fixed path algorithm has the highest percentage of remaining energy (wasted energy), which is constant at approximately 38%. This is due to the nodes at the outer periphery moving a greater distance resulting in those nodes becoming energy exhausted faster. The remaining energy for GRPGM is between 5 and 10%. BNGRAZ and RWP have minimal remaining energy meaning that the EDR for sensor nodes is balanced across the network.
5.4.4.2 Network Size

The network size is defined as the number of sensor nodes deployed across the region of interest. Figure 5.23 to Figure 5.28 show the WSN performance in terms of coverage, connectivity and lifetime under the four mobility algorithms whilst varying the number of sensor. The number of nodes deployed was increased from 9 to 30. The node speed was set to 0.1 m/s and the configuration duty time was set to 60 sec. The desired coverage period was set to 120 min. The vicinity around the reference point was set to 50×50 m². The number of nodes deployed consisted of 9, 15, 20, 25 and 30. These node quantities were selected to ensure that the number of nodes were incremented in stages, (not necessarily in equal increments).

Figure 5.23 plots the connectivity against the number of sensor nodes. The connectivity remains constant under the fixed path algorithm. The connectivity increases under GRPGM and RWP for an increased number of sensor nodes pointing to the fact that the network scales up well. Under these random based approaches (GRPGM and RWP), the increased number of nodes reduces the likelihood of the sensor node losing connectivity. This is due to there being a greater number of nodes within the region of interest and vicinity around the reference point for GRPGM. BNGRAZ causes a decrease in connectivity, with a deployment of 20 nodes resulting in the lowest connectivity. This confirms the scalability issues related to the CA algorithm, section 5.4.3.2.1. Performance predictions from a sensor node may contradict its neighbours.
and other nodes in the network, causing nodes to move into different directions thus resulting in loss in connectivity.

The instantaneous coverage increases with the size of the network due to the greater number of sensor nodes. The relation between coverage and network size is shown in Figure 5.24 for an increase in node number from 9 to 30. The fixed path algorithm gives the highest level of coverage increase from 20 to 70%, BNGRAZ from 15 to 50% and RWP from 8 to 40% respectively. For GRPGM 13% of the region is covered for 9 nodes and does not exceed 22%. This is due to the fact that the coverage for the GRPGM is controlled by the vicinity around the reference point which was set to 50×50 m².

Figure 5.23 Connectivity vs. number of sensor nodes

Figure 5.24 Coverage vs. number of sensor nodes

Figure 5.25 and Figure 5.26 evaluate the service interval and coverage period times with respect to the network size. As the number of nodes increase, the service interval times reduce. This is due to the increase in coverage and so nodes do not need to migrate as far to achieve total coverage. Once again, the fixed path algorithm achieves the shortest service intervals with the lowest mean and SD. GRPGM returns the longest service interval mean and with a SD between 6 and 9 min. This is due to the fixed vicinity around the reference point. The service intervals returned from RWP show the highest improvement as the network is scaled up. The service interval mean is 74 min faster when the number of nodes is increased from 9 to 30. In addition, the SD reduces by 7.5 min respectively. For a network size of 30 nodes, the service interval distribution
returned from RWP gives a mean 3 min higher than the fixed path. The service interval distributions returned from BNGRAZ give the worst distribution. As the number of nodes increase, the mean times reduce due to the increase in coverage (instantaneous). However, the SD increases, thus indicating that the spread of the distribution is higher, resulting in larger coverage periods. This confirms the scaling issues associated with the CA algorithm and so the uncertainty associated with the nodes’ coverage maps (areas which have not been covered by the WSN) increases.

![Mean service interval vs. number of sensor nodes](image1)

![Service interval SD vs. number of sensor nodes](image2)

The lifetime of the network operating under the fixed path, RWP, and BNGRAZ does not change as the network size increases, which is shown in Figure 5.27. The lifetime of GRPGM increases due to the extended pause period whilst all sensor nodes reach their positions. Figure 5.28 shows that the percentage of remaining energy remains constant for RWP and BNGRAZ. GRPGM shows some variation in remaining energy but this is due to the random aspects of the network (varying pause times). The decrease in remaining energy observed with the fixed path approach is due to the deployment configuration and roaming path i.e. the additional sensor nodes operate close to the periphery and thus have a high EDR.
5.4.4.3 Desired Coverage Period

The desired coverage period would generally be set by the application designer and refers to the desired time period at which total coverage is achieved. The desired coverage period controls the decrement rate of the coverage maps applicable to GRPGM and BNGRAZ. The performance of the network under fixed path and RWP is not influenced by this value because it only controls the decrement rate of the pixel point coverage weight associated with the coverage map, only used by GRPGM and BNGRAZ. Figure 5.29 to Figure 5.34 evaluate the performance of GRPGM and BNGRAZ with respect to changing the desired coverage period. The network consisted of 9 nodes moving at a speed of 0.05 m/s. The configuration duty time was set to 60 s and the vicinity around the reference point was set to 30×30 m².

Figure 5.29 and Figure 5.30 are plots of the connectivity and instantaneous coverage against the desired coverage period which was increased from 120 to 480 min. For BNGRAZ the connectivity remains constant up until a coverage period of 360 min. At this point, the connectivity decreases and the coverage increases. This is due to the network covering all areas of the region of interest and therefore the sensor nodes seem to spread out to maximise coverage, at the cost of connectivity. With GRPGM the coverage remains constant due to the fixed vicinity around the reference point. The decrease in connectivity may be due to the random positioning around the reference point.
The mean and SD for the service interval distributions against desired coverage period are plotted in Figure 5.31 and Figure 5.32. The mean service interval remains reasonably constant for BNGRAZ, varying between 130 and 134 min. The mean for GRPGM decreases at a desired coverage period of 240 min. Furthermore, there is also a decrease in connectivity for the same value, shown in Figure 5.29. This is due to the random nature of the algorithm. The SD reduces as the desired coverage period increases. This indicates that the network (mobility algorithm) finds it easier to identify regions which have not been covered when the coverage map decrements more slowly. Thus sensor nodes do not waste energy or time moving to regions which do not require coverage.
The plots of lifetime and remaining energy against the desired coverage periods are given in Figure 5.33 and Figure 5.34. The lifetime and remaining energy of the network which runs BNGRAZ do not change with respect to the desired coverage period, because the sensor node motion status is explicitly controlled by the MDT. There is some variation in lifetime and remaining energy for GRPGM but this is due to the random nature of the algorithm.

5.4.4.4 Vicinity Around Reference Point – GRPGM only

The vicinity around the reference point is only applicable to the GRPGM algorithm. The value sets the size of the region around the reference point. For example, a vicinity size of 30 m indicates an area of 30×30 m², with the reference point at the centre. The sensor nodes select a random position to physically relocate to within this area from a uniform distribution. Figure 5.35 to Figure 5.40 evaluate the performance of GRPGM against the size of the vicinity around the reference point. The plots consider a deployment of 9 and 15 nodes moving at a speed of 0.1 m/s. The desired coverage period was set to 120 min. Figure 5.35 and Figure 5.36 evaluate the coverage and connectivity against the size of the vicinity. As the size of the vicinity increases the sensor nodes have a larger area in which to randomly position themselves therefore increasing the chances of the nodes uniformly distributing themselves around the reference point. The random positions may result in nodes positioning themselves on top of each other or outside each others’ communication range. The net effect of this is
that connectivity decreases and coverage increases with the vicinity size. The results
show that the number of nodes and vicinity size should be selected in unison to ensure
that the network satisfies the performance requirements.

The service interval mean and SD against the vicinity size is given in Figure 5.37 and
Figure 5.38. The increase in the size of the vicinity results in a larger mean service
interval time and a larger SD. Therefore the service interval coverage period times
increase. This is due to the reduced connectivity. However, there is trade-off between
connectivity and instantaneous coverage and therefore the full effect of the reduced
connectivity is counteracted by the increase in coverage.
The lifetime and percentage of remaining energy increase with the size of the vicinity, shown in Figure 5.39 and Figure 5.40. This is due to the increased area where nodes can physically reposition resulting in nodes moving a greater distance. This results in an increased pause period whilst all nodes reach their position and individual nodes exhaust more energy, resulting in poor EDR balancing among the sensor nodes.

5.4.4.5 Configuration Duty Time – BNGRAZ only

The configuration duty time defines the time period between neighbour communications. The latter updates the neighbour information in the Contacts Information Tables (CITs). BNGRAZ is the only algorithm to use the CIT for performance predictions to decide the direction to move into. As the sensor node localisation is achieved using GPS, the configuration duty time does not influence the behaviour of fixed path, RWP, and GRPGM algorithms. Figure 5.41 to Figure 5.46 show the performance of BNGRAZ for a configuration duty time between 0 (sensor nodes are always aware of its neighbours position) and 120 s. The plots consider a network size of 9 and 15 nodes moving at a speed of 0.05 and 0.1 m/s. The desired coverage period was set to 120 min.

Figure 5.41 shows that as the configuration time is increased the connectivity reduces. The rate at which the connectivity reduces is dependent on the node speed. The slower a node moves the less distance its neighbours and itself will move before the CIT is updated. Therefore this reduces the uncertainty associated with neighbour positions between configuration times. Therefore the sensor nodes are less likely to move outside...
each others’ communication range. The uncertainty associated with neighbour information also influences the coverage as shown in Figure 5.42. The figure shows that a mobile WSN with sensor nodes that move faster yield a higher value of instantaneous coverage when the configuration duty time is 0. However, as the configuration duty time is increased to 120 s the mobile WSN with sensor nodes that move slower yield a higher level of coverage. Indicating that the WSN with sensor nodes that move faster are more likely to move closer together or further apart (leading to a loss in connectivity) thus reducing coverage (instantaneous).

The service interval times are also affected by the configuration duty times. This is shown via the mean and SD plots given in Figure 5.43 and Figure 5.44. The mean service interval time, as shown in Figure 5.43, seems to increase with the uncertainty associated with neighbour information. This is due to the reduction in instantaneous coverage, observed when the configuration duty time is increased. The service interval SD for nodes moving at a speed of 0.1 m/s varies by 2.5 min. For a node speed of 0.05 m/s and a deployment of 9 and 15 nodes, the SD varies by approximately 2 and 4 min respectively. The variation in service interval times is a result of increasing uncertainty about neighbour information and areas which have not been covered by the network. The latter is determined by the CA algorithm. As uncertainty associated with the neighbour information increases, the probability of the BNGRAZ returning accurate performance predictions reduces. The latter results in the sensor nodes antagonising
each other to remain connected and the sensor nodes may also move to areas which are believed to have not been covered, when in fact the areas have already been covered.

In terms of lifetime, the reduced communication energy overhead associated with increasing the configuration duty time is minimal when compared to the motion energy overhead. Figure 5.45 shows that motion energy overheads are unaffected by the increase in configuration duty time. In addition, there is only a slight increase in the percentage of remaining energy for the increased configuration duty time, shown in Figure 5.46. However, the largest increase is 0.005% for 9 nodes moving at a speed of 0.05 m/s.
The following plots evaluate the ability of a mobile WSN operating under the BNGRAZ algorithm to maintain performance (in terms of coverage, connectivity and lifetime) when the configuration duty time is increased to 5 min. The desired coverage period was set to 120 min. The results consider two arbitrarily selected cases. These are a deployment of 9 and 15 nodes travelling at a speed of 0.05 and 0.1 m/s respectively. Figure 5.47 and Figure 5.48 show that increasing the configuration duty time to 5 min results in the connectivity and coverage reducing. The results show that for the deployment of 15 nodes travelling at a speed of 0.1 m/s the connectivity drops below 90% and the instantaneous coverage decreases by 5% when the configuration duty time is greater than 50 s. However for the deployment of 9 nodes travelling at a speed of 0.05 m/s the configuration duty time is greater than 200 s before a similar decrease in connectivity and coverage is observed.

Figure 5.47 BNGRAZ connectivity vs. configuration duty time

Figure 5.48 BNGRAZ coverage vs. configuration duty time

Figure 5.49 and Figure 5.50 plot the mean and SD from the service interval distributions associated with the increasing configuration duty time. The increased uncertainty associated with the neighbour information effectively results in the service interval times increasing. The service interval mean for a deployment of 15 nodes travelling at a speed of 0.1 m/s only increases by 25 min and the SD increases by approximately 3 min when the configuration duty time is extended from 0 to 300 s. For a deployment of 9 nodes and a speed of 0.05 m/s the same variation in the configuration duty time results in the mean and SD increasing by 55 min and 2 min respectively. These results show the limits of the BNGRAZ algorithm, in terms of the extending
configuration duty time, associated with the coverage and connectivity performance criteria.

5.4.4.6 BNGRAZ Motion Decision Threshold (MDT) – BNGRAZ only

The MDT, see section 5.4.3.4, explicitly controls whether a sensor node will move or not whilst operating under BNGRAZ. Essentially a sensor node will only move if the probability of performance increasing or remaining constant is greater than or equal to the MDT. The value of the MDT is “inversely proportional” to the sensitivity (in terms of mobility) of the sensor node to performance predictions. Figure 5.51 to Figure 5.56 shows the change in network performance whilst increasing the MDT value from 25% to 27.4%. A MDT value of 25% was selected from evaluating the CPT values in Appendix B. It is the minimum value which causes the sensor nodes to constantly move. The MDT was increased until the mobile WSN operating under the BNGRAZ algorithm failed to meet the desired performance criteria, in terms of coverage (desired coverage period) and/or connectivity. The number of nodes deployed was 15 and their speed was set at 0.05 m/s. The configuration duty time was set to 60 s and the desired coverage period was set to 120 min.

Figure 5.51 and Figure 5.52 show that increasing the MDT results in the connectivity decreasing and the coverage marginally decreasing. This is due to some of the sensor nodes remaining static because the probability of performance increasing or remaining constant does not exceed the MDT.
The reduction in network mobility due to an increase in MDT results in larger service interval and coverage period times. This is reflected in Figure 5.53 and Figure 5.54 by the mean and SD from the service interval distributions increasing.

In turn, the reduced sensor node mobility extends the lifetime of the network. Figure 5.55 shows that increasing the MDT from 25% to 27.4% results in the network lifetime increasing by approximately 350 min. The reduced mobility also results in the percentage of total energy remaining increasing (deemed as wasted energy due to poor energy overhead balancing), see Figure 5.56.
These results show that the lifetime can be extended to approximately 2000 min, by setting the MDT to 26.6%, without connectivity and coverage decreasing by 10% and 5% respectively (see Figure 5.51 and Figure 5.52). Given this MDT value the mean service interval time is still within the desired coverage period (120 min), this is without considering the service interval SD.

**Figure 5.55 BNGRAZ lifetime vs. MDT**

**Figure 5.56 BNGRAZ remaining energy vs. MDT**

### 5.4.4.7 Discussion

The fixed path algorithm achieves the smallest service interval and coverage period times. However, the lifetime of the network is the lowest and the percentage of remaining energy is the highest (therefore the highest wasted energy). The performance of the fixed path algorithm is dependent on the defined roaming paths set by the application designer, based on the information about the application and region of interest. In addition, the algorithm is not fault tolerant and so the failure of a single node will severely deteriorate the performance, as the area which was covered by that WSN will no longer be covered. This will prevent the network from satisfying the application criteria and lead to a reduced network lifetime.

The RWP mobility model does not guarantee connectivity and sensor nodes can become disconnected from the base station for prolonged periods. Therefore sensor nodes may need to store sensing data during the periods of being disconnected to ensure coverage. In addition, the disconnection period is not bounded and therefore the sensing data maybe cleared from memory before the sensor node becomes re-connected.
This will also increase the latency of data delivery. The connection period must also be sufficient to ensure that all data can be successfully transmitted to the base station. Furthermore, there is no guarantee that the sensor nodes will collectively migrate to achieve total coverage and sensor nodes may position themselves such that their coverage regions overlap. These aspects must be considered when evaluating the performance of the algorithm.

The performance of GRPGM is influenced by the number of nodes and the size of the vicinity around the reference point. Therefore the parameters should be tuned to achieve the desired performance. The nature of the algorithm means that there is no guarantee that the distribution of the nodes will be uniform. Therefore nodes may position themselves such that they are effectively on top of each other or in the worst case scenario become disconnected. However, GRPGM provides the longest lifetime at the cost of service interval and coverage period times. Compared to the other algorithms the decrease in performance was the highest for a reduced node speed. In addition increasing the number of sensor nodes has minimal effect unless the vicinity size is increased. In the event of node failures there is still a likelihood of the network satisfying the application criteria (that is providing total coverage). However there is an increased likelihood of nodes positioning themselves such that they become disconnected.

The performance of the BNGRAZ algorithm is dependent upon the configuration duty time (uncertainty associated with neighbourhood information) and the speed of the sensor nodes. Therefore, the speed and configuration time must be selected in unison to ensure the WSN remains stable. The results presented show that increasing the configuration time results in sensor nodes becoming disconnected. The results showed that the BNGRAZ algorithm does not scale well due to the approximation from the CA algorithm. However, it out performs GRPGM when the node speed is reduced. In addition, the results showed that the lifetime of the network could be increased by varying the MDT. However this reduces service interval and coverage period times. The algorithm is fault tolerant and will still satisfy the application criteria in the event of timely or abrupt node failures.

Effectively, GRPGM, BNGRAZ and RWP are still capable of operating (i.e. providing total coverage) if all nodes fail except the commander node. The selected motion algorithm will invariably depend on the application criteria.
5.5 Summary

This Chapter has presented four alternative mobility algorithms for WSNs. Adopting a mobile WSN is based on the principle that the size of the network (number of sensor nodes) is not sufficient to provide total coverage at each instant in time. Therefore the sensor nodes migrate to provide coverage over time. BNGRAZ and GRPGM adopt a new grazing strategy which enables sensor nodes to evaluate which areas of the region of interest require coverage, in the case of GRPGM this calculation is carried out by the commander node only. The aim of these algorithms is to physically relocate sensor nodes such that the nodes remain connected and satisfy the coverage requirements.

The decentralised CA algorithm has been presented, see Section 5.4.3.2. This algorithm approximates the WSN coverage using only local neighbour information. The approximation is used to generate the evidence for BNGRAZ. The evaluation of this algorithm showed that it does not scale well, however its operation carries no additional communication overheads and provides nodes with information which reduces the level of uncertainty associated with determining the overall WSN coverage.

