Spatial Reuse Scheduling and Localization for Underwater Acoustic Communication Networks

by

Roee Diamant

B.Sc., The Technion, Israel Institute of Technology, 2002
M.Sc., The Technion, Israel Institute of Technology, 2007

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

The Faculty of Graduate and Postdoctoral Studies

(Electrical and Computer Engineering)

The University Of British Columbia

(Vancouver)

August 2013

© Roee Diamant 2013
Abstract

Ocean exploration, through the development of ocean-observation systems, is a key step towards a fuller understanding of life on Earth. Underwater acoustic communication networks (UWANs) will help to fulfill the needs of these ocean-observation systems, whose applications include gathering of scientific data, early warning systems, ecosystem monitoring and military surveillance. The data derived from UWANs is typically interpreted with reference to the location of a data collecting node, e.g. when reporting an event occurrence, or the location of an object itself is of interest, e.g. when tracking a moving underwater vehicle or diver. In this dissertation, we develop methods for localization and efficient data exchange in UWANs.

In the first part of this work, we focus on underwater localization (UWL). Since global positioning system signals do not propagate through water, UWL is often based on fusing information from acceleration-based sensors and ranging information to anchor nodes with known locations. We consider practical challenges of UWL. The propagation speed varies with depth and location, anchor and unlocalized nodes are not time-synchronized, nodes are moving due to ocean currents, propagation delay measurements for ranging of non-line-of-sight communication links are mistakenly identified as line-of-sight, and unpredictable changes in the ocean current makes it hard to determine motion models for tracking. Taking these features of UWL into account, we propose localization and tracking schemes that exploit the spatially correlated ocean current, nodes’ constant motion, and the periodicity of ocean waves.

In the second part of this thesis, we use location information to develop medium access control scheduling algorithms and channel coding schemes. We focus on adaptive scheduling in which each node transmits based on timely network information. Specifically, our scheduling algorithms utilize the long propagation delay in the channel and the sparsity of the network topology to improve throughput, reliability and robustness to topology changes. To evaluate performance, we have developed a simulator combining existing numerical models of ocean current and of power attenuation in the ocean. We have also verified simulation results in four sea experiments of different channel bathymetry structures, using both industry and self-developed underwater acoustic modems.
Preface

Hereby, I declare that I am the first author of this thesis. The following publications have resulted from the thesis research.

Journal Papers


**Conference Papers**


Preface

Unless stated differently, for all publications, I conducted the survey on related topics, identified the challenges, formalized the suggested solution, performed the analysis, and carried out all of the simulations and sea experiments. I also wrote all paper drafts. My supervisor, Prof. Lutz Lampe, guided my research, validated analysis and methodology, and edited the manuscripts for papers co-authored by him. Parts of the thesis are a result of research collaboration with additional contributors. The co-authors’ contribution is listed below.

1. Journal paper 1 and Conference paper 4: Ghasem Naddafzadeh Shirazi wrote roughly 10% of the simulation code and helped with editing the manuscript.

2. Journal paper 2 and Conference paper 7: Hwee-Pink Tan organized the Singapore sea trial, and helped with editing the manuscript.

3. Journal paper 3 and Conference paper 3: Wenbo Shi wrote the simulation code for the slotted-FAMA benchmark method (roughly 5% of the code); Wee-Seng Soh helped with editing the manuscript.

4. Journal paper 5: For this survey publication I was a co-author. Hwee-Pink Tan wrote the paper drafts, and conducted half of the survey. Marc Waldmeyer and Winston K.G. Seah helped with editing the manuscript. I conducted and wrote half of the survey and identified the research challenges. The material included in the thesis comprises only my contribution from this work.

5. Journal paper 7: Yunye Jin organized the sea trial, and helped with editing the manuscript.

6. Journal paper 9 and Conference papers 1 and 2: Lars Michael Wolff was co-supervised by me during this work. He wrote roughly 40% of the simulation code, and formalized roughly 20% of the suggested solution.
# Table of Contents

Abstract .......................................... ii

Preface ............................................ iii

Table of Contents ................................ vi

List of Tables .................................... x

List of Figures .................................... xi

List of Abbreviations and Symbols ................. xv

Acknowledgments ................................... xix

Dedication ....................................... xx

1 Introduction .................................... 1
  1.1 Underwater Localization and Tracking .......... 3
    1.1.1 Underwater Localization .................. 4
    1.1.2 Tracking the Location of Nodes .......... 9
    1.1.3 Considerations of Measurement Errors .... 11
  1.2 Spatial Reuse Scheduling in UWANs .......... 13
    1.2.1 Scheduling Algorithms for UWANs ....... 13
    1.2.2 Adaptive Transmissions ................. 17
  1.3 Open Problems Addressed in this Thesis ....... 17
    1.3.1 Challenges for Underwater Localization .. 17
    1.3.2 Challenges for Scheduling of UWANs ..... 19
  1.4 Thesis Structure ........................... 19

I Underwater Localization and Tracking .......... 21

2 UWL with Time-Synchronization and Propagation Speed Uncertainties .................................................. 22
  2.1 Intuition .................................... 23
  2.2 System Setup and Assumptions ................ 23
Table of Contents

2.3 The STSL Algorithm ........................................... 25
  2.3.1 Step 1: Time-Synchronization ........................... 26
  2.3.2 Step 2: Localization ................................... 28
  2.3.3 Extensions ............................................... 30
  2.3.4 Scalability ............................................... 31
  2.3.5 Pseudo-Code for STSL ................................. 32
2.4 Cramér-Rao Lower Bound .................................... 32
  2.4.1 General Cramér-Rao Lower Bound ...................... 32
  2.4.2 Application to STSL .................................. 34
2.5 Simulation Results ........................................... 34
2.6 Sea Trial Results ............................................ 39
  2.6.1 Channel and System Characteristics ................... 40
  2.6.2 Results ............................................... 41
2.7 Summary ..................................................... 43

3 Spatially Dependent Underwater Navigation ....................... 45
  3.1 System Model ............................................. 46
  3.2 The SSM and Measurement Model .......................... 47
    3.2.1 State Space Model (SSM) ............................ 48
    3.2.2 Measurement Model ................................. 48
    3.2.3 Cramér Rao Lower Bound (CRLB) .................... 51
  3.3 The DD-UT Scheme ....................................... 52
    3.3.1 Tracking ........................................... 52
    3.3.2 Drift Velocity Estimation ............................ 53
    3.3.3 Confidence Index (CI) ............................. 55
  3.4 Results .................................................. 57
    3.4.1 Simulations ........................................ 58
    3.4.2 Sea Trials .......................................... 63
  3.5 Conclusions .............................................. 66

4 LOS and NLOS Classification for UWL .............................. 67
  4.1 System Setup and Assumptions ............................ 68
    4.1.1 RSS-Based Range Measurements ...................... 69
    4.1.2 PDF for PD Measurements ........................... 70
    4.1.3 Remark on Algorithm Structure ...................... 71
  4.2 Step One: Identifying ONLOS Links ....................... 71
    4.2.1 Classification of non-ONLOS Links ................. 72
    4.2.2 Classification of ONLOS Links ...................... 73
  4.3 Step 2: Classifying LOS and SNLOS Links .................. 73
    4.3.1 Basic Idea .......................................... 73
    4.3.2 Equivalence Constraints ............................ 74
    4.3.3 Formalizing the Log-Likelihood Function ............ 74
# Table of Contents

4.3.4 Estimating the Distribution Parameters $\omega_1$ and $\omega_2$ .......................... 75  
4.3.5 Forming Initial Estimation $\theta^0$ .......................................................... 76  
4.3.6 Discussion .................................................................................................... 76  
4.3.7 Summarizing the Operation of the Classifier ............................................. 76  
4.3.8 Deriving the HCRLB .................................................................................... 78  
4.4 Performance Evaluation .................................................................................... 79  
4.4.1 Simulations .................................................................................................... 79  
4.4.2 Sea Trials ...................................................................................................... 82  
4.5 Conclusions ....................................................................................................... 87  

5 A Machine Learning Approach for Underwater DR ........................................... 89  
5.1 System Model .................................................................................................... 90  
5.2 The DR-A Method ............................................................................................ 93  
  5.2.1 Forming $\hat{A}_h$: Estimation of the Heading Angle ..................................... 93  
  5.2.2 Forming $\hat{A}^{h,p}$: Estimating the Time-varying Pitch Angle ......................... 94  
  5.2.3 Distance Estimation .................................................................................... 100  
  5.2.4 Summarizing the Operation of the DR-A Method ....................................... 100  
  5.2.5 Discussion .................................................................................................. 102  
5.3 Performance Evaluation ..................................................................................... 103  
  5.3.1 Simulation Results ..................................................................................... 103  
  5.3.2 Sea-Trial Results ....................................................................................... 107  
5.4 Conclusion ......................................................................................................... 110  

II Spatial Reuse MAC techniques for UWANs ..................................................... 112  
6 Time and Spatial Reuse Handshake Protocol for UWANs ................................. 113  
  6.1 System Model and Objectives ........................................................................ 113  
  6.1.1 MAC Throughput ....................................................................................... 115  
  6.1.2 Fairness ....................................................................................................... 116  
  6.1.3 Scheduling Delay ...................................................................................... 116  
  6.2 Maximizing Channel Utilization in Handshake Protocols ...................................... 116  
  6.2.1 Types of Primary Conflicts ......................................................................... 118  
  6.2.2 Formalizing Constraints ............................................................................ 118  
  6.2.3 Channel Utilization Maximization Problem ............................................... 122  
  6.3 The TSR Protocol - A Sub-Optimal Approach .................................................. 123  
  6.3.1 Priority of Control Packets ....................................................................... 126  
  6.3.2 Scheduling Control Packets ....................................................................... 127  
  6.4 Results ............................................................................................................ 131  
  6.4.1 Simulation Setting ...................................................................................... 133  
  6.4.2 Simulation Results ..................................................................................... 134  
  6.5 Conclusions .................................................................................................... 138
# Table of Contents

7 Robust Spatial Reuse Scheduling in UWANs .................................. 139
   7.1 Preliminaries ................................................. 140
      7.1.1 System Model ........................................... 140
      7.1.2 Objectives of Resource Allocation ....................... 141
   7.2 Formalizing the BSP ........................................... 143
      7.2.1 Basic Approach ............................................ 143
      7.2.2 Formalizing the T-BSP ................................... 144
      7.2.3 Formalizing the R-BSP ................................... 146
   7.3 Obtaining the Topology Matrix .................................. 154
      7.3.1 Obtaining the Topology Matrix ............................ 154
      7.3.2 Constructing the Conflict Graph ......................... 156
   7.4 Simulation Results ............................................ 157
      7.4.1 Model-Based Topology ..................................... 158
      7.4.2 Sea-Trial-Based Topology ................................. 162
   7.5 Conclusions .................................................... 165

8 Adaptive Error-Correction Coding Scheme for UWANs ...................... 167
   8.1 System Model .................................................. 168
   8.2 The Adaptive Coding Scheme ................................... 168
      8.2.1 Gain of Adaptive Coding .................................. 169
      8.2.2 Optimization of \( N_{\text{actual}}^{\text{t,i}} \) ...................... 172
      8.2.3 Extension to IR-HARQ ..................................... 173
   8.3 Implementation ................................................ 174
   8.4 Performance Results and Discussion ................................ 177
      8.4.1 Simulations ................................................ 177
      8.4.2 Sea Trial Results ........................................ 184
   8.5 Conclusions .................................................... 185

III Summary of Thesis and Further Research ................................. 187

Bibliography .......................................................... 192

Appendices .............................................................

A Alternating Optimization Approach for Solving (4.21) .................... 206

B Expressions for the HCRB ............................................. 208
List of Tables

3.1 Drift velocity estimation results from the Israel and Singapore sea trials. 64

4.1 Israel sea trial: distribution of estimated values of \( \beta_{m \text{last}} \), \( m = 1, 2, \ldots \) 84

4.2 Singapore sea trial: \( \frac{c \bar{x} - E(d_i)}{E(d_i)} \) 86
# List of Figures

1.1 Communications architecture for UWSNs [1] ........................................... 2
1.2 Mapping between the challenges and desirable properties of underwater localization ............................................................ 5
1.3 Illustration of AUV-aided localization ..................................................... 7
1.4 Illustration of various types of communication links: LOS, SNLOS and ONLOS links ................................................................. 12
1.5 Illustration of the vessel’s wave-induced motion ....................................... 13
1.6 Structure of thesis ................................................................................. 20

2.1 $\rho_{\text{err}}$ from (2.31) as a function of $1/\sigma^2$. Sound speed is known and all nodes are time-synchronized. Vertical bars show 95% confidence intervals of the simulation results for STSL .................................................. 35
2.2 $\rho_{\text{err}}$ from (2.31) as a function of $e_{\text{sync}}$. Sound speed is known and $\frac{1}{\sigma^2} = 46$ dB. Vertical bars show 95% confidence intervals of the simulation results for STSL ............................................................................ 36
2.3 $\rho_{\text{err}}$ from (2.31) as a function of $|c - \tilde{c}|$. All nodes are time-synchronized and $\frac{1}{\sigma^2} = 46$ dB. Vertical bars show 95% confidence intervals of the simulation results for STSL .................................................. 37
2.4 $\rho_{\text{err}}$ from (2.31) for the STSL algorithm as a function of number of iteration steps ................................................................. 38
2.5 Empirical PDF of $c_{\text{est}}$ for $c = 1500$ m/sec. ........................................... 39
2.6 $\rho_{\text{err}}$ from (2.31) for time-synchronization and sound speed uncertainties ................................................................................................. 40
2.7 Time-varying location of nodes in the sea trial ........................................... 41
2.8 $\tilde{T}_{\text{diff}}$ for all communication links as a function of time slots ............ 42
2.9 $\bar{\rho}_{\text{err}}$ from (2.32) as a function of $|c - \tilde{c}|$ for $W = 30$ time slots. R-STSL method ................................................................................. 43
2.10 Probability that $\bar{\rho}_{\text{err}} \leq x$ for STSL and MR-STSL ......................... 44