The limitations of BNGRAZ in terms of configuration duty time and MDT have been investigated, and in particular in terms of their impact on coverage, connectivity and lifetime. Increasing the configuration duty time resulted in a decrease in connectivity and instantaneous coverage which led to an increase in service interval times, effectively a deterioration in performance. Similarly, increasing the MDT has resulted in a loss of connectivity and reduced coverage; this is due to some sensor nodes moving less. However, on the positive side the impact of this reduction in motion results in an increased network lifetime.

The results generated from the simulations can be used to compare the performance of mobile WSNs operating under particular mobility algorithms, as well as highlighting their benefits/shortcomings. The selection of the optimum algorithm would invariably depend on the application criteria. These simulations are essentially a design tool which, albeit graphical, can help design a mobile WSN for a particular purpose.
Chapter 6

SELF-HEALING MOTION

6.1 Introduction

This chapter describes the self-healing motion strategy for WSNs. The reasons for adopting this type of behaviour are discussed, as well as the inherent performance benefits that are drawn from the technique. A novel Bayesian network motion (BayesMob) algorithm is outlined which enables a sensor node to predict the change in WSN coverage related to a sensor node moving in a given direction. These coverage predictions have imparted self-healing capabilities to the mobile WSN. Simulation results compare BayesMob to CoFi (see Section 2.4.3.5) and a static WSN.

6.2 The Principle of Self-healing

In a number of applications, sensor nodes that form a WSN are prone to abrupt failures. The latter may result from energy exhaustion (depleted batteries), malicious destruction or malfunction. Sensor nodes are generally powered by onboard batteries and depending on the proposed application, it may not be feasible to replace or replenish these batteries during the operation of the WSN. Hence, these batteries must provide sufficient energy to the sensor node throughout the lifetime of the WSN. The energy overheads related to communication may cause a single node to become energy exhausted. This may be due to varying traffic characteristics, resulting in a non-uniform network topology. Other factors such as edge effects (sensor nodes in the centre of the network are likely to die faster as they lie on more data forwarding routes) also make the energy distribution non-uniform. Therefore the aim of the network designer is to ensure that the EDRs of the sensor nodes are balanced.

The WSN may also be deployed into inhospitable regions with extreme environmental conditions. For example, sudden changes in the climate (extreme heat or flooding) may cause sensor nodes to malfunction. Another reason for failure is exposure to humans and animals which may result in malicious destruction.
The cumulative effect of the above factors (the list is by no means exhaustive) often results in sensor node failures and scenarios where a segment of the network runs out of energy before the rest of the WSN. This will result in the creation of a coverage hole where a subsection of the WSN becomes disconnected or a proportion of the region of interest not being covered by the WSN. If a section of the WSN becomes non-functional through the creation of the coverage hole, the system may not be able to meet the performance criteria. Hence, the WSN may be rendered useless and the remaining resources (energy within the remaining sensor nodes) within the network would be wasted.

If some of the remaining resources can be transferred to the coverage hole through the physical relocation of the sensor nodes, the wasted resources could be reduced and thus the WSN lifetime increased. This type of behaviour is generally referred to as self-healing. Figure 6.1 illustrates the self-healing in a WSN. The self-healing process will invariably incur additional energy overheads, due to driving motors and servos. Therefore, a motion algorithm is required to coordinate the relocation of the sensor nodes whilst considering the energy overheads.

Self-healing may be achieved by the movement of the neighbouring nodes of the failing node or through cascaded neighbour movements. Cascaded movements are when more than one hop neighbours of the failing node move to repair the coverage hole. The number of nodes that physically relocate depends on the level of redundancy. If the WSN contains a high level of redundancy, self-healing may be achieved by the movement of one-hop neighbours. However, if redundancy is low, cascaded movement of nodes may be required.
The following sections introduce and outline the new Bayesian network Mobility (BayesMob) self-healing algorithm created as part of this work. BayesMob coordinates the sensor node relocation to maintain coverage in the event of coverage holes.

6.3 Bayesian network Mobility (BayesMob)

BayesMob is a distributed self-healing motion control algorithm for WSNs created as part of the work presented here. It uses a Bayesian network to determine the probability of coverage (in terms of instantaneous coverage) increasing or remaining unchanged given that the sensor node under consideration moves in a particular direction. These coverage predictions are based on a possible move in one of the cardinal directions north, south, east, or west. A mobile sensor node then uses these predictions to determine whether motion would yield an increase in coverage and ultimately the motion direction. These coverage predictions have effectively imparted a self-healing behaviour by enabling a mobile sensor node to implicitly determine whether a coverage hole has been created.

The evidence fed into the Bayesian network is acquired through local neighbourhood information (essentially the position of one-hop neighbours). Each of the mobile sensor nodes that form part of the WSN would be capable of executing the BayesMob algorithm. Note that a WSN may be heterogeneous, where only a proportion of the sensor nodes contain mobile capabilities. The implication associated with the limited number of mobile nodes is that the useful lifetime of the network may be reduced when compared to a homogeneous WSN (all sensor nodes have motion capabilities). This is due to a fewer number of sensor nodes moving, resulting in a smaller number of coverage holes being repaired, thus adversely affecting the performance of the WSN.

The Bayesian network structure for predicting the probability of an increase in coverage is shown in Figure 6.2. The structure for BayesMob is similar to BnI which is part of BNGRAZ (section 5.4.3.1). The two algorithms differ in that $C_i$ represents coverage increase and not connectivity decrease. The definition of all parameters used is outlined in Table 6.1. BayesMob only considers discrete variables so the relationships between connected nodes are represented by a Conditional Probability Table (CPT), see Appendix E. The Bayesian network and CPTs were generated using Netica application software from Norsys [95] and the procedure described for
BNGRAZ, see section 5.4.3. The CPT values within BayesMob have been specified to ensure the mobile sensor nodes can identify a coverage hole and physically relocate to repair it.

![Bayesian network structure](image)

**Figure 6.2 Coverage increase Bayesian network structure**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>True</td>
</tr>
<tr>
<td>$F$</td>
<td>False</td>
</tr>
<tr>
<td>$N$</td>
<td>Cardinal direction north (45 to 135 degrees)</td>
</tr>
<tr>
<td>$S$</td>
<td>Cardinal direction south (225 to 315 degrees)</td>
</tr>
<tr>
<td>$E$</td>
<td>Cardinal direction east (45 to 45 degrees)</td>
</tr>
<tr>
<td>$W$</td>
<td>Cardinal direction west (135 to 215 degrees)</td>
</tr>
<tr>
<td>$I$</td>
<td>Cardinal direction indices (N,S,E,W)</td>
</tr>
<tr>
<td>$C_r$</td>
<td>Sensor node communication range</td>
</tr>
<tr>
<td>$Q_{i\epsilon}$</td>
<td>Coverage increase in cardinal direction $i \in {N,S,E,W}$ (True, False)</td>
</tr>
<tr>
<td>$N_{i\epsilon}$</td>
<td>Need to move in cardinal direction $i \in {N,S,E,W}$ (True, False)</td>
</tr>
<tr>
<td>$A_{i\epsilon}$</td>
<td>At least one neighbour is lying in cardinal direction $i \in {N,S,E,W}$ (True, False)</td>
</tr>
<tr>
<td>$d_{i\epsilon}$</td>
<td>At least one neighbour's distance $&lt; C_r$ in cardinal direction $i \in {N,S,E,W}$ (True, False)</td>
</tr>
<tr>
<td>$\Theta_{m}$</td>
<td>Motion angle of sensor node $m \in {N,S,E,W}$</td>
</tr>
</tbody>
</table>

The evidence required for predicting the probability of an increase in coverage is based on the position of neighbours. It is expressed as the probability of a neighbour lying in each one of the cardinal directions and probability that the closest neighbour's distance in each direction is less than the communication range ($C_r$). The belief in the evidence
$A_i$ and $d_i$ (see Table 6.1) is calculated from the CIT. Sections 5.4.3.1.1, 5.4.3.1.2 and 5.4.3.1.3 outline the procedure for calculating $d_i$ and $A_i$ respectively.

The joint probability mass function for BayesMob is given in (6.1). The probability of coverage increasing, given that the node moves in direction $i$ ($P(C_i = T \mid \theta_m = i)$), is calculated by using the joint probability mass function and marginalization, which is given in (6.2).

\[
P(C_i, \theta_m, N_N, N_S, N_E, N_W) = P(\theta_m)P(N_N)P(N_S)P(N_E)P(N_W)P(C_i \mid \theta_m, N_N, N_S, N_E, N_W)
\]  

(6.1)

Using the joint probability mass function and marginalizing

\[
P(C_i = T \mid \theta_m = i) = \frac{P(C_i = T, \theta_m = i)}{P(\theta_m = i)} = \sum_{N_N, N_S, N_E, N_W} P(N_N)P(N_S)P(N_E)P(N_W)P(\theta_m = i) \times P(C_i = T \mid \theta_m = i, N_N, N_S, N_E, N_W)
\]

(6.2)

where $P(N_N)$ to $P(N_W)$ (probability of the need to move into each of the cardinal directions) are calculated using (6.3).

\[
P(N_i = T) = \sum_{A_i, d_i \in [r, f]} P(A_i)P(d_i)P(N_i = T \mid A_i, d_i)
\]

(6.3)

Each sensor node evaluates the probability of the coverage increasing $P(C_i)$ in each cardinal direction, thereby establishing the need for motion and the general direction of motion. The final bearing in the resulting cardinal direction (ranging between ±45 degrees) is determined by the difference between $P(C_i)$ in adjacent cardinal directions.

If $P(C_i)$ is greater than a specified MDT the node moves in the direction determined by Algorithm 5.5, see Chapter 5. Algorithm 5.4 enables the sensor nodes to evaluate the predictions in all cardinal directions and move into a direction which considers all these predictions. The MDT defines the sensitivity of the sensor node. If this value is too low, the sensor node will continuously move and thus exhaust its limited energy reserve.

Setting the MDT too high will prevent the sensor node from moving to repair coverage holes. The MDT was set to 35% through trial and error testing.
6.4 Simulation & Results

All simulations have been generated using the WSN simulator. The geographical region of interest was set to a 100x100 m² area. Every sensor node is equipped with motion capabilities. The simulation parameters are the same as Chapter 5, see Table 5.2. The configuration duty time and the generation and transmission of data packets destined for the commander node was set to 2 min. In a real application, these may be offset to ensure they do not occur at the same time. Again a perfect MAC (Medium Access Control) protocol was assumed yielding faultless communication.

Two types of deployment strategies have been tested; fixed and random deployments. Under the fixed approach the sensor nodes are placed manually, thereby ensuring a uniform distribution of the sensor nodes and maximum coverage. The random approach positions the nodes following a uniform distribution. Deployment under this approach may be achieved by an air drop.

Simulations have been carried out for a range of sensor node densities (64, 81 and 100 nodes). The node density values were selected to provide a uniform spatial distribution across the region of interest (which is square shaped), whilst adopting a fixed deployment approach. For example, a deployment of 64 nodes which are evenly distributed across the region of interest yields an 8 by 8 node grid. The simulations evaluate the WSN coverage for the CoFi algorithm (see Section 2.4.3.5), static network and the BayesMob algorithm. The CoFi algorithm is an alternative self-healing algorithm for mobile WSNs and so a comparison between two alternative mobility algorithms is given. The static network consists of sensor nodes that do not physically relocate to repair coverage holes and so redundancy would be used to prevent coverage holes. Including a static network in the simulations provides a base line for validating the self-healing motion algorithms, effectively the worst case scenario. The three approaches are compared in terms of coverage loss and motion energy overheads. Node failures are induced using a uniform distribution and were made to occur at 30 min intervals in an attempt to replicate abrupt failures under variable WSN environmental conditions. The simulations assume that the dying sensor nodes operating under CoFi have sufficient time to coordinate the relocation of neighbours. This may not be the case in real application as sudden failures can disable the node instantly.
6.4.1 Fixed Deployment

First a fixed deployment scenario was considered, under which the sensor nodes were uniformly distributed over the region of interest to provide 100% coverage. Figure 6.3, Figure 6.4 and Figure 6.5 show the respective coverage loss plots for a deployment of 64, 81, and 100 nodes. These results show that the physical relocation of the sensor nodes under the BayesMob algorithm will sustain the WSN coverage for an increased percentage of node failures. This observation becomes more apparent as the sensor node density increases. For example, if the application criterion specifies a maximum coverage loss of 20%, in the case of the CoFi algorithm the WSN tolerates 35%, 43%, and 44% of nodes failing for 64, 81, and 100 nodes deployed respectively. Using the BayesMob algorithm the WSN tolerates 45, 58, and 68 percent nodes failing for the same node densities. As expected, both the CoFi and BayesMob algorithms sustain coverage for a greater percentage of node failures than the static WSN. However, for a deployment of 64 nodes, CoFi does not physically relocate sensor nodes and performs the same as a static WSN. Therefore the coverage loss curve shown in Figure 6.3 is the same for CoFi and static WSN. This is due to the low node density, and the CoFi algorithm will not move nodes at the cost of coverage.

The sharp increases in coverage loss observed in Figures 6.3 to 6.9 are due to a subsection of the WSN becoming disconnected (loss in connectivity). The algorithms fail to recover from the coverage loss when the number of sensor nodes deployed is not sufficient to cover the region of interest.
6.4.2 Random Deployment

Here the sensor nodes are deployed randomly according to a uniform distribution over the region of interest. Figure 6.6, Figure 6.7 and Figure 6.8 show the respective coverage loss plots. The results of the simulations show that the BayesMob algorithm tolerates 50%, 60% and 68% of nodes failing (for 64, 81 and 100 nodes deployed, respectively) before the coverage loss criterion (maximum coverage loss of 20%) is
exceeded. Under the same conditions, CoFi tolerates only 40%, 43% and 57% node failure before the coverage loss criterion is exceeded.

Figure 6.6 Coverage loss plot for a random deployment of 64 nodes

Figure 6.7 Coverage loss plot for a random deployment of 81 nodes

Figure 6.8 Coverage loss plot for a random deployment of 100 nodes

Figure 6.9 Motion energy overhead plot for CoFi vs. BayesMob
6.4.3 Energy Considerations

The percentage of total energy available to the WSN used for motion for both strategies (BayesMob and CoFi) has been evaluated and is presented (for a random deployment of 81 nodes) in Figure 6.9 as a function of the percentage of dead nodes. The static configuration is omitted, as there is no motion energy overheads associated with this approach. The total energy is calculated by multiplying the total number of sensor nodes by their individual energy reserves (total battery capacity) throughout the length of the simulation. The motion energy costs associated with the BayesMob exceed those of CoFi. Figure 6.9 shows that when 50% of the nodes fail the motion energy for BayesMob algorithm is approximately 0.2% of the total energy. The corresponding motion energy for the CoFi algorithm under the same conditions is approximately 0.025%. However, Figure 6.7 shows that when this level of node fails the coverage loss for the mobile WSN which incorporates the CoFi algorithm is greater than the 20% criterion. The coverage loss for the mobile WSN which incorporates BayesMob is approximately 11%. Indicating that the increased energy overhead associated with the BayesMob algorithm is justified when compared to the benefits in terms of the reduced coverage loss. As mentioned early BayesMob can tolerate up to 60% of nodes failing before the 20% coverage criterion is exceeded, at the cost of 0.5% of the total energy.

The motion energy for BayesMob rapidly increases as the percentage of nodes fail. This is due to sensor nodes moving a greater distance to repair the coverage holes. The CoFi algorithm will only attempt to repair coverage holes by relocating one-hop neighbours. Therefore, the motion distance is restricted at the cost of coverage. The decentralized nature of BayesMob means that it can also accommodate abrupt failures due to malicious destruction or malfunction.

6.5 Summary

Through the use of a Bayesian network based mobility scheme, a WSN has been imparted self-healing properties with regards to coverage losses induced by abrupt node failures. BayesMob has been shown to help sustain coverage in a WSN well beyond the capabilities of CoFi and a similar (density, operation) static network. Sensor nodes operating according to BayesMob, predict the WSN coverage variations using local information and decide where to move with a view to maximize or maintain the
coverage of the network. The simulations carried out have shown quantitatively that the WSN’s performance is enhanced.

The additional benefits of BayesMob over the CoFi algorithm is that physical relocation is not limited to one-hop neighbours and also the fact that multiple neighbours may physically reposition in response to a failure. This ensures a prolonged period of coverage and also motion energy overhead balancing between all neighbouring nodes. The CoFi algorithm is limited to failures resulting from energy exhaustion due to the dying node coordinates the relocation of neighbouring nodes. Therefore if an abrupt failure occurs, the sensor node would not be able to coordinate mobility. BayesMob can accommodate any type of node failure, due to individual nodes coordinating their own mobility. To ensure a fair comparison the simulation results presented above only considered node failures due to energy exhaustions. If abrupt failures were also considered the coverage loss associated with node failures for the CoFi algorithm would increase thus increasing the coverage loss benefits associated with the BayesMob algorithm.
Chapter 7
CONCLUSION AND FUTURE WORK

7.1 Conclusions

This thesis has explored mobility strategies and algorithms to improve the performance of mobile WSNs. The performance aspects considered here were coverage, connectivity and lifetime. As part of the work presented, and for the first time, Bayesian networks have been used to coordinate WSN sensor nodes' mobility. The Bayesian networks enable a sensor node to decide on a direction of motion by predicting the probability of performance improving/changing. The Bayesian Networks require evidence for deriving the performance predictions, which is obtained from the local neighbourhood topology and possibly a coverage approximation (derived from the CA algorithm), therefore keeping additional operation overheads to a minimum. The information relating to neighbours and contacts are stored using a novel mechanism, referred to as Contacts Information Table (CIT). The CIT stores the position of each contact as a mean and standard deviation and provides a mechanism for managing the uncertainty associated with the contacts' positions between contact communications (updating the CIT).

A novel biologically inspired mobility strategy has been presented. The mobility strategy ensures total coverage over time when the size of the mobile WSN (number of sensor nodes) is not sufficient to cover the entire region of interest at any given point in time. The strategy was inspired from herbivores grazing pastures where the area of interest is analogous to pasture and the grass analogous to information.

A new Coverage Against Time performance measure has also been presented, which provides a metric to evaluate the coverage of mobile WSN. The traditional coverage measure returns the fraction of the intended region of interest collectively covered by the WSN at a given point in time. For some mobile WSNs, particularly when the size of the WSN is not sufficient to provide total coverage, the traditional coverage measure will not evaluate coverage in a useful manner. The new coverage measure returns two metrics which consist of the coverage period (time taken by the network to achieve
total coverage) and the service interval for individual points within the geographical region of interest (the time period between the network being deployed and the point being initially covered and the time period between the points being covered in consecutive coverage periods). Therefore this measure provides an effective mechanism for measuring the time associated with covering the region of interest.