3.1 Simulated scenario: (a) simulated velocity field at depth 20 m, (b) simulated speeds of the anchor nodes and TN ........................................ 58
3.2 (a) Mean of $e_v(k)$ from (3.32) as a function of number of anchor nodes, (b) $P_D$ vs. $P_F$ for different $\sigma_{\text{anc}}^2$ and $\Theta_e$ values ........................................ 59
3.3 Simulated scenario: (a) speed of the TN, (b) tracking error $e_{\text{d}}^n$ from (3.40) ................................................................. 61
List of Figures

3.4 Tracking performance (simulations): (a) empirical C-CDF for MSNR=50dB, (b) \( e_k^d \) as a function of MSNR (thick line shows results for the CRLB). 62
3.5 Node locations in Cartesian coordinates: (a) Singapore sea trial, (b) Israel sea trial. 63
3.6 \( e_k^d \) from (3.40) vs. time: (a) Singapore sea trial, (b) Israel sea trial. 65
4.1 Classification results: (a) \( P_{rd,non-ONLOS} \) and \( P_{rd,ONLOS} \) vs. \( \gamma_{UB} \), (b) Empirical detection probabilities of LOS and SNLOS. 80
4.2 Empirical C-CDF of \( \rho_{err} \) from (4.27). 81
4.3 Estimation error of LOS and SNLOS distribution parameters vs. EM iteration number. 82
4.4 Israel harbor experiment: (a) location (picture taken from Google maps on Sep. 09), (b) results for ONLOS link classification. 83
4.5 Empirical C-CDF of \( \rho_{err} \) from (4.28), \( T_c = 10 T_s \). 85
4.6 Histogram of \( \rho_{err}^{(i)} \) from (4.29). Bin width 0.3 m, \( E(d_i) = 324.1 \) m. 86
4.7 Localization error, \( \rho_e \), as a function of the number of received packets, \( W \). 87
5.1 Flow chart for the DR-A algorithm. 90
5.2 Graph representation for positive and negative constraints. 96
5.3 Example of a modeled wave in the \( x - y \) plane. \( t = 10 \) seconds. 104
5.4 C-CDF results of \( \rho_{angle} \). 105
5.5 Average results of \( \rho_{err} \) as a function of (a) \( \hat{M} \) (for \( \hat{T}_c = T_c \)), (b) \( \hat{T}_c \) (for \( \hat{M} = 10 \)). \( T_c = 6 \) seconds. 106
5.6 Results of \( \rho_{err} \) as a function of \( \frac{1}{\zeta} \). \( \hat{M} = 10, \hat{T}_c = T_c \). 107
5.7 C-CDF results of \( \rho_{err} \). \( \frac{1}{\zeta} = 20 \) dB, \( \hat{M} = 10, \hat{T}_c = T_c \). 107
5.8 Sea trial: nodes location in Cartesian coordinates. (a) x-axis. (b) y-axis. 108
5.9 Sea trial: measured and projected acceleration along the x axis relative to \( g \) for \( \hat{M} = 5 \) and \( \hat{T}_c = 40 \) seconds. Boat 1. 109
5.10 Sea trial: (a) C-CDF results of \( \rho_{angle} \) for the two boats, (b) \( \rho_{angle} \) vs. the accumulated change of the boat’s heading angle within the time slot, \( T_{slot} = 300 \) seconds. 109
5.11 Sea trial: C-CDF results of \( \rho_{err} \) for the two boats. \( T_{slot} = 200 \) sec. 110
6.1 Illustration of channel reservation process (note that the first data packet of node \( j \) is smaller than the rest). 117
6.2 Illustration of different types of primary conflicts. 119
6.3 Illustration of packet exchange mechanism. 125
6.4 Structure of RTS, CTS, NT, and data packets. 131
6.5 Scheduling of RTS and CTS and determining of CS parameters in the TSR protocol. 132
6.6 Average of $\rho_{\text{through}}$ from (6.2) and $\frac{\rho_{\text{delay}}}{T_{i,j}}$ from (6.4) as a function of $R_{\text{msg}}^{\text{msg}} = \frac{T_{i,j}}{T_{i,j}}$ for TSR with and without DSSS. ... 134

6.7 Empirical C-CDF of $\rho_{\text{fair,through}}$ in (6.3) for the TSR, TDMA, Slotted-FAMA, BiC-MAC, and APCAP protocols. $|N| = 6$. ... 135

6.8 Empirical CDF of $\rho_{\text{delay}}$ in (6.4) for the TSR, Slotted-FAMA and BiC-MAC protocols. $|N| = 15$. ... 136

6.9 Empirical C-CDF of $\rho_{\text{through}}$ from (6.2) for the TSR, TDMA, Slotted-FAMA, APCAP, and BiC-MAC protocols. $|N| = 6$. (a) without DSSS signaling, (b) with DSSS signaling. ... 137

6.10 $\rho_{\text{change}}(|N| = 6, 10, 15)$ from (6.37). ... 138

7.1 Example: (a) sample topology for a UWAN. (b) constructing matrix $I_{\text{skel}}$ using Algorithms 4 and 5 for the sample network. (c) masking matrix used in (7.14). ... 149

7.2 Flow chart to obtain the R-BSP scheduling matrix $M$. ... 152

7.3 Probability (7.23) for $k = u - 1$ and empirical probability of $\Delta \geq u - 1$ when generating a Bernoulli random graph with $N$ nodes and edge probability $p$. ... 154

7.4 Convergence parameters $b_1$ and $b_2$ vs. transmission range ... 156

7.5 C-CDF of $\rho_{\text{through}}$ from (7.2) for matched and mismatched OTT and static model-based topology. ... 159

7.6 C-CDF of $\rho_{\text{through}}$ from (7.2), for static model-based topologies and different MPR scenarios. ... 160

7.7 C-CDF of $\rho_{\text{through}}$ from (7.2) for model-based topologies. (a) static (no outdated topology). (b) dynamic (slowly propagating topology) ... 161

7.8 CDF of $\rho_{\text{delay}}$ from (7.3) for static model-based topologies. ... 162

7.9 Structure of Sea Trial: (a) satellite picture of the sea trial location (picture taken from Google maps on September 29, 2009). (b) recorded network topologies. ... 163

7.10 Results of $\rho_{\text{through}}$ from (7.2) for topologies recorded in the sea trial and $T = 10$ time-slots. Each vertical line represents the time a topology change starts affecting results. ... 164

7.11 $\rho_{\text{delay}}$ from (7.3), for static and dynamic sea-trial-based topologies. ... 165

8.1 Gain $g_{\text{coding}}$ from (8.5) as a function of $\Delta_{t,i}$ from (8.2). ... 169

8.2 Gain $g_{\text{energy}}$ from (8.7) as a function of $T_h$ and $\Delta_{t,i}$ from (8.2). Non-erasure channel. ... 171

8.3 Illustration of the procedure of updating channel conditions for multiple packet transmission. ... 174

8.4 Illustration of adaptive coding implementation for single packet transmission. ... 175
List of Figures

8.5  Transmission loss vs. range. Output from the Bellhop simulator [2]. 178
8.6  PER as a function of channel erasure rate. .......................... 179
8.7  Channel utilization as a function of channel erasure probability (for $\Delta_{t,i} \approx 650$ m). .................................................. 180
8.8  Channel utilization as a function of $\Delta_{t,i}$ (for erasure probability of roughly 0.21). .................................................. 181
8.9  Error $\rho_p(m)$ as a function of packet number $m$. (Note that values in the y-axis are reversed). .............................. 182
8.10 Histogram of network latency in terms of number of packets needed till decoding. .................................................. 183
8.11 CDF of the number of symbols transmitted till successful decoding, $N_{\text{final}}$. .................................................. 183
8.12 CDF of network goodput. ........................................ 184
8.13 Satellite picture of the experiment location (picture taken from Google Earth on July 23, 2012.) The three locations of the nodes are marked. 185
8.14 Packet success rate for the three links from the sea trial. ........ 186
# List of Abbreviations and Symbols

## Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARQ</td>
<td>automatic repeat request</td>
</tr>
<tr>
<td>AUV</td>
<td>autonomous underwater vehicle</td>
</tr>
<tr>
<td>BSP</td>
<td>broadcast scheduling problem</td>
</tr>
<tr>
<td>CA</td>
<td>collision avoidance</td>
</tr>
<tr>
<td>C-CDF</td>
<td>complementary cumulative density function</td>
</tr>
<tr>
<td>CI</td>
<td>confidence index</td>
</tr>
<tr>
<td>CL</td>
<td>connectivity list</td>
</tr>
<tr>
<td>CRLB</td>
<td>Cramer-Rao lower bound</td>
</tr>
<tr>
<td>CS</td>
<td>communication session</td>
</tr>
<tr>
<td>CSI</td>
<td>channel state information</td>
</tr>
<tr>
<td>CTS</td>
<td>clear-to-send</td>
</tr>
<tr>
<td>CUMP</td>
<td>channel-utilization-maximization problem</td>
</tr>
<tr>
<td>DR-A</td>
<td>DR navigation using a single accelerometer</td>
</tr>
<tr>
<td>DR</td>
<td>dead-reckoning</td>
</tr>
<tr>
<td>DSSS</td>
<td>direct sequence spread spectrum</td>
</tr>
<tr>
<td>DVL</td>
<td>Doppler-velocity-logger</td>
</tr>
<tr>
<td>EKF</td>
<td>extended Kalman filter</td>
</tr>
<tr>
<td>EM</td>
<td>expectation-maximization</td>
</tr>
<tr>
<td>HCRLB</td>
<td>hybrid Cramer-Rao lower bound</td>
</tr>
<tr>
<td>HST-TDMA</td>
<td>hybrid spatial-reuse time-division multiple access</td>
</tr>
<tr>
<td>INR</td>
<td>interference-to-noise ratio</td>
</tr>
<tr>
<td>INS</td>
<td>inertial system</td>
</tr>
<tr>
<td>IR-HARQ</td>
<td>incremental redundancy hybrid automatic repeat request</td>
</tr>
<tr>
<td>LDPC</td>
<td>low-density parity check</td>
</tr>
<tr>
<td>LOS</td>
<td>line-of-sight</td>
</tr>
<tr>
<td>MAC</td>
<td>medium access control</td>
</tr>
<tr>
<td>MIS</td>
<td>maximal independent set</td>
</tr>
<tr>
<td>MPR</td>
<td>multiple packet reception</td>
</tr>
<tr>
<td>NLOS</td>
<td>non-line-of-sight</td>
</tr>
<tr>
<td>NT</td>
<td>notification</td>
</tr>
<tr>
<td>ONLOS</td>
<td>object-related NLOS</td>
</tr>
<tr>
<td>OTT</td>
<td>orthogonal topology-transparent</td>
</tr>
</tbody>
</table>
List of Abbreviations and Symbols

PCA principal component analysis  
PD propagation delay  
PDF probability density function  
PER packet error rate  
R-BSP robust BSP  
RS Reed-Solomon  
RTS request-to-send  
SD-UT spatially dependent underwater tracking  
SINR signal-to-interference-plus-noise ratio  
SNLOS sea-related NLOS  
SNR signal-to-noise ratio  
SSP sound speed profile  
SSM space state model  
SSV state space vector  
STSL sequential time-synchronization and localization  
T-BSP topology-dependent BSP  
TDMA time-division-multiple-access  
TN tracked node  
ToF time-of-flight  
TSR joint time and spatial reuse  
UAC underwater acoustic channel  
UKF unscented Kalman filter  
UL unlocalized  
UT underwater tracking  
UWAC underwater acoustic communication  
UWAN underwater acoustic communication networks  
UWL underwater localization

Notations and Operators

Bold upper case and lower case letters denote matrices and vectors, respectively. Accents \( \hat{x} \) and \( \tilde{x} \) represents estimation and approximation of \( x \), respectively. The remaining notation and operators used in this thesis are listed below.

- \( L \): Number of anchor nodes directly connected to the UL node
- \( j_i \): 2-D UTM coordinates of the UL node
- \( p_{l} \): 2-D UTM coordinates of the \( l \)th anchor node
- \( c \): Sound speed in water [m/sec]
- \( W \): Duration of the localization window
- \( N \): Number of packets transmitted during the localization window
- \( S_l \): Clock skew of the UL node relative to the \( l \)th anchor node
# List of Abbreviations and Symbols