Two new mobility algorithms have been presented which adopt the grazing strategy behaviour namely, the centralised GRPGM algorithm and the decentralised Bayesian network based algorithm BNGRAZ. Under GRPGM, the migration of the sensor nodes is coordinated by a commander node (mobile base station) broadcasting control signal in the form of reference points to the sensor nodes. The reference points are derived from sensor nodes positions which are encapsulated in the data packet headers. Under BNGRAZ, each sensor node uses a set of Bayesian networks to predict the probability of coverage and connectivity increasing or remaining constant, given that a sensor node moves in a particular direction.

The performance of the GRPGM and BNGRAZ algorithms was evaluated by comparison to a fixed roaming path approach and a random mobility algorithm (RWP). The results showed that the fixed approach provides the best performance in terms of coverage. However the approach is susceptible to poor energy balancing and node failures. Under the RWP algorithm there is no guarantee that the sensor node will remain connected and therefore the approach provides the worst performance. GRPGM and BNGRAZ achieved similar coverage against time performance but GRPGM maintains lower (instantaneous) coverage. BNGRAZ provides the highest level of fault tolerance and will satisfy the coverage criteria even in the event of timely and abrupt node failures. Under GRPGM failures may result in sub-sections of the WSN being disconnected due to nodes randomly selecting positions within the vicinity of the reference point. The evaluation described above provides a design tool which, albeit graphical, can help design a mobile WSN for a particular purpose.

This thesis has also presented a new Coverage Approximation (CA) algorithm for WSN. This algorithm enables each sensor node to approximate the WSN coverage via neighbour and contact information stored in the nodes' CIT. This information is obtained from network configuration and data packets which the sensor node forwards to the base station. Therefore, the algorithm carries no additional communication overheads. The coverage approximation enables sensor nodes to generate and maintain
a coverage map of the region of interest. This coverage map provides evidence for the Bayesian networks and enables a sensor node to predict the probability of discovering areas within the region of interest that require coverage. The CA algorithm for a deployment of 10 and 15 nodes approximates the actual coverage to within ± 10% in both cases. However, and because of scalability issues the maximum errors associated with the coverage approximation are 20% for 10 nodes and 40% for the 15 nodes case. This is due to the approximated coverage being incorrectly positioned within the region of interest. Analysis of the CA algorithm’s dependency on network size revealed that the maximum error associated with the coverage approximation increases with the size of the network.

A novel decentralised Bayesian network based self-healing algorithm (BayesMob) has been presented which coordinates the physical relocation of sensor nodes to counteract timely and abrupt node failures. The Bayesian network enables each sensor node to predict the probability of coverage increasing or remaining constant, given the sensor node motion direction. The evidence for these predictions is derived from local neighbouring node information which the sensor node retains. The performance of BayesMob is compared to CoFi and a static WSN. BayesMob maintains the coverage for a greater percentage of failed nodes which is shown via coverage loss plots. For example, a deployment of 100 nodes and a coverage loss criterion of 20%, a WSN running BayesMob will tolerate 72% of the nodes failing in comparison to 58% from a network running CoFi.

BayesMob can also tolerate abrupt node failures, such as malicious destruction or malfunction by opposition to its counterpart (CoFi) which is limited to node failures due to energy exhaustion. Under CoFi the failing sensor nodes coordinate the physical relocation of neighbouring sensor nodes. Therefore, an abrupt failure may result in the sensor node dying before it has sufficient time and resources to coordinate the self-healing. The reduced coverage loss associated with BayesMob comes at the cost of slightly larger motion energy overheads. Given the assumed hardware configuration, for 50% of the network failing, BayesMob consumes 0.2% of the total energy in comparison to 0.025% from CoFi. Whilst the latter has exceeded the coverage criterion of 20%, BayesMob has only lost 11% coverage. BayesMob can tolerate up to 60% of nodes failing before the coverage criterion is exceeded.
Finally, the work has resulted in a WSN simulation environment developed in Matlab. The simulator provides a tool for the evaluation of the performance of mobile WSN under various scenarios thereby enabling researchers and WSN application designers to test new mobility algorithms prior to deployment.

In summary, this thesis has presented the following contributions to the research on mobile WSNs:

- The creation of a new mobile WSN evaluation tool (WSN simulator).
- The WSN Coverage Against Time performance measure.
- The grazing mobility strategy which provides a new approach to achieving total coverage when the size of the mobile WSN is sufficient to completely cover the region of interest.
- The centralised Grazing Reference Point Group Mobility (GRPGM) algorithm.
- Incorporation of Bayesian networks into WSNs to predict coverage and connectivity changes with relation to motion characteristics.
- The Bayesian network based grazing algorithm (BNGRAZ).
- The Coverage Approximation algorithm (CA algorithm).
- The Bayesian network self-healing mobility algorithm (BayesMob).

The work underlying this thesis has resulted in the creation of a number of novel concepts for the coordination and management of motion in mobile WSNs. Firstly, a biologically inspired grazing strategy has been investigated and has led to the implementation and simulation of two algorithms. Secondly, the need to evaluate the effectiveness of this strategy led to the creation of a coverage measurement tool, based around the idea that coverage can only be achieved over time when considering a limited deployment of sensor nodes. Finally, the need to address the uncertainty associated with the use of limited information at the node level has led to the use of a technique to manage the decreasing accuracy (over time) and lack of information associated with the network as a whole, from the perspective of each and every node that forms the network. The author feels that although these concepts have been applied to mobile WSNs they have the potential to provide solutions to a class of problems which are constrained from limited information.
7.2 Future Work

The investigation of the CA algorithm scalability showed that the error associated with the coverage approximation increases as the size of the network increases. This is due to the limited topology information available to each sensor node. It is suggested that an investigation be undertaken to determine the extent of additional information required to improve the scalability, whilst considering the associated overheads. In addition, the investigation of alternative techniques to approximate the WSN coverage, such as implementing a Bayesian network, which would effectively make coverage predictions.

The Bayesian networks based mobility algorithms currently only consider the coverage and connectivity associated with mobility. The creation of Bayesian networks which also predict the network and sensor nodes’ lifetime based on the sensor nodes’ motion characteristics would be of value. Using these predictions a sensor node would determine the value of its existence and therefore evaluate whether it should exhibit altruistic or selfish behaviour.

The BNGRAZ and BayesMob assume that the dependence between a sensor node’s motion and the position of its neighbours can be represented by a Gaussian distribution. It is proposed that alternative functions could be investigated to represent the dependence between mobile sensor nodes.

Although the work presented here has not formally considered issues of stability and convergence of BNGRAZ and BayesMob, other than investigate the limitations of BNGRAZ in response to increasing configuration duty time and MDT, it is recommended that these issues are explored as part of any future work.

Ultimately the mobility strategies and algorithms could be implemented onto a physical mobile WSN platform to demonstrate the actual behaviour of the mobile WSN whilst adopting these algorithms.
References


[3] Berkeley Smart Dust project. [Internet]. [cited 2008 Mar 05]; Available from: http://robotics.eecs.berkeley.edu/-nister/SmartDust/.


[27] JavaSim, Department of Computing Science, University of Newcastle upon Tyne, [Internet]. [cited 2007 Nov 20]; Available from: http://javasim.ncl.ac.uk/.


Appendix A

PUBLICATIONS


A Bayesian network approach to a biologically inspired motion strategy for mobile wireless sensor networks

Matthew D. Coles*, Djamel Azzi, Barry P. Haynes, Alan Hewitt

Electronic and Computer Engineering Department, University of Portsmouth, Anglesea Building, Anglesea Road, Portsmouth, Hampshire, PO1 3DJ, United Kingdom

ARTICLE INFO

Article history:
Received 10 December 2007
Received in revised form 30 October 2008
Accepted 19 November 2008
Available online xxxx

Index terms:
Bayesian networks
Connectivity
Coverage
Lifetime
Mobility
Wireless sensor networks
Fault tolerant networks

ABSTRACT

Mobility strategies for wireless sensor networks (WSNs) are presented. We introduce a grazing mobility strategy for mobile WSNs, inspired by the foraging behaviour of herbivores grazing pastures. We present Bayesian network GRAZing (BNGRAZ) that implements the proposed WSN grazing strategy. BNGRAZ uses local neighbourhood information to predict coverage and connectivity performance changes related to sensor node motion characteristics. This enables a sensor node to predict the performance implications related to its direction of movement. We implement the BNGRAZ approach to grazing in a custom built mobile WSN simulator. The WSN performance criteria considered during the validation process include coverage, redundancy, connectivity, and network lifetime.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

Recent advances in sensor technology (in terms of size, power consumption, wireless communication and manufacturing costs) have enabled the prospect of deploying large quantities of sensor nodes to form wireless sensor networks (WSNs). These networks are created by distributing large quantities of usually small, inexpensive sensor nodes over a geographical region of interest in order to collect data relating to one or more variables. These nodes are primarily equipped with the means to sense, process and communicate data to other nodes and ultimately to a remote user(s). WSN nodes can also have mobility capabilities which enable them to roam the region of interest to harvest information. Sensor nodes may cooperate with their neighbours (within communication range) to form an ad-hoc network. WSN topologies are generally dynamic and decentralized. WSNs have a wide range of applications including military, environmental monitoring, health, home, space exploration, chemical processing, and disaster relief.

The majority of WSN research has assumed that the nodes are static and that once deployed they are unable to relocate. This limits the ability of WSNs to adapt to changing operating environments. A large number of applications involve a dynamic environment and/or do not necessarily need the deployment of large quantities of static nodes.

A mobile WSN varies from traditional static WSNs in the obvious sense. A fraction of the sensor nodes can have motion capabilities which enable the WSN to change position over time according to some strategy. This motion may be achieved by including motors and servos onto the node platform. Mobility capabilities may also be possible by attaching the nodes to other mobile entities. This gives the nodes the ability to physically change position in relation to neighbouring nodes and also the environment in which the nodes are situated. The nodes may move

Please cite this article in press as: M.D. Coles et al., A Bayesian network approach to a biologically inspired ..., Ad Hoc Netw. (2008), doi:10.1016/j.adhoc.2008.11.002
individually to optimize the performance of the network. Mobility strategies aimed at this have been presented by Kansal et al. [5–7], and Rao and Biswas [9]. Nodes may also move cooperatively around the geographical region of interest to adapt to the environment and/or application criteria.

When deploying a WSN in a region of interest a number of alternative approaches can be adopted. The region could be covered with a large quantity of static nodes, which achieve the desired coverage criteria at initial deployment. The network would generally incorporate a high level of redundancy in order to extend its lifetime, with redundant nodes providing a level of fault tolerance. The drawback of this approach is that optimum deployment is required to achieve the desired level of performance. When considering deployment to unmanned remote regions, optimum deployment becomes difficult to achieve.

An alternative approach is to deploy a smaller quantity of mobile nodes. These nodes would not achieve the desired level of coverage at any instantaneous point in time; however, over a finite period of time total coverage (specified coverage criterion) could be achieved. In order to achieve this nodes migrate around the geographical region collecting data and thus providing total coverage. This concept will be referred to as a grazing strategy, by analogy to a herd of herbivores feeding off pastures.

This paper presents a mobility scheme based on a decentralized Bayesian network GRAZing (BNGRAZ) algorithm that adopts the proposed WSN grazing strategy. BNGRAZ uses discrete Bayesian networks to predict the likelihood of deterioration in performance given that the sensor node moves in a particular direction. The evidence used for prediction is obtained from local neighbourhood information, which minimizes the communication overheads and provides scalability. The performance criteria considered under the BNGRAZ algorithm include connectivity and coverage. The paper also presents a distributed coverage approximation (CA) algorithm. This algorithm enables the sensor nodes to approximate the collective coverage of the WSN using only local knowledge of neighbouring node configuration. The CA algorithm is required for the successful operation of the BNGRAZ algorithm.

The simulation results have been obtained by implementing the proposed BNGRAZ algorithm using a custom built simulator which allows the evaluation of the performance of mobile WSNs.

Section 2 of this paper outlines related work on mobile WSNs, and their inherent performance implications and benefits. Section 3 discusses the grazing strategy in detail outlining the performance implication and benefits. Section 4 discusses the proposed decentralized BNGRAZ algorithm that aims to achieve the grazing motion behaviour in WSNs. In Section 5 we present the simulation results and compare the results to a generic fixed path approach. Finally, 6 concludes the paper and discusses future work.

2. Related work

WSNs are currently a widely researched topic. For a detailed review of architecture, management, communication protocols, and their current and potential applications, see [2].

Mobility as a control primitive for self-deployment of WSNs has been investigated. For example [18] proposes a distributed self-deployment protocol which uses Voronoi diagrams to discover coverage holes caused by non uniform deployment. The paper proposes three movement-assisted sensor deployment protocols which essentially relocate sensor nodes from densely deployed regions to areas with sparse coverage. In [14] a self-deployment protocol for heterogeneous WSNs is proposed.

Mobility as a control primitive for improving network performance has also been investigated. For example [6] proposes a distributed coverage fidelity (Co-Fi) algorithm that controls the relocation of a sensor node in order to repair coverage holes which are assumed to be a consequence of node failure. In [13] a dynamic coverage maintenance (DCM) scheme is proposed that also exploits the limited mobility of sensor nodes for active fault repair of the WSN. Four distributed rule based DCM algorithms are proposed which rely on local neighbourhood topology information for coordinating the sensor relocation.

In [4] an event based mobility scheme is proposed that coordinates the relocation sensor nodes to areas that require a higher sensing resolution. Using single dimensional mobility for improving sensing resolution and overcoming unpredictable environmental influences has also been investigated. In [5] a low complexity single dimension mobility strategy is proposed which has low energy actuation primitives. The nodes move along a single dimension to counteract a loss in coverage caused by environmental influences such as the presence of obstacles.

Low complexity mobility was also investigated in [10] and [11] through the development of the Network Information Management System (NIMS). NIMS's integrate distributed, embedded sensing and computing systems with infrastructure - supported mobility.

Using mobility as a control primitive to extend the network lifetime by balancing the energy discharge rate (EDR) between all sensor nodes has also been investigated. For example [9] proposes a biologically inspired mobility model for balancing the energy overhead related to communication.

The group migration of a WSN has also been investigated. For example [19] presents a mobility management protocol for group migration of sensor nodes. The paper suggests that in order for a WSN to provide the specified coverage or track moving targets, it should incorporate some level of mobility. In [15] a centralized group migration strategy is proposed. The commander (user) controls the motion of a cluster of mobile sensor nodes. The nodes are deployed to monitor a square area at a certain distance ahead of the commander's motion direction. As the speed and direction of the commander changes, the new positions of the sensor nodes are calculated by the proposed movement control algorithm. Other work assumes alternative infrastructures with nodes acting as commanders (referred to as actors) Vassilis et al [25]. In this work, we assume that the WSN has a single commander node which is essentially a sensor node capable of aggregating and forwarding all sensing data to an eventual recipient via a network gateway.
In [3] and [12] an alternative approach to mobility is proposed. The papers propose that benefits can be drawn by adopting a mobile base station or gateway. The sensor nodes would then remain static. The base station would reposition to regions with high throughput to reduce delay and communication overheads.

3. Grazing strategy

3.1. Herbivore grazing behaviour

Grazing can be classified as a type of foraging. The technical meaning of foraging within the science of behavioural ecology refers to predator–prey interactions, see [1]. Grazing differs from predation–prey foraging in that the organism being eaten is not killed, see [16] and [17]. The most commonly understood example of grazing is mammals feeding on grasslands although it can also refer to any plants, algae or plankton that are predated by reptiles, insects, birds etc.

3.2. WSN grazing strategy

We present here a mobility strategy for mobile WSNs that emulate the behaviour of herbivores grazing pastures. The geographical regions where we wish to monitor a phenomenon of interest are analogous to the pasture, and the mobile sensor nodes are analogous to the herbivores grazing.

Instead of deploying a large quantity of static sensor nodes to a geographical region of interest, which is generally the consensus for WSNs, we propose to deploy a smaller number of mobile sensor nodes that cooperatively migrate around the region of interest collecting the relevant data. At time $t$ the desired coverage level (total coverage) would not be achieved. However, at time $t + WSN$ migration time (time taken by the WSN to completely cover the region of interest) total coverage would have been achieved. This coverage measure is referred to as coverage against time ([20] gives a detailed description of this measure).

As the mobile sensor nodes migrate around the region of interest, connectivity must be maintained. The WSN should also seek to maintain a high level of instantaneous coverage. Incorporating node mobility allows the WSN to adapt to application and environment changes. The network also becomes fault tolerant. The grazing strategy is aimed towards applications where constant information about the region is not required, and thus lengthy periods between phenomena changes are common. The Grazing WSN would be capable of providing data at regular intervals. This interval would be specified by the user and is referred to as the coverage period. Using the analogy to grazing Fig. 1 depicts the correlation between the pasture height and the coverage weight. Coverage weight ranges from zero to one. A value of one indicates that data has recently been collected from that area or the animals have completely exhausted the grass in that area of the pasture. A value of zero indicates that data has not been collected from that area for a time period greater than or equal to the coverage period, or the grass is long and has not been grazed. All values between the limits would represent the coverage status or the grass length.

The sensor nodes would be continuously searching for fresh information that the WSN has not collected. This is similar to the herbivores searching for the highest nutritional food. Possible applications include mapping remote regions, pollution monitoring, landmine detection, and habitat monitoring.

Attaching motors and servos to the sensor node platform could also increase its size and weight. However, motion could alternatively be harnessed from external entities. For example, if we consider a pollution monitoring application within a city or town. The WSN could harness motion via attaching the sensor nodes to vehicles or humans carrying them (possibly in the form of an additional mobile phone feature). Localization and navigation are other factors that need to be addressed when considering mobile WSNs. Nodes should be location aware so that they can navigate successfully around the area of interest.

Kansal et al. [8] discusses these additional considerations when reviewing controlled mobility within WSNs.

The following section outlines the BNGRAZ algorithm that implements the grazing strategy.

4. BNGRAZ

The Bayesian network GRAZing (BNGRAZ) algorithm adopts a distributed approach to motion control. Each sensor node carries a set of Bayesian networks that use probabilistic reasoning to determine the optimum node motion direction. The node effectively predicts the performance benefits and implications of moving in a given direction. A Bayesian network is a graphical structure that describes how each sensor node evaluates the uncertainty within the WSN (see, e.g. [22]).

The BNGRAZ incorporates three discrete Bayesian networks that predict the probability of the WSN performance increasing or remaining constant/unchanged given the WSN topology and sensor node’s motion direction. The variables used in BNGRAZ are outlined in Table 1. Bayesian network 1 (Bn1) determines the probability
Table 1

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>True</td>
</tr>
<tr>
<td>F</td>
<td>False</td>
</tr>
<tr>
<td>N</td>
<td>Cardinal direction north</td>
</tr>
<tr>
<td>S</td>
<td>Cardinal direction south</td>
</tr>
<tr>
<td>E</td>
<td>Cardinal direction east</td>
</tr>
<tr>
<td>W</td>
<td>Cardinal direction west</td>
</tr>
<tr>
<td>i</td>
<td>Cardinal direction indices (N, S, E, W)</td>
</tr>
<tr>
<td>Ci</td>
<td>Sensor node communication range</td>
</tr>
<tr>
<td>Ci</td>
<td>Connectivity decrease in cardinal direction i ∈ {true, false}</td>
</tr>
<tr>
<td>Di</td>
<td>Discover un-serviced (area that has not been covered by the WSN) in cardinal direction i ∈ {true, false}</td>
</tr>
<tr>
<td>N</td>
<td>Need to move in cardinal direction i ∈ {true, false}</td>
</tr>
<tr>
<td>Oi</td>
<td>Optimum direction i ∈ {N, S, E, W}</td>
</tr>
<tr>
<td>Ai</td>
<td>Neighbour in cardinal direction i ∈ {true, false}</td>
</tr>
<tr>
<td>di</td>
<td>Neighbour distance &lt; Ci in cardinal direction i ∈ {true, false}</td>
</tr>
<tr>
<td>Ui</td>
<td>Un-serviced area (area that has not been covered by the WSN) in cardinal direction i ∈ {true, false}</td>
</tr>
<tr>
<td>0</td>
<td>Motion angle of sensor node</td>
</tr>
</tbody>
</table>

that connectivity will increase if the sensor node moves in one of the cardinal directions (Ci = F). Bayesian network 2 (Bn2) determines the probability that an un-serviced area will be discovered if the sensor node moves in one of the cardinal directions (Di = T). Bayesian network 3 (Bn3) aggregates the Ci and Di to determine the probability of the optimum direction Oi which is based on maintaining or increasing the WSN performance from connectivity and coverage perspective (discovering un-serviced areas). The structure for BNGRAZ is shown in Fig. 2.

Netica application software from Norsys [21] was used to create and validate the Bayesian networks. The Bayesian networks are used for predictive reasoning. A node is a parent of a child if there is an arc from the former to the latter. The BNGRAZ only considers discrete variables so the relationships between connected nodes are represented by a conditional probability table (CPT) [24]. Therefore, for each distinct instantiation of the parent node the value that the child node will take is specified via the CPT. The CPT values are generally specified using statistical data from the system, and or Bayes' theorem. The values within BNGRAZ have been specified with a view to achieve the desired response from the Bayesian networks.

The evidence that is fed into the Bayesian networks is acquired from local neighbourhood information. The predictions are based on a possible move in one of the cardinal directions north, south, east, or west. The joint probability function Bn3 is given in Eq. (1). The probability of Oi = i is calculated by marginalizing Eq. (1), which is shown in Eq. (2).

\[ P(Oi, Ni, Ns, Ne, Nw) = \frac{P(Ni)P(Ns)P(Ne)P(Nw)}{P(Oi, Ni, Ns, Ne, Nw)} \]

\[ P(Oi = i) = \sum_{Ni, Ns, Ne, Nw \in \{TF\}} P(Ni)P(Ns)P(Ne)P(Nw) \times P(Oi = i | Ni, Ns, Ne, Nw), \]

where P(Ni) to P(Nw) are calculated using Eqs. (3) and (4):

\[ P(Ni, Ci, Di) = P(Ci)P(Di)P(Ni | Ci, Di). \]

\[ P(Ni = T) = \sum_{Ci, Di} P(Ci)P(Di)P(Ni = T | Ci, Di). \]

4.1. Bayesian network 1 (Bn1): probability of a decrease in connectivity

Bn1 calculates the conditional probability of a decrease in connectivity if the sensor node moves in one of the cardinal directions given evidence from all neighbours positions such as the probability of a neighbour lying in each of the cardinal directions and probability that the closest neighbour's distance in each direction is less than the communication range (Ci). Sensor nodes transmit all sensor data to the commander node. Hence, connectivity is defined as the ability of a sensor node to communicate effectively with the commander node.

The joint probability function for Bn1 is given in Eq. (5). P(Ci = F | 0 = i) is calculated by using joint probability theory and marginalization, which is given in Eq. (6).

\[ P(Ci = F | 0 = i) = \frac{P(Ci = F, 0 = i)}{P(0 = i)} \]

\[ = \sum_{Ni, Ns, Ne, Nw \in \{TF\}} P(Ni)P(Ns)P(Ne)P(Nw) \times P(0 = i | Ci = F, 0 = i) \times P(Ci = F | 0 = i, Ni, Ns, Ne, Nw), \]

where P(Ni) to P(Nw) are calculated using Eq. (7):

\[ P(Ni = T) = \sum_{A, d \in \{TF\}} P(A)P(d)P(Ni = T | A, d). \]

The belief in the evidence A, and d, are calculated from a contacts information table (CIT) local to each sensor node. The CIT stores all known information about neighboring nodes and also sensor nodes that route data packets through the node destined for the commander node.
sensor node represents each of its contacts positions (motion) with two Gaussian distributions. Therefore, the CIT stores a mean ($\mu$) and a standard deviation ($\sigma$) for each of its contacts $x$ and $y$ coordinates. The CIT also contains additional information such as a timestamp associated with last communication and a relay count associated with the number of communication hops to the contact. The motion characteristics such as motion direction ($\theta$) and speed are also stored.

A sensor node must reconfigure at regular intervals to maintain a valid contacts table. The reconfiguration allows sensor nodes to exchange information with neighbouring nodes for WSN maintenance purposes. This carries a communication overhead resulting in the need to extend the time between reconfigurations; this in turn, leads to increased uncertainty. Sensor nodes handle this uncertainty by representing contacts locations as Gaussian distributions which vary in time. This expresses the fact that information becomes less accurate the longer a node goes without being updated with contact information. In addition, the assumed Gaussian distributions provide a mechanism for quantifying the interdependencies between sensor nodes. When the contacts table is updated during reconfiguration, the CIT stores the contacts current $x$ and $y$ coordinates, which represent its contacts mean position. Also the standard deviations associated with $x$ and $y$ are $\sigma_x = \sigma_y = 1$ m (arbitrarily initial value) which allows for localization errors.

During the period between reconfigurations and assuming the sensor node has not been in contact with its contacts, it periodically increases the $\sigma_x, \sigma_y$ associated with each contact entry thus increasing the uncertainty associated with the contacts coordinates. The standard deviation increase is calculated using Eq. (8).

$$\sigma = \sigma + \nu \cdot \text{Update}, \quad \text{(8)}$$

where $\nu$ is the node speed and $t_{\text{update}}$ table update duty cycle time.

4.2. Probability of neighbour distance less than communication range

The Gaussian distributions associated with each of the contacts coordinates are used to calculate the probability a neighbour's distance is less than the communication range ($P(d(j) = T)$) where $j = 1, \ldots, n$, and $n$ equals the total number of neighbours from the contacts table). If $x$ and $y$ are independent Gaussian random variables with nonzero means then the distance $z = \sqrt{x^2 + y^2}$ has a $f_z(z)$ can be represented by a Rician distribution, which is given in Eq. (9).

$$f_z(z) = \frac{z e^{-\frac{(z^2 + \mu^2)}{2\sigma^2}}}{\sigma^2} \cdot \text{I}_0\left(\frac{2z\mu}{\sigma^2}\right), \quad \text{(9)}$$

where

$$\mu = \sqrt{\mu_x^2 + \mu_y^2}, \quad \theta = \text{tan}^{-1}\left(\frac{\mu_y}{\mu_x}\right), \quad \sigma = \sigma_x = \sigma_y,$$

$$\mu_x = \mu \cos \phi, \quad \mu_y = \mu \sin \phi,$$

and

$$\psi \leq \text{mod}(\psi, 2\pi).$$

Eq. (10) is the modified Bessel function of the first kind and zeroth-order.

$$P(d(j) = T) \text{ in the cardinal direction } i \text{ is approximated by numerically integrating the Rician distribution between zero and the } C_i.$$

4.3. Probability of a neighbour lying in each of the cardinal directions

$$P(A(j)); j = 1, \ldots, n \text{ is approximated using Algorithm 1, because the p.d.f } f_\theta(\theta) \text{ of the angle } \theta = \text{tan}^{-1}\left(\frac{\mu_y}{\mu_x}\right) \text{ is intractable when } x \text{ and } y \text{ are independent Gaussian random variables with nonzero means:}$$

All the $P(d(j) = T)$ and $P(A(j) = i)$ are aggregated to approximate the evidence for Bn2. This is shown in Eqs. (11) and (12). $P(d(j) = T)$ is scaled with respect to the belief that the neighbour is in direction $i$.

4.4. Coverage approximation (CA) algorithm for Bn2

When predicting the probability of discovering an unserviced area (Bn2) for the BNGRAZ algorithm, the WSN coverage is required. The communication overheads

$$P(A(j) = i) = 1 - \prod_{j=1}^{n} P(A(j) = i), \quad \text{(11)}$$

$$P(d(j) = T) = 1 - \prod_{j=1}^{n} P(d[j]) \times P(A(j) = i). \quad \text{(12)}$$

Algorithm 1. Approximate $P(A(j) = T)$

$x, y$ = sensor node's coordinates, and $\sigma = \sigma_x = \sigma_y$

$\mu_{\text{distance}} = \sqrt{(x - x_j)^2 + (y - y_j)^2}$ = mean distance from neighbor.

If $\sigma < (4 \times \mu_{\text{distance}})$

Assume the $f_\theta(\theta)$ p.d.f is represented by a Gaussian distribution with

$$\sigma_\theta(\sigma) = 100 - 100e^{-\left(\frac{\mu_\theta}{\sigma_{\text{distance}}}\right)}$$

$$\mu_\theta = \text{tan}^{-1}\left(\frac{\mu_y}{\mu_x}\right).$$

//Probability of direction is approximated by integrating between
45 to 135 and 405 to 495 degrees for north
135 to 225 and -225 to -135 degrees for west
225 to 315 and -135 to -45 degrees for south
315 to 360 and 0 to 45 and -45 to 0 and 360 to 405
degrees for east*/

else

Assume $f_\theta(\theta)$ is represented by a uniform distribution
//assume a uniformed distribution between 0 and 360 degrees/

end

4.4. Coverage approximation (CA) algorithm for Bn2

When predicting the probability of discovering an unserviced area (Bn2) for the BNGRAZ algorithm, the WSN coverage is required. The communication overheads

$$\psi \leq \text{mod}(\psi, 2\pi).$$
restrict this information from being frequently broadcast to all sensor nodes. Also the sensor nodes are unable to accurately calculate the WSN coverage, because they are not aware of the positions of all nodes within the WSN. The CA algorithm produces a decentralized approximation of the WSN coverage at the level of each node. The sensor node uses this approximation to generate and maintain a coverage map, which represents the coverage status of the region of interest. Essentially, which areas have been covered by the WSN?

A sensor node is only aware of its own position and that of its neighbouring nodes and possibly sensor nodes that relay data packets through the node. This information is stored in the CIT discussed above. The CA algorithm uses this local neighbourhood information and the total number of sensor nodes within the WSN to approximate the WSN coverage.

The coverage map is generated by representing the region of interest by a grid. Each intersection of a horizontal and vertical grid line represents a pixel point therefore; the coverage map is represented by a matrix of pixel points. The value associated with each pixel point represents its coverage status at that point and ranges between 0 and 1. The pixel point value 1 indicates the point has been covered by the WSN. If the point has not been covered or the time period between previous coverage is greater than the desired coverage period the value will equal 0. Fig. 3 shows a graphical representation of generating the coverage map using pixel points. The coverage resolution is related to the distance between the horizontal and vertical line and thus the quantity of pixel points.

This method of using grid line intersections has been implemented in order to minimize the computational overhead related to calculating the precise area covered by the sensor nodes. This becomes computationally heavy when dealing with multiple sensor node coverage regions.

The pixel point matrix size is calculated using Eq. (13).

\[
Md_x = 1 + \frac{S_x}{R} \quad Md_y = 1 + \frac{S_y}{R}
\]

(13)

where \(Md_x\) is the number of columns and \(Md_y\) is the number of rows. Therefore, the total number of pixel points that define the coverage map equals \(Md_x \times Md_y\). \(S_x\) is the length of the region along the \(x\) and \(S_y\) is the length of the region along the \(y\). \(R\) is the desired coverage resolution and specifies the distance between the pixel points. This has been set to 10 m for the proposed CA algorithm.

The CA algorithm is executed by the sensor node during the reconfiguration period, when neighbouring node positions are updated. The CA algorithm is outlined in Algorithm 2.

CA first approximates the total coverage area of the WSN by calculating the mean distance from neighbouring nodes and also the total number of sensor nodes. This is generated in the form of a circle. CA determines the centre point of this circle by evaluating the direction of neighbouring nodes and also other contacts.

The approximated coverage area is then transposed onto the coverage map. A pixel point is assigned a value of 1 if it is within the approximated WSN coverage area. All other pixel point values are then decremented linearly with time at a rate defined by the coverage period. Therefore, if we assume a pixel point is covered at time \(t\), its corresponding value equals one. Then at \(t + \text{coverage period}\) assuming the point has not been recovered the pixel point will equal zero. Validation results for the CA algorithm are presented in Section 5.

**Algorithm 2. WSN CA (coverage approximation)**

*/approximate the coverage area of the WSN/*

1. Calculate the mean distance from neighbouring nodes.
2. Approximate an assumed circular coverage area of the WSN using the mean distance and quantity of sensor nodes.
3. Determine the radius of the approximated coverage area.
4. Rules to determine the centre position of the approximated circular WSN coverage area/*
5. Calculate the direction of all neighbouring nodes.
6. Determine the cardinal direction (north, south, east, and west) of neighbouring nodes based on their direction.
7. If neighbouring nodes lie in all cardinal directions and no other routing information is available.
   - Position approximated WSN coverage centre at nodes coordinates and add to coverage map.

else

if only neighbouring node information is available and in one or more of the cardinal directions there are no neighbours.
   - Shift approximated WSN coverage centre by WSN coverage radius minus sensing range. To the centre of all other cardinal directions that contain neighbours.

else if other routing information is available from contact nodes that rely on the node forwarding data packets to the commander node.
   - Shift equals distance from contact plus sensing range minus WSN approximated coverage radius. Shifted in the direction of contact node.

end

4.5. Bayesian network 2 (Bn2): probability of discovering un-served area (\(P(D_i = T | \theta = 1)\))

Bn2 calculates the probability of the sensor node discovering some area within the geographical region of interest that requires coverage by the WSN given that the sensor node moves in one of the cardinal directions and given an approximated coverage picture of the region of interest. This is referred to as discovering un-served area. The joint probability function for Bn2 is given in Eq. (14).

The evidence for Bn2 \((U_n, U_s, U_l, U_w)\) represent the fraction of un-served area in each of the cardinal directions. The un-served area values are calculated by evaluating a local coverage map generated and maintained by the CA algorithm as discussed above.

Please cite this article in press as: M.D. Coles et al., A Bayesian network approach to a biologically inspired ..., Ad Hoc Netw. (2008), doi:10.1016/j.adhoc.2008.11.002
The values assigned to $U_N$, $U_S$, $U_E$, $U_W$ are calculated by referencing the sensor node's position with respect to the coverage map. The fraction of uncovered pixel points against the total number of points, north of the sensor node yields $U_N$, south of = $U_S$, east of = $U_E$, and west of = $U_W$.

The probability of discovering un-serviced area given that the node move in direction $i$ ($P(D_i = T|\theta = i)$) is calculated by using joint probability theory and marginalization, which is given in Eq. (15).

$$P(D_i, \theta, U_N, U_S, U_E, U_W) = P(\theta)P(U_N)P(U_S)P(U_E)P(U_W)$$

$$P(D_i|\theta, U_N, U_S, U_E, U_W)$$

$$P(D_i = T|\theta = i) = \frac{P(D_i = T, \theta = i)}{P(\theta = i)}$$

$$= \sum_{U_N, U_S, U_E, U_W} P(U_N)P(U_S)P(U_E)P(U_W)$$

$$\times P(\theta = i) \times P(D_i = T|\theta = i, U_N, U_S, U_E, U_W).$$

$$\text{Eq. (15)}$$

4.6. Selecting the motion status and direction

The sensor node finally selects the motion direction by evaluating probability of the optimum direction $P(O_a)$ and determining which direction yields the maximum probability of performance being maintained or increased. These are taken from the output of BN3. The sensor node will then move in this direction if the maximum likelihood of performance increase or maintenance is greater than 0.26 (26%), else the node will become static until the next BNGRAZ execution.

5. Results and discussion

5.1. Simulation set up

All simulations have been generated using a custom built WSN simulator. The geographical region of interest was set to a $100 \times 100$ m area. Every sensor node is equipped with the capability of movement. Table 2 details the settings of the simulation parameters. The sensor nodes were configured to generate and transmit data packets destined for the commander node in a duty period of 2 min. We assume a perfect medium access control (MAC) protocol.

The sensor nodes were uniformly randomly deployed to a $20 \times 20$ m area at the centre of the region of interest, which ensures initial connectivity. The simulations were run until the commander node becomes energy exhausted, which we define as the network lifetime.

The simulation results evaluate the performance of the BNGRAZ algorithm with respect to the coverage period, reconfiguration duty time, quantity of sensor, and the nodes' speed. The user defined coverage period represents the desired total coverage frequency which determines the decrement rate of the pixel points' values within the coverage map. The reconfiguration duty time defines the period between neighbor communications used to update localization information within the CIT.

In Section 5.2 we present the validation results for the CA algorithm. Section 5.3 evaluates the connectivity when increasing the reconfiguration duty time and sensor node motion speed. Section 5.4 evaluates the instantaneous coverage when increasing the reconfiguration duty time and the number of sensor nodes. Finally, the coverage against time with respect to the number of sensor nodes, sensor node speed, reconfiguration duty time, and coverage period is evaluated in Section 5.5.