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O_l$</td>
<td>Clock offset of the UL node relative to the $l$th anchor node [sec]</td>
</tr>
<tr>
<td>$T_{pd}^i$</td>
<td>Propagation delay of the $i$th packet [sec]</td>
</tr>
<tr>
<td>$T_i$</td>
<td>Transmission local time of the $i$th packet [sec]</td>
</tr>
<tr>
<td>$R_i$</td>
<td>Reception local time of the $i$th packet [sec]</td>
</tr>
<tr>
<td>$d_{i,i'}$</td>
<td>Self estimation of distance between locations $j_i$ and $j_{i'}$</td>
</tr>
<tr>
<td>$\psi_{i,i'}$</td>
<td>Self estimation of angle between locations $j_i$ and $j_{i'}$ [rad]</td>
</tr>
<tr>
<td>$\Delta$</td>
<td>Threshold for location quantization [m]</td>
</tr>
<tr>
<td>$\sigma^2$</td>
<td>Variance of ToA measurement noise [sec$^2$]</td>
</tr>
<tr>
<td>$s_{li}$</td>
<td>Ratio between $T_{pd}^i$ and the actual propagation delay</td>
</tr>
<tr>
<td>$a_k$</td>
<td>State space vector</td>
</tr>
<tr>
<td>$y_k$</td>
<td>Measurement vector</td>
</tr>
<tr>
<td>$h_k$</td>
<td>Measurement model vector</td>
</tr>
<tr>
<td>$r_k$</td>
<td>3-D coordinates of the TN</td>
</tr>
<tr>
<td>$r_{anc}^k$</td>
<td>3-D coordinates of the anchor node</td>
</tr>
<tr>
<td>$d_k$</td>
<td>Distance vector $r_k - r_{anc}^k$</td>
</tr>
<tr>
<td>$v_k$</td>
<td>Velocity of the TN</td>
</tr>
<tr>
<td>$\hat{v}_{\text{drift}}$</td>
<td>Drift velocity estimate of the TN</td>
</tr>
<tr>
<td>$v_{anc,\text{drift}}^k$</td>
<td>Estimated drift speed and heading direction of the anchor node</td>
</tr>
<tr>
<td>$\bar{c}_k$</td>
<td>Average sound speed between the anchor and TN</td>
</tr>
<tr>
<td>$\tau_{\text{INS}}$</td>
<td>Time interval between consecutive INS measurements</td>
</tr>
<tr>
<td>$\tau_{\text{range}}$</td>
<td>Time interval between consecutive range measurements</td>
</tr>
<tr>
<td>$m_{\text{INS}}^k$</td>
<td>INS measurement vector</td>
</tr>
<tr>
<td>$m_{\text{ToF}}^k$</td>
<td>Ranging measurement</td>
</tr>
<tr>
<td>$m_{\text{Doppler}}^k$</td>
<td>Doppler shift measurement</td>
</tr>
<tr>
<td>$x_i$</td>
<td>PD measurements</td>
</tr>
<tr>
<td>$x_{\text{LOS}}$, $d_{\text{LOS}}$</td>
<td>Delay and distance in the LOS link, respectively</td>
</tr>
<tr>
<td>$X$</td>
<td>Vector of PD measurements $x_i$ of the same communication link</td>
</tr>
<tr>
<td>$d_{\text{RSS,LB}}$</td>
<td>Lower bound of RSS-based range measurement</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>Propagation loss factor</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Absorption loss factor</td>
</tr>
<tr>
<td>$M$</td>
<td>Assumed number of classes in PD model</td>
</tr>
<tr>
<td>$k_m$</td>
<td>Weight of the $m$th distribution in the mixture distribution model</td>
</tr>
<tr>
<td>$\omega_m$</td>
<td>Vector of parameters of the $m$th distribution</td>
</tr>
<tr>
<td>$\theta$</td>
<td>Vector of parameters of the distribution of $X$</td>
</tr>
<tr>
<td>$T_{\text{LIR}}$</td>
<td>Upper bound on the length of the channel impulse response</td>
</tr>
<tr>
<td>$c$</td>
<td>Propagation speed in the channel</td>
</tr>
<tr>
<td>$d_{\text{ONLOS}}$</td>
<td>Distance of the ONLOS link</td>
</tr>
<tr>
<td>$\Lambda_l$</td>
<td>Group of $x_i$ measurements with the same distribution $\omega_m$</td>
</tr>
<tr>
<td>$\lambda_l$</td>
<td>Classifier for group $\Lambda_l$</td>
</tr>
<tr>
<td>$\varrho_i$</td>
<td>Classifier for $x_i$</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>$d_{i,j}$</td>
<td>distance traveled by the vessel at time period $[t_i, t_j]$</td>
</tr>
<tr>
<td>$N$</td>
<td>number of acceleration measurements</td>
</tr>
<tr>
<td>$T_c$</td>
<td>coherence time of acceleration along the horizontal plane</td>
</tr>
<tr>
<td>$\angle_n$</td>
<td>the vessel orientation with respect to the reference system</td>
</tr>
<tr>
<td>$\alpha_n$</td>
<td>the vessel heading angle with respect to $\angle_n$</td>
</tr>
<tr>
<td>$\rho_n$</td>
<td>the vessel pitch angle in time $t_n$</td>
</tr>
<tr>
<td>$\mathbf{a}_n$</td>
<td>the vessel 3-D acceleration in the horizontal plane coordinate system</td>
</tr>
<tr>
<td>$\hat{\mathbf{a}}_n$</td>
<td>acceleration measurement</td>
</tr>
<tr>
<td>$\hat{\mathbf{A}}$</td>
<td>vector of acceleration measurements</td>
</tr>
<tr>
<td>$\hat{\mathbf{A}}^h$</td>
<td>Matrix $\hat{\mathbf{A}}$ projected to match the vessel heading</td>
</tr>
<tr>
<td>$\hat{\mathbf{A}}^{h,p}$</td>
<td>Matrix $\hat{\mathbf{A}}$ projected into the horizontal plane along the vessel heading</td>
</tr>
<tr>
<td>$\tilde{\mathbf{M}}$</td>
<td>pre-defined number of pitch-states</td>
</tr>
<tr>
<td>$\Delta_l$</td>
<td>group of same pitch-state</td>
</tr>
<tr>
<td>$L$</td>
<td>number of groups $\Delta_l$</td>
</tr>
<tr>
<td>$\delta_l$</td>
<td>classifier for group $\Delta_l$</td>
</tr>
<tr>
<td>$\hat{\Theta}_p$</td>
<td>the distribution parameters of $\hat{\mathbf{A}}$ at the $p$th EM iteration</td>
</tr>
<tr>
<td>$\mu_{m,x}$</td>
<td>mean of the $m$th class for the $x$ axis</td>
</tr>
<tr>
<td>$\sigma_{m,x}$</td>
<td>standard deviation of the $m$th class for the $x$ axis</td>
</tr>
<tr>
<td>$k_m$</td>
<td>prior probability of the $m$th class</td>
</tr>
</tbody>
</table>
Acknowledgments

First and foremost, I would like to express my utmost gratitude to my supervisor, Professor Lutz Lampe, for his assistance, support, and guidance throughout the course of this work. Your fine sense of constructive criticism was invaluable to me throughout the work.

I would also like to thank Dr. Hwee-Pink Tan and Dr. Mandar Chitre for their valuable comments and assistance.

I would like to thank my parents for believing in me and encouraging me throughout my life. I am indebted to my father and my mother for their endless love, care, and support.

Last but not least, I would like to thank my lovely wife, Hadas, for raising our family, my greatest accomplishment, joy, and pride. Without your support none of this could have happened.
To Hadas, Ronya, Gili, and Itamar

“speak to the earth, and it shall teach thee”
Chapter 1

Introduction

The oceans with their diverse biology systems, vast energy resources, and climate and history records of our planet, have always attracted researchers and industries. In the last decade, ocean exploration has considerably increased through the use of ocean-observation systems, autonomous underwater vehicles (AUVs), and fixed or mobile sensor networks. These submerged devices need to report the collective data back to base stations or to share information in the setting of a wireless communication network. Wireless communication underwater is usually established using acoustic transducers since radio frequency communication is only possible for very short distances underwater. Underwater acoustic communication (UWAC) can fulfill the needs of a multitude of underwater applications, including: oceanographic data collection, warning systems for natural disasters (e.g., seismic and tsunami monitoring), ecological applications (e.g., pollution, water quality and biological monitoring), military underwater surveillance, assisted navigation, industrial applications (offshore exploration), to name just a few [3]. For example, in offshore engineering applications, underwater sensors can measure and report parameters such as foundation strength and mooring tensions to monitor the structural health of deepwater mooring systems. In addition, underwater acoustic communication networks (UWANs) comprise communication between surface stations and AUVs. Two common communications architectures for UWANs are shown in Figure 1.1.

Most of the UWAC research in the past has concentrated on the development of models for the underwater acoustic channel (UAC) and the design of secure point to point links. Only recently, networking aspects of UWAC consisting of a fixed but ad-hoc infrastructure (alike base stations in a cellular network) and mobile AUVs (alike mobile phones) have started to attract significant interest in the research community to enable fast, reliable, and cost effective UWAN [4]. One of the major challenges of UWANs is resource assignment which has become the bottleneck for UWAN applications. Scheduling for UWANs is governed by several factors such as low sound speed (approximately 1500 m/sec [5]), half-duplex communication (due to design constraints of the acoustic transducers [6]), large power attenuation, long delay spread, time varying propagation channel, and a very limited signal bandwidth due to absorption loss (which increases with frequency) and noise level (which decreases with frequency) [7], as well as transducer constraints. These factors lead to generally poor availability and reliability performance of UWAN systems and pose engineering challenges that are very different from those experienced in radio frequency wireless networks [8, 9]. In particular, even though UWAC systems have been implemented
Chapter 1. Introduction

for many years, many design and implementation problems remain to be solved towards developing advanced communication network systems enabling applications with high quality of service requirements.

The long propagation delay in the channel, as well as the often sparse nature of UWANs, make location-dependent medium access control (MAC) protocols highly attractive to improve latency and throughput performance of UWANs via spatial-reuse techniques. In this context, underwater localization (UWL) of *unlocalized* (UL) UWAN nodes (infrastructure elements and mobile devices) is a key element towards efficient communication networks enabling, e.g., an ocean-observation system. (We note that the network has to provide its own localization, since global positioning system (GPS) systems do not work underwater.) Moreover, sensed data can only be interpreted meaningfully when referenced to the *location* of the sensor. The following are desirable properties of UWL:

- **Accuracy**
  The location of the sensor for which sensed data is derived should be accurate and unambiguous for meaningful interpretation of data. Localization algorithms usually minimize the distance between the estimated and the true location.

- **Speed**
  Since nodes constantly move, the localization and its tracking procedure should be fast so that it reports the actual location when data is sensed.

- **Wide Coverage**
Chapter 1. Introduction

The localization scheme should ensure that most of the nodes in the network can be localized.

- **Low Communication Costs**

  Accurate localization requires ranging estimation to anchor nodes whose location is known, usually via UWAC. Since the nodes are battery-powered and may be deployed for long durations, communication overhead should be minimized.

In addition to the above quantifiable properties, practical considerations such as ease and cost of deploying reference nodes and other required infrastructure should be taken into account too.

The remainder of this chapter is organized as follows. In Section 1.1, we present a survey of UWL and tracking schemes, and identify important challenges that need to be addressed. In Section 1.2, we describe the state-of-the-art in UWAN MAC protocols with a focus on spatial reuse scheduling that utilize location information. In Section 1.3, we list the open problems in UWL and UWAN MAC design addressed in this thesis. Finally, in Section 1.4 we present the structure of this thesis.

1.1 Underwater Localization and Tracking

Although localization has been widely studied for terrestrial wireless sensor networks, existing techniques cannot be directly applied to UWANs because of the following unique characteristics:

**Deployment of Anchor Nodes**  Assuming depth sensors are used, to localize underwater nodes deployed in the 3D sea environment, reference locations of at least three anchor nodes are required. However, since due to restrictions on energy consumption for long deployment period and sparse network topology, localization coverage is limited, and a node may not always be in the communication range of at least three anchor nodes.

**Node Mobility**  Underwater nodes will inevitably drift due to the water current, winds, shipping activity etc. [10]. While anchor nodes attached to surface buoys can be precisely located through GPS updates, it is difficult to maintain submerged underwater nodes at precise locations. This may affect localization accuracy, as some distance measurements may have become obsolete by the time the node position is estimated. Furthermore, due to the unpredictable nature of the ocean current it is hard to track the location of drifting nodes using a predefined motion model.

**Inter-Node Time Synchronization**  Since GPS signals are severely attenuated underwater, it cannot be used to time-synchronize nodes deployed underwater to
Chapter 1. Introduction

compensate for clock drifts due to both offset and skew. Consequently, the accuracy of time-of-flight (ToF)-based range measurement may be affected. Furthermore, the speed of sound underwater is five orders of magnitude lower than RF propagation over the air. Hence, both clock skew and offset should be compensated for.

Signal Reflection In near-shore or harbor environments, where obstacles may exist between nodes, non-line-of-sight (NLOS) signals reflected from objects (e.g., vessels, harbor wall), or multipath (indirect) signals from the sea surface or bottom can be mistaken for line-of-sight (LOS) signals, and may significantly impact the accuracy of range measurement.

Sound speed variation Unlike the speed of light which is constant, the speed of sound underwater varies with water temperature, pressure and salinity, giving rise to refraction. Without measuring the sound speed, the accuracy of distance measurements based on time-of-arrival approaches may be degraded.

Asymmetric Power Consumption Unlike RF modems, acoustic modems typically consume much more power (order of tens of Watts) in transmit mode compared to receive mode (order of milliWatts). This asymmetry in transmission mode makes it preferable for ordinary nodes to be localized through passive / silent listening.

Low bit rate Compared to RF communications, the bit rates achievable with acoustic communications is significantly lower. As a result, localization packets holding anchors’ location information are long and have high impact on network throughput.

Figure 1.2 maps the above challenges to each desirable localization performance metric.

In the following, we start with a survey of the state-of-the-art in UWL. Next, we present current works on tracking the time-varying location of underwater nodes, and discuss approaches to mitigate localization measurement errors.

1.1.1 Underwater Localization

We review both range-free and range-based UWL techniques. In range-free schemes, UL nodes may infer their proximity to anchor nodes (e.g., in terms of number of hops) so as to achieve coarse localization, e.g., in an area instead of a specific location. Range-based approaches rely on time and/or bearing information to evaluate the distance to anchor nodes, usually using acoustic communications. The UL node then utilizes multilateration/angulation to estimate its own location.
Chapter 1. Introduction

Figure 1.2: Mapping between the challenges and desirable properties of underwater localization.

Range-free Localization

Range-free UWL schemes are designed for cases where range measurements suffer from large errors due to node mobility or strong attenuation. In [11], a UWL scheme based on an assumed attenuation model is proposed. When an anchor node $i$ transmits at power $P_i$, the UL node can receive the transmission as long as it falls within the anchor node’s transmission range, which depends on the transmission power. Hence, by deploying several reference nodes that transmit beacons at multiple power levels, the plane is divided into many small sub-regions defined by intersecting circles. By receiving reports from each UL node of the minimum transmit power at which it received the respective anchor node beacon, a central sink can estimate the location of the UL nodes. However, it is a centralized scheme where mobility is not considered. A range-free UWL scheme for moving anchors is presented in [12]. Here, an AUV traverses a preprogrammed route and performs directional (vertical) beaconing periodically. Assuming that the AUV moves with constant and known speed and knows its position underwater accurately, relying on the AUV periodic broadcasts, the UL node can estimate its position within a circle formed by the intersection of the transmitted beams with the horizontal plane.