Table 2: Simulation parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Reasoning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication range</td>
<td>20 m</td>
<td>Typical ranges observed in external environments with Berkeley motes</td>
</tr>
<tr>
<td>Sensing range</td>
<td>10 m</td>
<td>Sensing range is half the communication range</td>
</tr>
<tr>
<td>Mobility cost</td>
<td>17.8758 J/m</td>
<td>XE robot platforms [23]</td>
</tr>
<tr>
<td>Total initial energy</td>
<td>87,480 J</td>
<td>Capacity of 6 V alkaline battery</td>
</tr>
</tbody>
</table>

Please cite this article in press as: M.D. Coles et al., A Bayesian network approach to a biologically inspired ..., Ad Hoc Netw. (2008), doi:10.1016/j.adhoc.2008.11.002
5.2. CA algorithm validation

When presenting the BNGRAZ algorithm we proposed the CA algorithm. This algorithm allows each sensor node to approximate the WSN coverage, based on local neighbourhood information. We adopt two approaches to validate the CA algorithm. First, we evaluate the size of the approximated coverage area against the actual coverage. Secondly, we evaluate the position of the approximated coverage with respect to the sensor node, against the actual coverage. Ten nodes were randomly deployed to a 20 x 20 m area with the nodes adopting the BNGRAZ algorithm. Under the BNGRAZ the initial coverage is low due to random deployment. The sensor nodes then physically relocate to maximize coverage (instantaneous coverage and coverage against time).

Fig. 4 show the average size of the sensor nodes approximated coverage area and the maximum deviations. The actual size of the coverage area is also presented for comparison. The results show that this approximation is accurate to ±10%.

The position of the approximated coverage with respect to region of interest is evaluated with the coverage error plot shown in Fig. 5. The latter shows the average error related to the sensor nodes approximated coverage and the maximum deviations around the average. The coverage error plot is generated by evaluating each sensor nodes coverage map pixel point values against the actual coverage. If a pixel point within the nodes’ coverage map is incorrectly plotted as covered or uncovered it results in a coverage error. The coverage error is represented as a percentage of pixel points incorrectly plotted against the total number of pixel points. The coverage error plot shows that maximum coverage error does not exceed approximately 31%. The BNGRAZ algorithm can handle this error due to the probabilistic nature of the algorithm.

5.3. Connectivity

Maintaining a degree of connectivity within the WSN is vital to performance. Connectivity is defined as the ability of a sensor node to communicate with the commander node through single hop or multi hop routing. Connectivity ensures that sensor data can be successfully transmitted to the user or application system. Therefore, we evaluate the connectivity of the WSN while adopting BNGRAZ.

The reconfiguration duty time was increased from 0 (i.e. all nodes are aware of neighbours position all the time, ideal but not feasible) to 120 s. The connectivity was evaluated over twenty trails and the average was record for each reconfiguration duty time. The WSN consisted of fifteen nodes and the connectivity was evaluated for a sensor node speed of 0.1 and 0.05 m/s. Fig. 6 shows the resultant connectivity when considering these parameters.

The results show that an increase in the reconfiguration duty time will lead to errors which will have an adverse effect on connectivity. This is due to sensor nodes being left to guess their neighbour positions in the absence of accurate information.

Fig. 6 also shows that the node speed influences the connectivity as it affects the likelihood that a neighbour will move out of communication range between the reconfiguration periods. As the sensor node speed is reduced the reconfiguration duty time can be extended. The sensor node speed and configuration duty time must therefore, be selected to ensure that connectivity is maintained.

5.4. Instantaneous coverage

Traditionally, coverage within a WSN is defined as the fraction of the region of interest collectively covered against the total size of the region of interest. When con-
considering the grazing strategy, coverage defined in this way, only points to the fraction of the region of interest covered by the mobile sensors. Coverage becomes a time dependent variable also expressing the fact that the total coverage cannot be achieved at any instant in time. Total coverage is achieved in time by adopting the grazing strategy.

The instantaneous coverage will effectively depend on the number of sensor nodes deployed, sensing range, and also the reconfiguration duty time. The reconfiguration duty time was increased from 0 to 120 s. We considered a deployment of 9 and 15 nodes. The instantaneous coverage was evaluated over twenty trails and the average was recorded for each reconfiguration duty time. Fig. 7 shows the resultant coverage when considering these parameters.

The results show that an increase in the reconfiguration duty time will lead to a reduction in the instantaneous coverage. This is due to errors caused by the sensor node being left to guess its neighbours’ positions in the absence of complete information about all the nodes in the network.

5.5. Coverage against time

When considering the grazing strategy the instantaneous coverage does not give a true representation of coverage. This is due to the WSN migrating around the region and coverage is achieved over time. Therefore, we need to evaluate the time taken for the WSN to converge to the desired level of coverage, referred to as total coverage. This is defined as the actual coverage period and is effectively the time period between the initial deployment and total coverage. Or the time period between two consecutive total coverage periods.

This paper adopts the service interval (for each pixel point) as a performance criterion for the proposed grazing algorithm. The service interval is the time period between a point in the region of interest being covered in one coverage period and recovered in the next coverage period.

The region of interest was represented by 441 pixel points. We evaluate the pixel point service interval time with respect to the sensor nodes’ speed, the desired coverage period and the quantity of sensor nodes deployed. The results of the service interval times are presented in pixel point service interval distributions. The service interval distributions were generated from 20 simulation trails.

Fig. 8 compares pixel point service interval distributions for a deployment of 9 and 15 sensor nodes. The node speed was set to 0.1 m/s and a reconfiguration duty time of 30 s. The desired coverage period was set to 120 min. All the service interval distributions have a peak around the origin (0 min) which is due to the area covered at initial deployment and the beginning of a new coverage period. Fig. 8
shows that for a deployment of 9 nodes approximately 25% of the pixel point service interval times are within 10 min. For a deployment of 15 nodes this increases to approximately 38%. This is due to the increased instantaneous coverage. The increased instantaneous coverage also shifts the distribution to the left implying an increased likelihood of achieving the desired coverage period. The deployment of 9 nodes flattens after 0–10 min and there is higher likelihood of the service interval time lying between 60 and 100 min.

The reconfiguration duty time was also varied to evaluate the BNGRAZ service interval distribution. Fig. 9 shows the pixel point service interval distributions for a reconfiguration time of 30 and 120 s. Again the results were taken from twenty simulation trails. The node speed was set to 0.05 m/s, 9 nodes were deployed and the desired coverage period was 120 min. The increased reconfiguration duty time has an adverse affect on the pixel point service interval time, due to the reduced connectivity and instantaneous coverage. This can be observed by the reduced peak around the origin. Also the distribution shifts to the right implying a reduced likelihood of the distribution achieving the desired coverage period. The majority of the pixel point service interval times for a reconfiguration duty time of 120 s the times range from 0 to 160 min.

Fig. 10 shows the pixel point service interval distributions for a sensor node speed of 0.1 and 0.05 m/s. The reconfiguration duty time was set to 60 s to ensure connectivity remains above 90%. We deployed a WSN with 15 nodes with a desired coverage period of 120 min. For a node speed of 0.1 m/s 38% of the pixel point service interval times are within 10 min, whereas a node speed of 0.05 m/s yields 32%. Also the bulk of the pixel point service interval times for a speed of 0.1 m/s range from 0 to 120, whereas for a speed of 0.05 m/s the interval times range from 0 to 150 min. As anticipated the node speed directly affect the migration time of the WSN and thus the actual coverage period. However, the faster a sensor node moves the greater its EDR which will result in a reduced node lifetime.

The desired coverage period (set by the user) defines the decrement rate of the coverage maps, which is outlined in Section 4. The desired coverage period was varied in order to evaluate its effect on the pixel point service interval distribution. The region of interest was deployed with 15 nodes each moving at a speed of 0.05 m/s. The configuration duty time was set to 60 s. The distribution is shown in Fig. 11.

As the desired coverage period is extended the resultant distributions flatten and shift to the right. Observations suggest that this is due to the decrement period of the coverage map being extended. Therefore, when each node approximates which area within the region requires coverage this will affect the probability of discovering an un-serviced area. The net result will be that the WSN takes longer to migrate around the region of interest.

When defining the desired coverage period the application designer should consider the node speed. Alternatively the node speed will define the desired coverage period. Also the size of the region of interest will play a role.
in the selection of the above. If the desired coverage period is too small in comparison to the node speed and region size the WSN will not migrate successfully around the region of interest. This will be due to the map decrementing faster than the WSN can migrate. The net result will be that the WSN oscillates around its current position.

6. Conclusion

This paper has presented a novel biologically inspired mobility strategy for WSNs. We argue that instead of deploying a large number of static sensor nodes which provide total coverage at any instant in time, a relatively smaller number of mobile nodes can be deployed within the region of interest. These nodes adopt a grazing strategy to achieve the desired coverage over time. We proposed a decentralized discrete Bayesian network to implement the grazing strategy (BNGRAZ). The algorithm predicts the WSN's performance with respect to a sensor node's mobility. This is to enable the latter (sensor node) to derive the optimum direction of motion. Performance predictions are derived at the level of the sensor node based on neighbour positions and a measure of the overall WSN coverage. A sensor node's position is assumed to be Gaussian in nature with a variable spread; this approach takes into account the fact that the knowledge about a particular node's position will become less accurate the longer one goes without communicating with it.

Simulation of the impact of varying node speed, total number of nodes, reconfiguration duty time and desired coverage period has been carried out. The results have been presented and show promising prospects for the use of the grazing strategy. Work is under way to compare the performance of the latter with that of alternative mobility strategies.

The work presented here has also generated models which are realistic in essence and enable the WSN designer to test various mobility scenarios and evaluate their performance quantitatively ahead of implementation/development in the face of uncertainty–uncertainty associated with measurements and uncertainty resulting from constrained resources with a view to maximize system lifetime.

References


Matthew David Coles received a B.Eng (Hons) degree in electrical and electronic engineering from the University of Portsmouth (UK) in 2004. His research interests include mobile wireless sensor networks, biologically inspired navigational methods, and artificial intelligence for robotic systems. He is currently working towards a PhD at the University of Portsmouth.

Please cite this article in press as: M.D. Coles et al., A Bayesian network approach to a biologically inspired ..., Ad Hoc Netw. (2008), doi:10.1016/j.adhoc.2008.11.002
Djamel Azzī received an Ingéniorat D'État en Electronique from the University of Constantine (Algeria) in 1988. He also has an M.Sc. (1991) and Ph.D. (1993) from UMIST Manchester (UK) in Control Engineering. His early research involved developing predictive building office models for environmental control applications. His current research concentrates on wireless sensor networks and artificial intelligence for robotic systems. He currently hold a Principal Lecturer's post at the University of Portsmouth.

BARRY HAYNES was employed as an engineering apprentice with Baker Perkins Chemicals Ltd from 1968 to 1974 receiving a B.Sc. (hons) in Electro-Mechanical Engineering from Staffordshire Polytechnic in 1973. From 1974 to 1984 he was engaged as a systems project engineer with Westland Helicopters and the British Hovercraft Corporation and during that period received an M.Sc. in Control Engineering from Portsmouth Polytechnic. In 1984 he joined the staff of Portsmouth Polytechnic, subsequently the University of Portsmouth, and obtained a Ph.D. in Adaptive Control using AI techniques. He is currently a principal lecturer at the University of Portsmouth teaching Control System Design, Artificial Intelligence and Systems Analysis and Simulation. His research interests are robotics, and the application of artificial intelligence techniques in Data Analysis and Adaptive Control Systems.

ALAN HEWITT received a B.A. degree in Mathematics with Physics from the University of York, an M.Sc. in Electronics from the University of Southampton and Ph.D. from Portsmouth Polytechnic (now the University of Portsmouth). He is a senior lecturer at the University of Portsmouth, teaching Digital Communications and Broadcasting; Network Design and Management and Digital Signal Processing. He is a member of the Institution of Electronic and Electrical Engineers, the Audio Engineering Society and the Higher Education Academy.

Please cite this article in press as: M.D. Coles et al., A Bayesian network approach to a biologically inspired ..., Ad Hoc Netw. (2008), doi:10.1016/j.adhoc.2008.11.002
A self-healing mobile wireless sensor network using predictive reasoning

Barry Haynes, Mathew Coles and Djamel Azzi
Electronic and Computer Engineering, University of Portsmouth, Portsmouth, UK

Abstract
Purpose - The paper aims to investigate performance benefits associated with adopting a mobile wireless sensor network (WSN). Sensor nodes are generally energy constrained due to the latter being acquired from onboard battery cells. If one or more sensor nodes fail, possible coverage holes may be created which could invariably lead to a reduced network lifetime. The paper proposes that instead of rendering the entire WSN inoperative, sensor nodes should physically change position within the region of interest thus adaptively altering the WSN topology with a view of recovering from failures. This type of motion will be referred to as "self healing".

Design/methodology/approach - This paper presents a mobility scheme based on Bayesian networks for predictive reasoning (BayesMob) which is essentially a distributed self healing algorithm for coordinating physical relocation of sensor nodes. Using the algorithm, sensor nodes can predict the performance of the WSN in terms of coverage given that the node moves in a given direction. The evidence for this hypothesis is acquired from local neighborhood information.

Findings - The paper compares BayesMob with an alternative algorithm - Coverage Fidelity Algorithm - and shows that BayesMob maintains a higher level WSN coverage for a greater percentage of failures, thus increasing the useful lifetime of the WSN.

Research limitations/implications - The physical relocation of sensor nodes will incur energy overhead, therefore the tradeoffs between all application criteria should be investigated before implementation.

Originality/value - This paper presents a Bayesian network based motion coordination algorithm for WSN which repairs coverage holes caused by energy exhaustion and/or abrupt node failures.

Keywords Wireless, Sensors, Radio networks, Predictive process, Programming and algorithm theory

1. Introduction

Advances in sensor technology (in terms of size, power consumption, wireless communication and manufacturing costs) have enabled the prospect of deploying large quantities of sensor nodes to form a wireless sensor network (WSN). These networks are created by distributing large quantities of usually small, inexpensive sensor nodes over a geographical region of interest with a view to collect data relating to one or more variables. These nodes are primarily equipped with the means to sense, process and communicate data to other nodes and ultimately to a remote user(s). Sensor nodes may cooperate with their neighbors (within communication range) to form an ad hoc network. WSN topologies are generally dynamic and decentralized. Sensor nodes can also have mobility capabilities which enable them to physically relocate with relation to neighboring nodes and the environment in which they are situated. WSNs have a wide range of applications including military, environmental monitoring, health, home, space exploration, chemical processing, and disaster relief (Akyildiz et al., 2002).

The proposed application environments may be dynamic and the network designer may have limited knowledge of the region of interest. Also some of the proposed applications environment may be unmanned/unexplored terrain. In most cases the network designer would have little control over the exact deployment configuration. Such scenarios may include deployment via air drop.

The WSN deployment configuration is crucial to the network satisfying the performance criteria and operational lifetime. Even if the sensor nodes are deployed uniformly across the region of interest as time passes, sensor nodes may fail randomly due to energy exhaustion, malfunction or malicious destruction. Non-uniform traffic distribution and edge effects will directly influence the energy usage of the sensor nodes. The cumulative result of these factors may cause coverage holes and possibly detach a segment of the WSN. The implication of these failures may result in the WSN performance deteriorating thus preventing the performance criteria from being met. The net result of these failures is a reduced useful lifetime.

A proactive method of efficiently balancing the sensor nodes energy discharge rate (EDR) is to place nodes and assign tasks such that coverage holes are never formed in the WSN. This method may give an optimal solution but this approach to deployment is impractical for WSNs, as the
network designer would require a comprehensive knowledge of the application environment (Ganeriwal et al., 2004). Generally when considering the application environment this information is unavailable. Many of the foreseen applications are within regions where human intervention is not always possible. In such situations deployment is random and the network designer has limited influence over the exact node placement.

If coverage holes appear the WSN can be rendered inoperative and the remaining active sensor nodes would be wasted. We propose that when such holes are created the WSN topology is reorganized via physically relocating surrounding sensor nodes to repair the coverage hole. Therefore, the remaining resources will be utilized via a motion control algorithm and thus extend the useful lifetime of the WSN. This motion algorithm is deemed as self-healing. Figure 1 shows the self-healing mechanism, where sensor nodes physically move to repair coverage holes.

In this paper, the performance criterion considered is sensing coverage, which is defined as the fraction of the total intended area actually covered by the WSN (Meguerdichian et al., 2001). We propose an algorithm referred to as Bayesian network mobility (BayesMob) that uses mobility as an adaptive actuation facility for automated deployment and repair of the WSN with the sole objective of salvaging lost coverage. BayesMob incorporates a discrete Bayesian network which enables a sensor node to reason about the WSN coverage and actions required to maintain the required level of performance. It is a distributed mobility control algorithm which enables each sensor node to compute their optimum direction of movement with a view to maintain or increase the WSN coverage. The BayesMob algorithm predicts the probability of coverage increasing given that a node moves in a particular direction. Therefore, a sensor node may behave altruistically and expend energy on moving with a view to enhance the overall coverage of the WSN.

These predictions are derived from local neighborhood information. The accuracy of these predictions is dependant on the reconfiguration rate. The reconfiguration is the process of communicating with neighboring nodes to perform localization, navigation, neighbor discovery, synchronization and possibly generating routing tables. The reconfiguration process carries energy overheads due to the communication. Therefore, a trade-off exists between the reconfiguration rate and the WSN lifetime. The physical relocation of the sensor nodes also carries energy overheads due to the energy expended driving the motors and servos.

This approach to extend the network lifetime has already been proposed, by creating the Coverage Fidelity (CoFi) algorithm (Ganeriwal et al., 2004), (see related work). BayesMob differs in that sensor nodes predict the performance implications using a discrete Bayesian network. Also each sensor node coordinates their own motion and do not rely on the dying sensor node(s) for instructions.

In Section 2 we outline the related work that considers mobile WSN, and the inherent performance implications and benefits. Section 3 presents the BayesMob algorithm, and discusses the pros and cons when implementing the algorithm. The results of simulations are presented in Section 4, and finally Section 5 concludes the paper.

2. Related work

Mobility as a control primitive for self-deployment of WSNs has been investigated. For example, Wang et al. (2006) proposed a distributed self-deployment protocol which uses Voronoi diagrams to discover coverage holes caused by non-uniform deployment. The paper proposes three movement-assisted sensor deployment protocols which essentially relocate sensor nodes from densely deployed regions to areas with sparse coverage. Miao et al. (2006) proposed a self-deployment protocol for heterogeneous WSNs.