A different variant of range-free UWL schemes based on finger-printing have been proposed in [13]. Such schemes involve an offline (or training) stage prior to the online (or prediction) stage. The setup comprises an acoustic signal source capable of transmitting at $M$ different frequencies. During the offline stage, a receiver is placed at a reference location (with known position), and collects $N$ samples of received power at each frequency to constitute an $M \times N$ acoustic-signal map. This is repeated at each reference locations. In the online stage, the receiver measures received power samples from $M$ different frequencies and compare these to the ones samples in the
offline stage using a likelihood function and a “probabilistic-weighted” summation of different reference locations.

Range-based Localization

Range-based localization typically comprises the following steps:

1. **Range measurement**: Each UL node estimates its distance from each anchor node within its communication range using either received signal strength indicator (RSSI), time difference of arrival (TDoA) or time of arrival (ToA). Since the path loss in UAC is usually time varying and multipath effect can result in significant energy fading, the RSSI method is not the primary choice for underwater localization. Hence, most proposed range-based localization schemes use either TDoA or ToA for ranging. The TDoA method involves computing the time difference of arrival between beacons from different anchor nodes transmitted using acoustic signalling, and the ToA method performs ranging based on the relationship among transmission time, speed and distance.

2. **Location estimation**: Each UL node then estimates its position, typically, according to the intersection of various circles centered at each reference node with radii corresponding to the range measurements. In general, to localize a node in $d$-dimensional space, the number of independent range measurements required should be at least $d + 1$.

3. **Tracking**: The location estimate is refined e.g., using measurements from onboard sensors, measurement error models, mobility models, etc.

Range-based UWL method can be classified into 1) single-stage schemes which rely solely on message exchanges with the anchor nodes, and 2) multi-stage schemes in which newly localized nodes can serve as anchor nodes. The key innovation of the first type of UWL schemes lie in how they address localization inaccuracy due to time-synchronization and measurement errors and availability of anchor nodes. While it is usually assumed that clock offset is the main cause of time-synchronization errors, also clock skew cannot be neglected for UWL due to the long propagation delay in the UAC [14]. Furthermore, since anchor nodes are usually submerged we cannot assume these nodes to be time synchronized. Regarding this problem, [14] suggested to estimate both skew and offset based on packet exchange with an already synchronized node. Alternatively, the problem of time-synchronization can be avoided by using TDoA techniques. In [15] “silent positioning” is provided, i.e., UL nodes do not transmit any beacon signal and just listen to the broadcasts of reference nodes to self-position, reducing the communication costs. The scheme relies on TDoA over multiple beacon intervals, and thus does not require time synchronization amongst nodes. However, this kind of “reactive beaconing” makes it susceptible to failure due to transmission losses that are prevalent in harsh UACs. An improvement has been
suggested in [16], where a dynamic mechanism for leader reference node identification and a time-out mechanism to trigger beaconing in the event of transmission loss is presented. However, the scheme does not manage motion of nodes.

![AUV Aided Localization Diagram](image)

Figure 1.3: Illustration of AUV-aided localization.

Allowing node mobility is important for UWL, where the deployment of fixed reference nodes such as surface buoys is time consuming, limits the localization coverage and may be infeasible or undesirable. Several algorithms have been suggested to compensate for node movements, either by regarding these movements as ToA noise [17] or by applying mobility prediction [10]. A few UWL schemes rely on a moving anchor node, where, as illustrated in Figure 1.3, a single node can localize the network. In the AUV-aided scheme proposed in [18], the AUV obtains position updates by rising to the surface to use GPS, and then dives to a predefined depth and periodically performs a two-way message exchange with UL nodes. A 2D localization is achieved once successful two-way message exchanges take place in at least three non-collinear AUV locations. However, UL nodes should remain static. Instead of AUVs, the “Dive’N’Rise” localization scheme [19] uses a weight/bladder mechanism to control the diving/rising of each mobile beacon. These beacons update their positions at the surface, and broadcast them when they dive to a certain depth. However, these approaches consider node movements as an undesired phenomenon and do not utilize the possibility of additional ToA measurements when coupled with self-localization.
Chapter 1. Introduction

While, ideally, three range measurements should be enough to localize a node in a 2D plane through triangularization, ranging errors require the use of multiple range measurements through multilateration. Instead of the commonly-adopted circle-based least-squares multilateration, [20] suggested a hyperbola-based approach. The premise is that when range measurement errors due to imperfect time synchronization, or varying speed in acoustic transmission exist, two hyperbolas always intersect with each other with one cross point, or partial solution, while two circles will likely intersect with either two or zero cross point(s). Several works suggested methods to compensate for location ambiguities such as flips and rotations that arise due to NLOS-related range estimation errors. In [21], a three-phase algorithm is suggested, where first an ambiguity-free sub-tree of nodes is determined. Then, multilateration is performed where the node is first assumed to be located in the center of a rectangular area. Finally, a refinement phase is performed using a Kalman filter to mitigate remaining noise. A robust algorithm for mitigating localization ambiguities is suggested in [22] by rejecting measurements leading to ambiguities, e.g., when there are insufficient anchor nodes or when the location of anchor nodes is almost collinear. In [23], an NLOS factor (i.e., the difference between the arrival times of the NLOS and LOS-based signals) is estimated using a maximum likelihood estimator based on an attenuation model, and NLOS-based measurements are incorporated after a factor correction instead of being rejected. The problem of localization when all measurements are obtained from NLOS links is considered in [24], where the relationship between anchor node distances and the NLOS factor is used to improve localization.

Another significant challenge of UWL is the variability of the sound speed in water as it depends on water temperature, depth and salinity [5]. Considering this problem, [25] suggested to first estimate the sound speed using packet exchange between floating buoys on the seabed and the sea surface. Alternatively, several works suggested sound speed estimation based on measuring the channel characteristics and a sound speed model, e.g., [26], [27]. Differently, [28] suggested to jointly estimate the node location and the propagation speed in the channel by considering the propagation speed as an additional variable. However, they made the assumptions that all nodes in the network are static, at least four anchor nodes are available and that all nodes are time synchronized, which do not hold true in most UWL applications. While changes in the sound speed profile (SSP) with depth have been considered for UWL (e.g., [29]), for shallow water and short to medium range applications, a good approximation for the depth-related sound speed in water is a single parameter, $\bar{c}$, which changes as a function of the mean depth of the transmitter and receiver and can be considered as an average sound speed. In [28], $\bar{c}$ is treated as a system parameter and is separately estimated during the initial localization process. However, due to changes in the depth of the tracked node (TN), the evaluated parameter $\bar{c}$ may shift and its value should be tracked over time.