Using mobility as a control primitive to extend the network lifetime by balancing the EDR between all sensor nodes has also been investigated. For example, Rao and Biswas (2005) proposed a biologically inspired mobility model for balancing the energy overhead related to communication. The algorithm adopts a preventative approach to the creation of coverage holes due to node energy exhaustion. The mechanism was inspired by observing the natural grouping behavior of Emperor penguin communities in the Antarctic regions. The scheme however, does not consider node failure due to malfunction or malicious destruction.

Mobility as a control primitive for improving network coverage has also been investigated. For example, Ganeriwal et al. (2004) proposed a distributed CoFi algorithm that controls the relocation of a sensor node in order to repair coverage holes which are assumed to be a consequence of node failure.

Sekhar et al. (2005) proposed a dynamic coverage maintenance (DCM) scheme that also exploits the limited mobility of sensor nodes for active fault repair of the WSN. Four distributed rule-based DCM algorithms are presented which rely on local neighborhood topology information for coordinating the sensor relocation. The proposed DCM algorithms only relocate one hop neighbors of the dying sensor node, therefore, jeopardizing the effectiveness of the DCM scheme when limited redundancy is available within the vicinity of the dying node.
Butler and Rus (2003) proposed an event-based mobility scheme that coordinates the relocation of sensor nodes to areas that require a higher sensing resolution due to environmental, application and topology (nodes failing or moving) changes. Two distributed algorithms are proposed which use a history and history-free technique. The trade-offs between computation, memory and accuracy of the node's positions is also given. Using single dimensional mobility for improving sensing resolution and overcoming unpredictable environmental influences has also been investigated. Kansal et al. (2004b) presented a low complexity single dimension mobility strategy which has low energy actuation primitives. The nodes move along a single dimension to counteract a loss in coverage caused by environmental influences such as the presence of obstacles.

Low complexity mobility was also investigated by Kansal et al. (2004a) and Pon et al. (2005a, b) through the development of the Network InfoMechanical System (NIMS). NIMS's integrate distributed, embedded sensing and computing systems with infrastructure supported mobility. The papers suggest that the NIMS's motion capability enables the network to adapt to environment, application and topology changes.

3. Bayesian self healing algorithm

3.1 Motivation

The lifetime of a WSN is directly influenced by the ability of the network to satisfy the application criteria. The latter would generally define an acceptable level of coverage and connectivity that the WSN should maintain. Sensor nodes in the WSN do not die simultaneously for a variety of reasons. Failure may occur due to energy exhaustion, malicious destruction, or malfunction. The phenomena that the WSN is detecting may itself be non-uniform. For example, a traffic monitoring application may yield varying traffic densities in alternative areas of the region of interest. The sensor nodes closer to the base station are likely to die faster as they would forward greater number of data packets than nodes on the outer periphery (this phenomenon is referred to as an edge effect).

These networks are also generally deployed in inhospitable environments where nodes may have to tolerate extreme environmental conditions. Therefore, abrupt sensor node failure may occur. These sensor node failures may impair the WSN coverage and connectivity, via the creation of coverage holes. A coverage hole is defined as an area of the region not covered by the WSN, or a sub-section of the WSN being disconnected due to the node failure. When these coverage holes are created the WSN may fail to satisfy the application criteria and therefore the remaining energy within the WSN would be wasted.

We propose to utilize the remaining resources by relocating the energy proficient sensor nodes to repair the coverage holes. Therefore, the nodes may give up their current position and expend energy to relocate, effectively resulting in the WSN healing itself. However, the physical relocation of the sensor nodes will reduce their energy reserve, due to the energy necessary to drive motors and servos. Hence, a method is required to determine whether the energy expended due to physical relocation would yield performance benefits, to assist with the decision-making process.

The existing self-healing algorithms outlined in the related work section are centralized, where the failing sensor nodes coordinate the relocation of the neighboring nodes. Therefore, these algorithms only consider failures due to energy exhaustion where nodes have sufficient time and energy to coordinate the relocation of neighboring nodes. Also the algorithms only consider the relocation of one hop neighbors therefore, the recovery from a node failure may be jeopardized when considering limited local redundancy.

3.2 Bayesian mobility (BayesMob)

The BayesMob algorithm coordinates the sensor node relocation to maintain coverage in the event of node failures which cause coverage holes. BayesMob provides a distributed approach to motion control by adopting a Bayesian network, which uses probabilistic reasoning to determine the optimum node motion direction (in terms of coverage). The node effectively predicts the performance benefits and implications of moving in a given direction. A Bayesian network is a graphical structure that describes how each sensor node evaluates the uncertainty within the WSN (Korb and Nicholson, 2004).

BayesMob incorporates a discrete Bayesian network which predicts the probability of the WSN coverage increasing or remaining constant/unchanged given the WSN topology and sensor node's motion direction. The variables used in BayesMob are outlined in Table 1.

The Netica application software from Norsys (Netica, 2007) was used to create and validate the Bayesian networks. The Bayesian networks are used for predictive reasoning. BayesMob only considers discrete variables so the relationships between connected nodes are represented by a conditional probability table (CPT), (Coles, 2007). The CPT values are generally specified using statistical data from the system, and/or Bayes' theorem. The values within BayesMob have been specified by the designer with a view to achieve the desired response from the Bayesian networks.

The information used as evidence fed into the Bayesian networks is acquired from the local neighborhood. The predictions are based on a possible move in one of the cardinal directions north, south, east, or west.

BayesMob calculates the conditional probability of an increase in WSN coverage if the sensor node moves in one of the cardinal directions given evidence based on neighbors'
A self-healing mobile wireless sensor network

Barry Haynes, Mathew Cole and Djamel Azzouz

positions. The evidence is expressed as the probability of neighbors lying in each one of the cardinal directions and the probability that the closest neighbor's distance in each direction is less than the communication range \( R_c \). The BayesMob structure is shown in Figure 2. The joint probability function for BayesMob is given in equation (1):

\[
\begin{align*}
P(C_i, \theta, N_N, N_S, N_E, N_W) &= P(C_i, \theta) P(N_N) P(N_S) P(N_E) P(N_W) \times P(C_i | \theta, N_N, N_S, N_E, N_W) \quad (1) \\
&= \prod_{N_N, N_S, N_E, N_W} P(C_i, \theta, N_N, N_S, N_E, N_W)
\end{align*}
\]

The probability of coverage increasing given that the node moves in direction \( i \) \( P(C_i = T | \theta = i) \) is calculated by using joint probability theory and marginalization, which is given in equation (2):

\[
P(C_i = T | \theta = i) = \frac{P(C_i = T, \theta = i)}{P(\theta = i)} = \sum_{N_N, N_S, N_E, N_W} P(C_i, \theta, N_N, N_S, N_E, N_W) P(\theta = i) \times P(C_i | \theta, N_N, N_S, N_E, N_W)
\]

where \( P(N_N) \) to \( P(N_W) \) (probability of the need to move into each of the cardinal directions) is calculated using equation (3):

\[
P(N_i = T) = \sum_{A_i, d_i \in [T, P]} P(A_i) P(d_i) P(N_i = T | A_i, d_i)
\]

The belief in the evidence \( A_i \) and \( d_i \) are calculated from a contacts information table (CIT) local to each sensor node. The CIT stores all known information about neighboring nodes and also sensor nodes that route data packets through the node destined for the commander node. An example of this table is shown in Figure 3. The sensor nodes represent each of its neighbors' positions (motion) with two Gaussian distributions. Therefore, the CIT stores a mean \( \mu_1 \) and a standard deviation \( \sigma_1 \) for each of its neighbors' \( x \) and \( y \) coordinates. The CIT also contains additional information such as a timestamp associated with last communication and a relay count associated with the number of communication hops to the contact. The motion characteristics such as motion direction \( \theta \) and speed are also stored.

A sensor node must reconfigure at regular intervals to maintain a valid contacts table. The reconfiguration allows sensor nodes to exchange information with neighboring nodes for WSN maintenance purposes. This carries a communication overhead resulting in the need to extend the time between reconfigurations; this in turn, leads to increased uncertainty. Sensor nodes handle this uncertainty by representing contacts locations as Gaussian distributions which vary in time. This expresses the fact that information becomes less accurate the longer a node goes without being updated with contact information. In addition, the assumed Gaussian distributions provide a mechanism for quantifying the interdependencies between sensor nodes. When the contacts table is updated during reconfiguration, \( \mu_1, \sigma_1 \), store the contacts' current \( x \) and \( y \) coordinates, which represent its neighbor's mean position coordinates. Also the standard deviations associated with \( x \) and \( y \) \( \sigma_x = \sigma_y = 1 \text{ m} \) (arbitrarily initial value) which allows for localization errors.

During the period between reconfigurations and assuming the sensor node has not been in contact with its neighbors, it periodically increases \( \sigma_x, \sigma_y \) associated with each contact entry thus increasing the uncertainty associated with the contacts' coordinates. The standard deviation increase is calculated using equation (4):

\[
\sigma = \sigma + v \cdot t_{update}
\]

where \( v \) is the node speed and \( t_{update} \) table update duty cycle time.

### 3.3 Probability of neighbor distance being less than the communication range

The Gaussian distributions associated with each of the contacts' coordinates are used to calculate the probability that neighbor's distance is less than the communication range \( P(d_i(j = T) \) where \( j = 1 \ldots n \) and \( n \) equals the total number of neighbors from the contacts table). If \( x \) and \( y \) are independent Gaussian random variables with nonzero means then the distance \( z = \sqrt{x^2 + y^2} \) has a probability density function \( f_d(z) \) which can be represented by a Rician distribution, which is given in equation (5):

\[
f_d(z) = \frac{\mu e^{-(z^2 + \mu^2)/2\sigma^2}}{\sigma^2} I_0 \left( \frac{2\mu z}{\sigma^2} \right)
\]

where:

\[
\mu = \sqrt{\mu_x^2 + \mu_y^2} \quad \theta = \tan^{-1} \left( \frac{Y}{X} \right) \quad \sigma = \sigma_x = \sigma_y
\]

\[
\mu_x = \mu \cos \phi \quad \mu_y = \mu \sin \phi
\]

and:

\[
I_0(\eta) = \frac{1}{2\pi} \int_0^{2\pi} e^{i\cos(\phi - \sigma)} d\phi = \frac{1}{\pi} \int_0^{\pi} \eta e^{i\cos \phi} d\phi
\]
Equation (6) is the modified Bessel function of the first kind and zeroth order.

\[ P(d_i(j) = T) \]

in the cardinal direction \( i \) is approximated by numerically integrating the Rician distribution between zero and the \( C_i \).

### 3.4 Probability of a neighbor lying in each of the cardinal directions

\[ P(A_i(j) = T) \] is approximated using Algorithm 1 because the p.d.f \( f_\theta(\theta) \) of the angle \( \theta = \tan^{-1}(y/x) \) is intractable when \( x \) and \( y \) are independent Gaussian random variables with nonzero means:

The probabilities \( P(d_i(j) = T) \) and \( P(A_i(j) = T) \) are aggregated to approximate the evidence for BayesMob. This is shown in equations (7) and (8). \( P(d_i(j) = T) \) is scaled with respect to the belief that the neighbor is in direction \( i \):

\[
P(A_i = T) = 1 - \prod_{j=1}^{n} P(A_i(j) = i)
\]

\[
P(d_i = T) = 1 - \prod_{j=1}^{n} P(d_i(j)) \times P(A_i(j) = i)
\]

**Algorithm 1**

Approximate \( P(A_i(j) = T) \) \( x, y \) = sensor node’s coordinates, and \( \sigma = \sigma_x = \sigma_y \):

\[
\text{distance} = \sqrt{(x - x_j)^2 + (y - y_j)^2}
\]

- mean distance from neighbour

if \( \sigma < (4 \times \text{distance}) \)

Assume the \( f_\theta(\theta) \) p.d.f is represented by a Gaussian distribution with:

\[
\sigma_\theta(\theta) = 100 - 100e^{-|\theta/\mu_\text{distance}|}
\]

\[
\mu_\theta = \tan^{-1}(\frac{\mu_x}{\mu_y})
\]

/*Probability of direction is approximated by integrating between

45 to 135 and 405 to 495 degrees for north

135 to 225 and -225 to -135 degrees for west

225 to 315 and -135 to -45 degrees for south

315 to 360 and 0 to 45 and -45 to 0 and 360 to 405 degrees for east*/

else

Assume \( f_\theta(\theta) \) is represented by a uniform distribution

/*assume a uniform distribution between 0 and 360 degrees*/

### 3.5 Selecting the motion status and direction

The sensor node selects the motion direction by evaluating probability of coverage increase \( P(C_i) \) and determining which direction yields the maximum probability of the coverage being maintained or increased and also evaluating the adjacent cardinal direction probabilities. If \( P(C_i) \) is greater than a specified threshold the node moves in the direction determined by Algorithm 2, else it will remain static. The threshold defines the motion status of the sensor node by setting the sensitivity of the BayesMob algorithm. If this value is too low the sensor node will continuously move and thus exhaust their limited energy reserve. Alternatively setting the threshold too high will prevent the sensor node from moving to repair coverage holes. Therefore, this value specifies how responsive a sensor node is to a coverage hole. The threshold was set to 35 per cent through trial and error testing.

**Algorithm 2**

Selecting motion direction

Calculate the difference between adjacent cardinal directions probabilities \( P(C_i = 1) \) if difference < 2 per cent and difference > -2 per cent motion direction = direction which yields maximum probability

else motion direction = direction which yields maximum probability + difference between adjacent cardinal direction probabilities

### 4. Simulation and results

#### 4.1 Simulation set up

All simulations have been generated using a custom-built Matlab based WSN simulator. The geographical region of interest was set to a 100 \( \times \) 100 m area. Every sensor node is equipped with motion capabilities. Table II details the settings of the simulation parameters. The sensor nodes were configured to generate and transmit data packets destined for the commander node in a duty period of 2 min.

The reconfiguration duty time defines the period between neighbor communications used to update localization information within the CIT. The former was set to 2 min. We assumed a perfect medium access control (MAC) protocol, therefore the practical implications associated with communication were not considered. Having said that, in a practical implementation the repeat mechanism built into the MAC protocol would ensure successful transmissions at the cost of negligible packet delay. The latter is of the order of hundreds of milliseconds (upper estimate) which is negligible when compared with the time duration of the system at hand.

Two types of deployment strategies have been tested; fixed and random deployments. Under the fixed approach the sensor nodes would be placed manually, thus ensuring a uniform distribution of the sensor nodes and maximum coverage. Alternatively the random approach positions the sensor nodes following a uniform distribution.

Table simulations have been carried out for a range of sensor node densities (64, 81 and 100 nodes for the 100 \( \times \) 100 m\(^2\)). The latter values were selected to provide a uniform spatial distribution across the region of interest (which is square shaped), whilst adopting a fixed deployment approach. For example, a deployment of 64 nodes which are evenly

**Table II Simulation parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Rationale</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication range</td>
<td>20 m</td>
<td>Typical ranges observed in external environments with Berkeley motes</td>
</tr>
<tr>
<td>Sensing range</td>
<td>10 m</td>
<td>Sensing range is half the communication range</td>
</tr>
<tr>
<td>Mobility cost</td>
<td>17.8758 J/m</td>
<td>Capacity of 6 v alkaline battery</td>
</tr>
<tr>
<td>Total initial energy</td>
<td>87,480 J</td>
<td></td>
</tr>
</tbody>
</table>
A self-healing mobile wireless sensor network
Barry Haynes, Mathew Coles and Djamel Azziz

distributed across the region of interest yields 8 x 8 node grid. The simulations evaluate the WSN coverage for the CoFi algorithm (Ganeriwal et al., 2004) and whilst adopting BayesMob. We present results that compare both approaches in terms of coverage loss and motion energy overheads. The node failures are induced via a uniform distribution and occur at 30 min intervals which attempts to emulate abrupt failures under volatile application environment conditions.

4.2 Fixed deployment
First we consider a fixed deployment scenario, under which the sensor nodes are uniformly distributed over the region of interest to provide 100 per cent coverage. Figures 4-6 show the respective coverage loss plots for a deployment of 64, 81, and 100 nodes. These results show that the physical relocation of the sensor nodes under the BayesMob algorithm will sustain the WSN coverage for an increased percentage of node failures. This observation becomes more apparent as the sensor node density increases. For example, if the application criterion specifies a maximum coverage loss of 20 per cent, with CoFi the WSN could tolerate 35, 43, and 44 per cent of nodes failing for 64, 81, and 100 nodes deployed, respectively. Using the BayesMob algorithm the WSN tolerates 45, 58, and 66 per cent of nodes failing for the same node densities.

The sharp increases in coverage loss observed in all figures are due to a subsection of the WSN becoming disconnected (loss in connectivity). The BayesMob fails to recover the coverage loss when the number of sensor nodes deployed is not sufficient to cover the region of interest.

4.3 Random deployment
Here, the sensor nodes are deployed randomly according to a uniform distribution over the region of interest. Figures 7-9 show the respective coverage loss plots. The results of the simulations show that the BayesMob algorithm tolerates 50, 60 and 68 per cent of nodes failing (for 64, 81 and 100 nodes deployed, respectively) before the coverage loss criterion is exceeded. Under the same conditions, the CoFi tolerates only 40, 43 and 57 per cent node failure before the coverage criterion is breached.

4.4 Motion energy overheads
The percentage of total energy available to the WSN used for motion for both strategies (BayesMob and CoFi) has been evaluated and is represented in Figure 10 as a function of the percentage of dead nodes. The total energy is calculated by multiplying the total number of sensor nodes by their individual energy reserves (total battery capacity) throughout the length of the simulation. The motion energy costs associated with the BayesMob exceed those of CoFi. When 80 per cent of the WSN has failed the motion energy for BayesMob algorithm is approximately 1.7 per cent of the total energy. The corresponding motion energy for the CoFi algorithm under the same conditions is approximately 0.025 per cent.

The motion energy for BayesMob rapidly increases as the percentage of nodes fail. This is due to sensor nodes moving...
A self-healing mobile wireless sensor network

Barry Haynes, Mathew Coles and Djamel Azzi

5. Conclusion

This paper has shown how, through the use of a Bayesian network based mobility scheme, a WSN can be imparted self-healing properties with regards to coverage losses induced by energy exhaustion and abrupt node failures (such as those caused by malicious interference). The paper has described how sensor nodes operating according to BayesMob can predict WSN coverage variations using only local information and compute a direction of motion with a view to maximize or maintain the coverage of the network. The results of comparing BayesMob with the CoFi algorithm have been presented and show that BayesMob maintains coverage for a greater percentage of dead nodes, albeit at the expense of increased energy overheads — which may be acceptable given the application coverage requirements.

BayesMob is a promising technique which is currently being extended to coordinate the motion of nodes within a mobile WSN to improve additional performance criteria such as lifetime (for example edge effect minimisation where a sensor node would base their next move on the estimated EDR of neighboring nodes). It is also planned to create mobility strategies which would provide varying degrees of sensing resolution across the region of interest, dependent on varying application criteria and operating conditions.
A self-healing mobile wireless sensor network

Barry Haynes, Mathew Coles and Djamel Azzid

References


Netica (2007), application software and instructions, available at: www norsys.com


Further reading


Corresponding author

Mathew Coles can be contacted at: matt.coles@port.ac.uk

To purchase reprints of this article please e-mail: reprints@emeraldinsight.com
Or visit our web site for further details: www.emeraldinsight.com/reprints

Sensor Review

Volume 28 · Number 4 · 2008 · 326–333
Appendix B

SOURCE CODE

The source code for the WSN simulator and mobility algorithms presented are available on the CD.
Appendix C

ANGLE PROBABILITY DENSITY FUNCTION

This appendix outlines the workings for validating the statement that the angle p.d.f. from two random variables (x and y) is computationally heavy to compute when x and y (which represent the neighbours coordinates from the CIT) are independent normal random variables with non-zero means. Given the two random variables x and y and a function $\tan^{-1}(y/x)$, a new random variable $\theta$ is formed.

$$\theta = \tan^{-1}\left(\frac{y}{x}\right)$$

(C.1)

Given the joint p.d.f. $f_{xy}(x,y)$ the $f_{\theta}(\theta)$ (the p.d.f. of $\theta$) is obtained by first determining the function $(y/x)$ and forming a new random variable $z$, given (C.2).

$$z = \frac{y}{x} = \tan \theta$$

(C.2)

The proof below was taken from Papoulis [96] and shows that the p.d.f. associated with $z$ is intractable when dealing with a joint p.d.f. $f_{xy}(x,y)$ with non-zero means.

Let $z = y/x$ then in order to determine $f_{z}(z)$ (the p.d.f. of $z$) we have

$$F_{z}(z) = P\{y/x \leq z\} = P\{(x,y) \in D_{z}\}$$

(C.3)

Where $D_{z}$ in the $xy$ plane represents the region where the inequality $y/x \leq z$ is satisfied, for example see Figure C.1.
The inequality $y/x \leq z$ can be rewritten as $y \leq xz$ if $x > 0$, and $y \geq xz$ if $x < 0$. Hence the event $\{y/x \leq z\}$ needs to be conditioned by the event $A = \{x > 0\}$ and its complement $\bar{A}$.

Since $A \cup \bar{A} = \Omega$, because of the Partition Theorem, we have:

$$P\{y/x \leq z\} = P\{y/x \leq z \cap (A \cup \bar{A})\}$$
$$= P\{y/x \leq z, x > 0\} + P\{y/x \leq z, x < 0\}$$
$$= P\{y \leq xz, x > 0\} + P\{y \geq xz, x < 0\}$$

Integrating over these two regions we get

$$F(z) = \int_{y=0}^{\infty} \int_{x=-\infty}^{\infty} f_{xy}(x,y) \, dx \, dy + \int_{y=0}^{\infty} \int_{x=0}^{\infty} f_{xy}(x,y) \, dx \, dy$$

Differentiation with respect to $z$ gives $f_z(z)$ (the p.d.f. of $z$)

$$f_z(z) = \tan \theta = \int_{-\infty}^{\infty} x \, f_{xy}(xz,x) \, dx$$

$$f_{xy}(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{[(x-\mu_x)^2+(y-\mu_y)^2]}{2\sigma^2}}$$

The $x$ and $y$ are joint Gaussian random variables with non-zero means $\mu_x \neq \mu_y \neq 0$ (C.7) and since $x$ and $y$ are assumed to be uncorrelated and assuming that $\sigma_x = \sigma_y = \sigma$.

Replacing (C.7) into (C.6) yields:

$$f_z(z) = \frac{1}{2\pi\sigma^2} \int_{-\infty}^{\infty} x \left| e^{-\frac{[(x-\mu_x)^2+(x-\mu_y)^2]}{2\sigma^2}} \right| \, dx$$

Figure C.1 One function of two random variables (region where inequality $y/x \leq z$ is satisfied)
This integration in (C.9) is only tractable through numerical integration due to the exponential of the quadratic function and the $x$ term. This is computationally heavy and would require each node to carry a lookup table. Therefore an algorithm which approximates $f_s(z) = \tan \theta$ using a normal distribution was developed, presented in section 5.4.3.1.2.
Appendix D

CA ALGORITHM MAXIMUM ERROR

This appendix characterises the maximum error associated with the CA algorithm. It demonstrates the coverage error increases with respect to the WSN size making the algorithm not scalable.

The CA algorithm assumes that the WSN coverage is a circular area. The area of this circle and associated radius are obtained using equations (D.1) and (D.2).

\[
\text{Coverage}_{\text{area}} = \pi R^2 = \left( \mu_d \left( \sqrt{i_n} - 1 \right) + 2S_n \right)^2
\]  

\[
\text{Coverage}_{\text{radius}} = R = \sqrt{\frac{\text{Coverage}_{\text{area}}}{\pi}}
\]  

where \( \text{Coverage}_{\text{area}} \) is the approximated coverage region area, \( R \) is the radius of the approximated WSN coverage region, \( i_n \) equals the number sensor nodes that form the WSN, \( S_n \) is the sensing range, and \( \mu_d \) is the mean neighbour distance.

Figure D.1 shows the coverage approximation error. The error associated with respect to positioning the coverage can be divided into: (i) the area which is covered when it is actually not covered by the WSN, (ii) the area which is not covered when it is actually covered by the WSN. Where \( x_n \) and \( y_n \) are the centre coordinates of the approximated coverage region, \( x_a \) and \( y_a \) are the centre coordinates of the actual coverage region and \( R_a \) is the radius of the actual coverage region.
The area of the symmetric lens resulting from the circles intersect (area which is correctly approximated) is calculated using (D.3), see [97]. Equation (D.3) uses the formula for calculating the area of a circular segment of radius $R_a$ (or $R_c$) and a distance of $d_a$ (or $d_n$) shown in Figure D.1.

$$A = A(R_a, d_a) + A(R_c, d_n)$$

$$= R_a^2 \cos^{-1}\left(\frac{d_a}{R_a}\right) - d_a\sqrt{R_a^2 - d_a^2} + R_c^2 \cos^{-1}\left(\frac{d_n}{R_c}\right) - d_n\sqrt{R_c^2 - d_n^2}$$

\[(D.3)\]

where

$$d_a = d_{\text{error}} - d_n$$

\[(D.4)\]

The coverage error, which equates the sum of the convex areas either side of the lens in Figure D.1, is calculated using (D.5).

$$\text{Coverage}_{\text{error}} = \pi R_c^2 - R_c^2 \cos^{-1}\left(\frac{d_{\text{error}} - d_a}{R_c}\right) - (d_{\text{error}} - d_a)\sqrt{R_c^2 - (d_{\text{error}} - d_a)^2}$$

$$+ \pi R_a^2 - R_a^2 \cos^{-1}\left(\frac{d_{\text{error}} - d_n}{R_a}\right) - (d_{\text{error}} - d_n)\sqrt{R_a^2 - (d_{\text{error}} - d_n)^2}$$

\[(D.5)\]
The coverage error is dependent on the distance error \( d_{\text{error}} \) which is calculated via equation (D.6) and the radius of the approximated coverage error \( R_c \).

\[
d_{\text{error}} = \sqrt{(x_a - x_n)^2 + (y_a - y_n)^2}
\]  
(D.6)

The maximum distance error occurs when the sensor nodes are all positioned in a line and the sensor node (which executes the CA algorithm) at the end only has one neighbour and has no additional contact information. Therefore the CA algorithm positions the approximated coverage region, such that the sensor node is on the periphery and in the direction of the remaining nodes. The distance error (between actual and approximated centre of the WSN) is given in (D.7). Where \( R_c \) is the approximated coverage radius (equation (D.2)), \( \mu_d \) is the distance between its neighbour and \( S_r \) is the homogeneous sensing range.

\[
d_{\text{error}} = \mu_d - (R_c - S_r) + \frac{\mu_d(i_a - 3)}{2}
\]  
(D.7)

This analysis demonstrates that the distance error is dependent on the sensing range, mean distance from neighbours and most importantly the number of sensor nodes that form the WSN. The CA algorithm therefore has scalability issues.
Appendix E

CONDITIONAL PROBABILITY TABLES

E.1  BNGRAZ CPT

This section outlines the CPTs for the BNGRAZ algorithm. All variables are defined in Table 5.1. The values in these tables were generated from expert knowledge.

E.1.1  Bn1 Probability of a Decrease in Connectivity

| $A_i$ | $d_i$ | $P(N_i = T | A_i, d_i)$ | $P(N_i = F | A_i, d_i)$ |
|-------|-------|--------------------------|--------------------------|
| T     | T     | 0.024                    | 0.976                    |
| T     | F     | 0.976                    | 0.024                    |
| F     | T     | 0.005                    | 0.995                    |
| F     | F     | 0.005                    | 0.995                    |
Table E.2 Bn1 C PT

| \(N_w\) | \(N_s\) | \(N_e\) | \(N_w\) | \(P(C_i = T | \theta, N_s, N_e, N_w)\) | \(P(C_i = F | \theta, N_s, N_e, N_w)\) |
|--------|--------|--------|--------|----------------------------|----------------------------|
| T      | T      | T      | T      | 0.3                       | 0.3                       |
| T      | T      | T      | S      | 0.3                       | 0.3                       |
| T      | T      | T      | E      | 0.3                       | 0.3                       |
| T      | T      | T      | W      | 0.3                       | 0.3                       |
| T      | T      | T      | N      | 0.3                       | 0.3                       |
| T      | T      | T      | B      | 0.3                       | 0.3                       |
| T      | T      | T      | W      | 0.3                       | 0.3                       |
| T      | T      | F      | T      | 0.3                       | 0.3                       |
| T      | T      | F      | S      | 0.3                       | 0.3                       |
| T      | T      | F      | E      | 0.3                       | 0.3                       |
| T      | T      | F      | W      | 0.3                       | 0.3                       |
| T      | T      | F      | N      | 0.3                       | 0.3                       |
| T      | T      | F      | B      | 0.3                       | 0.3                       |
| T      | T      | T      | E      | 0.3                       | 0.3                       |
| T      | T      | T      | W      | 0.3                       | 0.3                       |
| T      | T      | F      | T      | 0.3                       | 0.3                       |
| T      | T      | F      | S      | 0.3                       | 0.3                       |
| T      | T      | F      | E      | 0.3                       | 0.3                       |
| T      | T      | F      | W      | 0.3                       | 0.3                       |
| T      | T      | F      | N      | 0.3                       | 0.3                       |
| T      | T      | F      | B      | 0.3                       | 0.3                       |
| T      | T      | T      | E      | 0.3                       | 0.3                       |
| T      | T      | T      | W      | 0.3                       | 0.3                       |
| T      | T      | F      | T      | 0.3                       | 0.3                       |
| T      | T      | F      | S      | 0.3                       | 0.3                       |
| T      | T      | F      | E      | 0.3                       | 0.3                       |
| T      | T      | F      | W      | 0.3                       | 0.3                       |
| T      | T      | F      | N      | 0.3                       | 0.3                       |
| T      | T      | F      | B      | 0.3                       | 0.3                       |
| T      | T      | T      | E      | 0.3                       | 0.3                       |
| T      | T      | T      | W      | 0.3                       | 0.3                       |
| T      | T      | F      | T      | 0.3                       | 0.3                       |
| T      | T      | F      | S      | 0.3                       | 0.3                       |
| T      | T      | F      | E      | 0.3                       | 0.3                       |
| T      | T      | F      | W      | 0.3                       | 0.3                       |
| F      | F      | T      | T      | 0.3                       | 0.3                       |
| F      | F      | T      | S      | 0.3                       | 0.3                       |
| F      | F      | T      | E      | 0.3                       | 0.3                       |
| F      | F      | T      | W      | 0.3                       | 0.3                       |
| F      | F      | T      | N      | 0.3                       | 0.3                       |
| F      | F      | T      | B      | 0.3                       | 0.3                       |
| F      | F      | T      | E      | 0.3                       | 0.3                       |
| F      | F      | T      | W      | 0.3                       | 0.3                       |
| F      | F      | T      | N      | 0.3                       | 0.3                       |
| F      | F      | T      | S      | 0.3                       | 0.3                       |
| F      | F      | T      | E      | 0.3                       | 0.3                       |
| F      | F      | T      | W      | 0.3                       | 0.3                       |
| F      | F      | T      | N      | 0.3                       | 0.3                       |
| F      | F      | T      | S      | 0.3                       | 0.3                       |
| F      | F      | T      | E      | 0.3                       | 0.3                       |
| F      | F      | T      | W      | 0.3                       | 0.3                       |
| F      | F      | T      | N      | 0.3                       | 0.3                       |
| F      | F      | T      | S      | 0.3                       | 0.3                       |
| F      | F      | T      | E      | 0.3                       | 0.3                       |
| F      | F      | T      | W      | 0.3                       | 0.3                       |
### E.1.2 Bn2 probability of discovering un-serviced area

Table E.3 Bn2 D1 CPT

| $U_N$ | $U_S$ | $U_E$ | $U_W$ | $\theta_m$ | $P(D=T | \theta_m, U_N, U_S, U_E, U_W)$ | $P(D=F | \theta_m, U_N, U_S, U_E, U_W)$ |
|-------|-------|-------|-------|------------|---------------------------------|---------------------------------|
| T     | T     | T     | T     | N          | 0.5                             | 0.5                             |
| T     | T     | T     | T     | S          | 0.5                             | 0.5                             |
| F     | T     | T     | T     | S          | 0.5                             | 0.5                             |
| T     | T     | T     | T     | N          | 0.5                             | 0.5                             |
| T     | T     | T     | T     | S          | 0.9                             | 0.1                             |
| T     | T     | T     | F     | E          | 0.9                             | 0.1                             |
| T     | T     | T     | F     | W          | 0.9                             | 0.1                             |
| T     | F     | T     | T     | N          | 0.9                             | 0.1                             |
| T     | F     | T     | T     | S          | 0.5                             | 0.1                             |
| T     | T     | F     | T     | W          | 0.9                             | 0.1                             |
| T     | T     | F     | F     | N          | 0.9                             | 0.1                             |
| T     | T     | F     | F     | S          | 0.9                             | 0.1                             |
| T     | T     | F     | F     | E          | 0.2                             | 0.8                             |
| T     | T     | F     | F     | W          | 0.2                             | 0.8                             |
| T     | F     | T     | T     | N          | 0.9                             | 0.1                             |
| T     | F     | T     | T     | S          | 0.9                             | 0.1                             |
| T     | F     | T     | T     | E          | 0.9                             | 0.1                             |
| T     | F     | T     | T     | W          | 0.9                             | 0.1                             |
| F     | T     | T     | T     | N          | 0.9                             | 0.1                             |
| F     | T     | T     | T     | N          | 0.9                             | 0.1                             |
| F     | F     | T     | T     | S          | 0.9                             | 0.1                             |
| F     | F     | T     | T     | E          | 0.9                             | 0.1                             |
| F     | F     | T     | T     | W          | 0.9                             | 0.1                             |
| F     | T     | F     | T     | N          | 0.9                             | 0.1                             |
| F     | T     | F     | T     | S          | 0.9                             | 0.1                             |
| F     | T     | F     | T     | E          | 0.9                             | 0.1                             |
| F     | T     | F     | T     | W          | 0.9                             | 0.1                             |
| F     | T     | F     | F     | N          | 0.9                             | 0.1                             |
| F     | T     | F     | F     | S          | 0.9                             | 0.1                             |
| F     | T     | F     | F     | E          | 0.9                             | 0.1                             |
| F     | T     | F     | F     | W          | 0.9                             | 0.1                             |
| F     | T     | F     | F     | N          | 0.9                             | 0.1                             |
| F     | F     | T     | F     | N          | 0.9                             | 0.1                             |
| F     | F     | T     | F     | S          | 0.9                             | 0.1                             |
| F     | F     | T     | F     | E          | 0.9                             | 0.1                             |
| F     | F     | T     | F     | W          | 0.9                             | 0.1                             |
| F     | F     | T     | F     | N          | 0.9                             | 0.1                             |
| F     | F     | T     | F     | S          | 0.9                             | 0.1                             |
| F     | F     | T     | F     | E          | 0.9                             | 0.1                             |
| F     | F     | T     | F     | W          | 0.9                             | 0.1                             |
| F     | F     | T     | F     | N          | 0.9                             | 0.1                             |
| F     | F     | T     | F     | S          | 0.9                             | 0.1                             |
| F     | F     | T     | F     | E          | 0.9                             | 0.1                             |
| F     | F     | T     | F     | W          | 0.9                             | 0.1                             |
| F     | F     | T     | F     | N          | 0.9                             | 0.1                             |
| F     | F     | T     | F     | S          | 0.9                             | 0.1                             |
| F     | F     | T     | F     | E          | 0.9                             | 0.1                             |
| F     | F     | T     | F     | W          | 0.9                             | 0.1                             |
| F     | F     | T     | F     | N          | 0.9                             | 0.1                             |
| F     | F     | T     | F     | S          | 0.9                             | 0.1                             |
| F     | F     | T     | F     | E          | 0.9                             | 0.1                             |
| F     | F     | T     | F     | W          | 0.9                             | 0.1                             |
| F     | F     | T     | F     | N          | 0.9                             | 0.1                             |
| F     | F     | T     | F     | S          | 0.9                             | 0.1                             |
| F     | F     | T     | F     | E          | 0.9                             | 0.1                             |
| F     | F     | T     | F     | W          | 0.9                             | 0.1                             |
| F     | F     | T     | F     | N          | 0.9                             | 0.1                             |
| F     | F     | T     | F     | S          | 0.9                             | 0.1                             |
| F     | F     | T     | F     | E          | 0.9                             | 0.1                             |
E.1.3 Bn3 Probability of Maintaining or Increasing Performance

### Table E.4 Bn3 $N_{N_{1}}..N_{N_{w}}$ CPT

| $D_{i}$ | $C_{i}$ | $P(N_{i} = T | C_{i}, D_{i})$ | $P(N_{i} = F | C_{i}, D_{i})$ |
|---------|---------|-------------------------------|-------------------------------|
| T       | T       | 0.025                         | 0.975                         |
| T       | F       | 0.975                         | 0.025                         |
| F       | T       | 0.025                         | 0.975                         |
| F       | F       | 0.10                          | 0.90                          |

### Table E.5 Bn3 $O_{d}$ CPT

| $N_{N}$ | $N_{S}$ | $N_{E}$ | $N_{W}$ | $P(O_{d} = N | N_{N}, N_{S}, N_{E}, N_{W})$ | $P(O_{d} = S | N_{N}, N_{S}, N_{E}, N_{W})$ | $P(O_{d} = E | N_{N}, N_{S}, N_{E}, N_{W})$ | $P(O_{d} = W | N_{N}, N_{S}, N_{E}, N_{W})$ |
|---------|---------|---------|---------|-----------------------------------------|-----------------------------------------|-----------------------------------------|-----------------------------------------|
| T       | T       | T       | T       | 0.25                                    | 0.25                                    | 0.25                                    | 0.25                                    |
| T       | T       | T       | F       | 0.31                                    | 0.31                                    | 0.37                                    | 0.01                                    |
| T       | T       | F       | T       | 0.31                                    | 0.31                                    | 0.01                                    | 0.37                                    |
| T       | T       | F       | F       | 0.48                                    | 0.48                                    | 0.02                                    | 0.02                                    |
| T       | F       | T       | T       | 0.37                                    | 0.01                                    | 0.31                                    | 0.31                                    |
| T       | F       | T       | F       | 0.48                                    | 0.02                                    | 0.48                                    | 0.02                                    |
| T       | F       | F       | T       | 0.48                                    | 0.02                                    | 0.02                                    | 0.48                                    |
| T       | F       | F       | F       | 0.94                                    | 0.02                                    | 0.02                                    | 0.02                                    |
| F       | T       | T       | T       | 0.01                                    | 0.37                                    | 0.31                                    | 0.31                                    |
| F       | T       | T       | F       | 0.02                                    | 0.48                                    | 0.48                                    | 0.02                                    |
| F       | T       | F       | T       | 0.02                                    | 0.48                                    | 0.02                                    | 0.48                                    |
| F       | T       | F       | F       | 0.02                                    | 0.94                                    | 0.02                                    | 0.02                                    |
| F       | F       | T       | T       | 0.02                                    | 0.02                                    | 0.48                                    | 0.48                                    |
| F       | F       | T       | F       | 0.02                                    | 0.02                                    | 0.94                                    | 0.02                                    |
| F       | F       | F       | T       | 0.02                                    | 0.02                                    | 0.02                                    | 0.94                                    |
| F       | F       | F       | F       | 0.25                                    | 0.25                                    | 0.25                                    | 0.25                                    |
E.2 BayesMob CPT

This document outlines the CPTs for the BayesMob algorithm. All variables are defined in Table 6.1. The values in these tables were generated from expert knowledge.

Table E.6 $N_N..N_w$ CPT

| $A_i$ | $d_i$ | $P(N_i = T | A_i, d_i)$ | $P(N_i = F | A_i, d_i)$ |
|-------|-------|--------------------------|--------------------------|
| T     | T     | 0.024                    | 0.976                    |
| T     | F     | 0.35                     | 0.65                     |
| F     | T     | 0.45                     | 0.55                     |
| F     | F     | 0.976                    | 0.024                    |
Table E.7 Bn1 \( C_i \) CPT

<table>
<thead>
<tr>
<th>( N_N )</th>
<th>( N_S )</th>
<th>( N_T )</th>
<th>( N_W )</th>
<th>( \theta_n )</th>
<th>( P(C_i = T \mid \theta_n, N_N, N_S, N_T, N_W) )</th>
<th>( P(C_i = F \mid \theta_n, N_N, N_S, N_T, N_W) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>N</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>S</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>E</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>W</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>N</td>
<td>N</td>
<td>0.2</td>
<td>0.8</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>S</td>
<td>N</td>
<td>0.2</td>
<td>0.8</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>E</td>
<td>N</td>
<td>0.2</td>
<td>0.8</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>W</td>
<td>N</td>
<td>0.2</td>
<td>0.8</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>N</td>
<td>N</td>
<td>0.2</td>
<td>0.8</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>S</td>
<td>N</td>
<td>0.2</td>
<td>0.8</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>E</td>
<td>N</td>
<td>0.2</td>
<td>0.8</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>W</td>
<td>N</td>
<td>0.2</td>
<td>0.8</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>N</td>
<td>N</td>
<td>0.2</td>
<td>0.8</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>S</td>
<td>N</td>
<td>0.2</td>
<td>0.8</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>E</td>
<td>N</td>
<td>0.2</td>
<td>0.8</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>W</td>
<td>N</td>
<td>0.2</td>
<td>0.8</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>N</td>
<td>N</td>
<td>0.1</td>
<td>0.9</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>S</td>
<td>N</td>
<td>0.1</td>
<td>0.9</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>E</td>
<td>N</td>
<td>0.1</td>
<td>0.9</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>W</td>
<td>N</td>
<td>0.1</td>
<td>0.9</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>N</td>
<td>N</td>
<td>0.9</td>
<td>0.1</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>S</td>
<td>N</td>
<td>0.9</td>
<td>0.1</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>E</td>
<td>N</td>
<td>0.9</td>
<td>0.1</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>W</td>
<td>N</td>
<td>0.9</td>
<td>0.1</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>N</td>
<td>N</td>
<td>0.8</td>
<td>0.2</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>S</td>
<td>N</td>
<td>0.8</td>
<td>0.2</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>E</td>
<td>N</td>
<td>0.8</td>
<td>0.2</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>W</td>
<td>N</td>
<td>0.8</td>
<td>0.2</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>N</td>
<td>N</td>
<td>0.8</td>
<td>0.2</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>S</td>
<td>N</td>
<td>0.8</td>
<td>0.2</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>E</td>
<td>N</td>
<td>0.8</td>
<td>0.2</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>W</td>
<td>N</td>
<td>0.8</td>
<td>0.2</td>
</tr>
</tbody>
</table>

190
Appendix F

ANGLE DISTRIBUTION APPROXIMATION

This appendix presents the angle distributions, generated from two jointly random normal distributions with non-zero means, for a selection of x (or y) SD values. The mean x and y values for the results given below were 10 m. Each plot includes the angle distribution with a normal fit. These were generated using the \texttt{x\_y\_angle.m} function, source code given in Appendix B, and the Matlab distribution fit tool (dfittool). Figure F.1 shows the flow diagram for \texttt{x\_y\_angle.m}. The respective angle SD against the x (or y) SD value are shown in Figure F.28.

![Flow diagram for \texttt{x\_y\_angle.m} function](image)

Figure F.1 Flow diagram for \texttt{x\_y\_angle.m} function
APPENDIX F - ANGLE DISTRIBUTION APPROXIMATION

Figure F.2 Angle distribution with normal fit (x or y) SD = 1 m

Figure F.3 Angle distribution with normal fit (x or y) SD = 1.5 m

Figure F.4 Angle distribution with normal fit (x or y) SD = 2 m

Figure F.5 Angle distribution with normal fit (x or y) SD = 2.5 m

Figure F.6 Angle distribution with normal fit (x or y) SD = 3 m

Figure F.7 Angle distribution with normal fit (x or y) SD = 3.5 m
APPENDIX F – ANGLE DISTRIBUTION APPROXIMATION

Figure F.8 Angle distribution with normal fit (x (or y) SD = 4 m)

Figure F.9 Angle distribution with normal fit (x (or y) SD = 4.5 m)

Figure F.10 Angle distribution with normal fit (x (or y) SD = 5 m)

Figure F.11 Angle distribution with normal fit (x (or y) SD = 6 m)

Figure F.12 Angle distribution with normal fit (x (or y) SD = 7 m)

Figure F.13 Angle distribution with normal fit (x (or y) SD = 8 m)
APPENDIX F - ANGLE DISTRIBUTION APPROXIMATION

Angle Distribution with Normal Fit (x or y SD = 9 m)

Figure F.14 Angle distribution with normal fit (x or y SD = 9 m)

Angle Distribution with Normal Fit (x or y SD = 10 m)

Figure F.15 Angle distribution with normal fit (x or y SD = 10 m)

Angle Distribution with Normal Fit (x or y SD = 11 m)

Figure F.16 Angle distribution with normal fit (x or y SD = 11 m)

Angle Distribution with Normal Fit (x or y SD = 12 m)

Figure F.17 Angle distribution with normal fit (x or y SD = 12 m)

Angle Distribution with Normal Fit (x or y SD = 13 m)

Figure F.18 Angle distribution with normal fit (x or y SD = 13 m)

Angle Distribution with Normal Fit (x or y SD = 15 m)

Figure F.19 Angle distribution with normal fit (x or y SD = 15 m)
Figure F.20 Angle distribution with normal fit
(x (or y) SD = 17 m)

Figure F.21 Angle distribution with normal fit
(x (or y) SD = 19 m)

Figure F.22 Angle distribution with normal fit
(x (or y) SD = 21 m)

Figure F.23 Angle distribution with normal fit
(x (or y) SD = 23 m)

Figure F.24 Angle distribution with normal fit
(x (or y) SD = 25 m)

Figure F.25 Angle distribution with normal fit
(x (or y) SD = 27 m)
Figure F.26 Angle distribution with normal fit
(x or y) SD = 29 m

Figure F.27 Angle distribution with normal fit
(x or y) SD = 31 m

Figure F.28 Angle SD against x (or y) SD
Appendix G

SERVICE INTERVAL DISTRIBUTIONS

This appendix presents a sample of the service interval distributions which accompany the results and simulations outlined in Chapter 5. These include the service interval distributions for BNGRAZ, GRPGM, RWP and fixed path whilst varying the following parameters: sensor node speed, size of the WSN, desired coverage period, configuration duty time, and the size of the vicinity around the reference point.

G.1 Varying Sensor Node Speed

These distributions consider a deployment of 15 sensor nodes. The configuration duty time was set to 60 s and the size of the vicinity around the reference plot was set to 50×50 m². The desired coverage period was set to 120 min.

![Figure G.1 BNGRAZ service interval distributions](image)
Service interval distribution (GRPGM, 15 nodes)

Figure G.2 GRPGM service interval distributions

Service interval distribution (RWP, 15 nodes)

Figure G.3 RWP service interval distributions
G.2 Varying the Size of the WSN (number of sensor nodes)

The sensor node speed was set to 0.1 m/s and the configuration duty time was set to 60 s. The size of the vicinity around the reference point was set to 50×50 m² and the desired coverage period was set to 120 min.

**BNGRAZ**

![Figure G.4 Fixed path service interval distributions](image)

**Figure G.5** BNGRAZ 9 nodes service interval distribution

**Figure G.6** BNGRAZ 15 nodes service interval distribution
APPENDIX G – SERVICE INTERVAL DISTRIBUTIONS

Figure G.7 BNGRAZ 20 nodes service interval distribution

Figure G.8 BNGRAZ 25 nodes service interval distribution

Figure G.9 BNGRAZ 30 nodes service interval distribution

GRPGM

Figure G.10 GRPGM 9 nodes service interval distribution

Figure G.11 GRPGM 15 nodes service interval distribution
Figure G.12 GRPGM 20 nodes service interval distribution

Figure G.13 GRPGM 25 nodes service interval distribution

Figure G.14 GRPGM 30 nodes service interval distribution

Figure G.15 RWP 9 nodes service interval distribution

Figure G.16 RWP 15 nodes service interval distribution
APPENDIX G – SERVICE INTERVAL DISTRIBUTIONS

Figure G.17 RWP 20 nodes service interval distribution

Figure G.18 RWP 25 nodes service interval distribution

Figure G.19 RWP 30 nodes service interval distribution

Fixed Path

Figure G.20 fixed path 9 nodes service interval distribution

Figure G.21 fixed path 15 nodes service interval distribution
APPENDIX G – SERVICE INTERVAL DISTRIBUTIONS

Service interval distribution (fixed path, speed 0.1 m/s):

Figure G.22 fixed path 20 nodes service interval distribution

Figure G.23 fixed path 25 nodes service interval distribution

Figure G.24 fixed path 30 nodes service interval distribution
G.3  Varying the Configuration Duty Time

The number of sensor nodes deployed was 9 and the desired coverage period was set to 120 min. The first five distributions consider a node speed of 0.1 m/s and the following five distributions consider a node speed of 0.05 m/s.

Figure G.25 BNGRAZ configuration duty time 10 s (speed 0.1 m/s)

Figure G.26 BNGRAZ configuration duty time 30 s (speed 0.1 m/s)

Figure G.27 BNGRAZ configuration duty time 60 s (speed 0.1 m/s)

Figure G.28 BNGRAZ configuration duty time 90 s (speed 0.1 m/s)
APPENDIX G - CA ALGORITHM MAXIMUM ERROR

Service interval distribution (BNGRAZ, 9 nodes, speed 0.1 m/s)

Figure G.29 BNGRAZ configuration duty time
120 s (speed 0.1 m/s)

Service interval distribution (BNGRAZ, 9 nodes, speed 0.05 m/s)

Figure G.30 BNGRAZ configuration duty time
10 s (speed 0.05 m/s)

Figure G.31 BNGRAZ configuration duty time
30 s (speed 0.05 m/s)

Figure G.32 BNGRAZ configuration duty time
60 s (speed 0.05 m/s)

Figure G.33 BNGRAZ configuration duty time
90 s (speed 0.05 m/s)
The number of sensor nodes deployed was 15 and the desired coverage period was set to 120 min. The node speed was set to 0.1 m/s and the configuration duty time was increased to 5 min.
Figure G.37 BNGRAZ configuration duty time
60 s (15 nodes)

Figure G.38 BNGRAZ configuration duty time
90 s (15 nodes)

Figure G.39 BNGRAZ configuration duty time
120 s (15 nodes)

Figure G.40 BNGRAZ configuration duty time
150 s (15 nodes)

Figure G.41 BNGRAZ configuration duty time
180 s (15 nodes)

Figure G.42 BNGRAZ configuration duty time
210 s (15 nodes)
APPENDIX G – CA ALGORITHM MAXIMUM ERROR

Service interval distribution (BNGRAZ, 15 nodes, speed 0.1 m/s): Service interval distribution (BNGRAZ, 15 nodes, speed 0.1 m/s)

Figure G.43 BNGRAZ configuration duty time

240 s (15 nodes)

Figure G.44 BNGRAZ configuration duty time

270 s (15 nodes)

Figure G.45 BNGRAZ configuration duty time

300 s (15 nodes)
G.4 Varying the Size of the Vicinity Around the Reference Point

These distributions consider a deployment of 9 and 15 nodes with the sensor node speed set to 0.1 m/s. The desired coverage period was set to 120 min and the configuration duty time was set to 60 s.

Figure G.46 GRPGM vicinity size $30 \times 30 \text{ m}^2$ (9 nodes)

Figure G.47 GRPGM vicinity size $35 \times 35 \text{ m}^2$ (9 nodes)

Figure G.48 GRPGM vicinity size $40 \times 40 \text{ m}^2$ (9 nodes)

Figure G.49 GRPGM vicinity size $45 \times 45 \text{ m}^2$ (9 nodes)
APPENDIX G – CA ALGORITHM MAXIMUM ERROR

Figure G.50 GRPGM vicinity size 50×50 m² (9 nodes)

Figure G.51 GRPGM vicinity size 55×55 m² (9 nodes)

Figure G.52 GRPGM vicinity size 60×60 m² (9 nodes)

Figure G.53 GRPGM vicinity size 30×30 m² (15 nodes)

Figure G.54 GRPGM vicinity size 35×35 m² (15 nodes)
APPENDIX G – CA ALGORITHM MAXIMUM ERROR

Service interval distribution (GRPGM, 15 nodes, speed 0.1 m/s)

Figure G.55 GRPGM vicinity size 40×40 m² (15 nodes)

Figure G.56 GRPGM vicinity size 45×45 m² (15 nodes)

Figure G.57 GRPGM vicinity size 50×50 m² (15 nodes)

Figure G.58 GRPGM vicinity size 55×55 m² (15 nodes)

Figure G.59 GRPGM vicinity size 60×60 m² (15 nodes)
G.5 Desired Coverage Period

The following plots evaluate the service interval distributions with respect to the desired coverage period. The number of nodes deployed was 9 and the speed was set to 0.05 m/s. The configuration duty time was set to 60 s and the vicinity around the reference point was set to $30 \times 30$ m$^2$. The Coverage Period (CP) (desired) values considered were 120, 240, 360, 420 and 480 min.

**BNGRAZ**

![Service interval distribution (BNGRAZ, 9 nodes, speed 0.05 m/s) for CP 120 min](image1)

![Service interval distribution (BNGRAZ, 9 nodes, speed 0.05 m/s) for CP 240 min](image2)

![Service interval distribution (BNGRAZ, 9 nodes, speed 0.05 m/s) for CP 360 min](image3)

![Service interval distribution (BNGRAZ, 9 nodes, speed 0.05 m/s) for CP 420 min](image4)

![Service interval distribution (BNGRAZ, 9 nodes, speed 0.05 m/s) for CP 480 min](image5)
APPENDIX G – CA ALGORITHM MAXIMUM ERROR

Figure G.64 BNGRAZ desired coverage period
420 min

GRPGM

Figure G.65 GRPGM desired coverage period
120 min

Figure G.66 GRPGM desired coverage period
240 min

Figure G.67 GRPGM desired coverage period
360 min

Figure G.68 GRPGM desired coverage period
420 min
G.6 MDT (BNGRAZ)

The following plots evaluate the service interval distributions with respect to the MDT applicable to BNGRAZ. The number of nodes deployed was 15 and the speed was set to 0.05 m/s. The configuration duty time was set to 60 s and desired coverage period was set to 120 min.

Figure G.69 GRPGM desired coverage period
480 min

Figure G.70 BNGRAZ MDT = 25%

Figure G.71 BNGRAZ MDT = 25.4%
Service interval distribution (BNGRAZ, speed 0.05 m/s)

Figure G.72 BNGRAZ MDT = 25.8%

Figure G.73 BNGRAZ MDT = 26.2%

Figure G.74 BNGRAZ MDT = 26.6%

Figure G.75 BNGRAZ MDT = 27%

Figure G.76 BNGRAZ MDT = 27.4%