The key innovations of proposed schemes within the category of multi-stage UWL lie in the trade-off between minimizing error propagation and delay while maximizing
coverage and energy efficiency. In [30], a two-phase algorithm is proposed where a relative coordinate system is built using the first three discovered UL nodes, followed by a selection of new nodes to localize based on their proximity from the new localized nodes. However, the first-stage discovery algorithm requires high volume of message exchange, node mobility is not considered, and only the relative coordinates from the primary seed node is acquired. To combat nodes’ mobility, the authors in [31] proposed a joint localization and synchronization scheme. The 3D network is partitioned into cells, and localization is performed at the cell level. The authors determined the required sensor node density, as well as cell partitioning in order to localize all nodes.

Since multi-stage localization might lead to error-propagation in the network, self-evaluation of the localization accuracy is required to determine if the localized node is eligible to serve as a new anchor node. Several works suggested methods for such self-evaluation, relying either on a node’s expected location in the network [30] or by assigning each node with a confidence index. In [32], a hard-decision confidence index is used based on detecting localization based on outdated range information. Alternatively, in [33], soft decision confidence value is obtained by normalizing the expected position error with the sum of the Euclidean distance to anchor nodes. However, in doing so these methods reuse the information used already for localization, which might cause biased self-evaluation of the localization accuracy.

1.1.2 Tracking the Location of Nodes

After obtaining initial location information via UWL, in case of a moving node, a tracking procedure begins where the time-varying location of the node is recursively estimated. This process is referred to as underwater tracking (UT). Since GPS is not available underwater, UT has similarities to indoor navigation. However, due to the difficulty of modeling motion at sea, UT includes additional challenges. First, the unpredictable water current with fast changes in speed and direction, and the existence of water turbulences, cause irregularities in the motion of the TN [34]. Second, the speed of acoustic signals changes with depth. Last, compared to RADAR applications, where directivity is applied and the emitter is fixed or slowly moving and the propagation speed is known, uncertainties in the sound speed as well as the continuous motion of nodes also make it more challenging to incorporate Doppler shift measurements in the UT scheme.

The scenario adopted by most UT algorithms consists of several anchor nodes and a moving TN, usually an AUV, with an initial location estimate [34]. A variety of on-board sensors are used. The most common are inertial sensors, providing acceleration measurements. Since acceleration is affected by the vessel pitch and roll, an inertial system (INS) also includes a gyrocompass, vibrating angular rate sensors, or Pendulum tilt sensors [34, 35]. During the UT process, occasionally, external information through packet exchange with anchor node is also available. This includes
ranging information as well as Doppler shift measurements. The latter are commonly used in RADAR and Doppler velocity log (DVL) systems, where angle-to-target is measured and propagation speed is assumed known, and it provides information of the relative velocity of a tracked object. In [36], Doppler shift is also used for UWL, assuming a fixed scenario with known sound speed. However, since in underwater acoustic communication omnidirectional transducers are used, the TN and the anchor nodes constantly move, and the sound speed is unknown and may change over time, incorporating Doppler shift measurements into UT is challenging.

External and internal measurements for UT are often incorporated using an state space model (SSM) to update a state space vector (SSV), and a key point is how to reliably determine the SSM. In [37], this problem is handled through a bank of Kalman filters (KFs), each of which uses a different motion model, and the output of each branch is combined according to the accumulated filter errors. To combat model inaccuracies, [38] considered both regular and random motion, and the two components are combined as inner-states of two Markov-like random states. While such an approach offers flexibility in determining the SSM, it is difficult to fit a tracking filter to such a model. Indeed, a common SSM is of fixed speed with a random Gaussian acceleration [34]. To set the model parameters under motion irregularities, the velocity of the TN can directly be measured using a DVL [34]. Unfortunately, DVLs consume non-negligible energy and work well only close to the ocean bottom or surface. Alternatively, the velocity can be estimated by calculating the motor-induced thrust force [39]. However, such velocity measurement is only relative to the water medium, and thus may not be accurate in the presence of high ocean current.

Updating the SSM in time requires the implementation of a tracking filter. In [39], an extended Kalman filter (EKF)-based method is presented to fuse measurements from on-board sensors with occasional range measurements to anchor nodes. An interesting combination of the KF and the EKF is presented in [40], where the former accounts for motion in the surge direction and the latter for angular motion. The scheme includes outlier mitigation using a probabilistic data association filter at the input to the KF. In [41], fusion of sensor information and ranging is formulated as a maximum likelihood optimization problem and the solution is found by a non-linear weighted least-squares optimization. Alternatively, [42] suggested incorporating maximum likelihood data association in an EKF scheme. By using ranging and bearing measurements to several anchor nodes, compass sensor data, and DVL-related speed measurements, good tracking performance is obtained. In [43], it was suggested to include previous locations of the TN in the SSV to combat delays in the range information due to the slow sound speed in the channel, and instead of an EKF, an extended information filter is used to reduce complexity in case the information matrix is sparse. A similar approach is presented in [35], where a post-processing centralized EKF is used to incorporate both anchor and TN sensor information. The used SSV is highly detailed and includes the TN pose, its depth, and its linear and angular velocities.
1.1.3 Considerations of Measurement Errors

Localization and tracking accuracy is highly affected by measurement errors, where the dominant factors are ranging and acceleration measurement errors [34]. The former is affected by the long delay spread in the channel and strong multipath, and the latter is caused by ambiguities of the node orientation, mostly due to ocean waves. In this section, we review current approaches to mitigate such errors.

Propagation Delay Errors

Ranging information, required for both UWL and tracking, is obtained from estimating the ToF of a received signal, using either ToA or TDoA techniques. ToF measurements for range estimation can be obtained (i) from multiple impulse-type signals transmitted in a short period of time, or (ii) from the symbols of a received data packet. The former is a standard in many ultra short baseline systems for ranging underwater (e.g., [44]) and involves inspecting the output of an energy detector [45]. The latter involves inspecting the estimated channel impulse response by performing a matched filter operation at the receiver [46], or by performing a phase-only correlation and using the kurtosis metric to mitigate channel enhanced noise [47]. The ToF is then estimated by setting a detection threshold to identify the arrival of the first path. In [48], a fixed threshold is set based on the channel noise level and a target false alarm probability. In [49], an adaptive threshold is used based on the energy level of the strongest path. A good overview of practical ToF estimators is given in [45].

When estimating the ToF, one has to lock on to a certain arrival path, believed to be the LOS path between sender and receiver. Existing UWL schemes, e.g., [15, 33], implicitly assume that PD measurements correspond to the LOS link between the transmitter and receiver. However, signals can arrive from NLOS communication links in several ways, as illustrated in Figure 1.4. For the node pairs \((u, a_2)\) and \((u, a_3)\), sea surface and bottom reflections links (referred to as sea-related NLOS (SNLOS)) exist, respectively, in addition to an LOS link. For \((u, a_1)\), the signal arrives from the reflection off a rock (referred to as object-related NLOS (ONLOS)). Lastly, between nodes \(u\) and \(a_2\), there is also an ONLOS link due to a ship. While it is expected that power attenuation in the LOS link is smaller than in NLOS links, it is common that the LOS signal is not the strongest. This is because, as shown in multipath models [50, 2] as well as measurements [46], the UAC consists of groups of NLOS links with small path delay, but significant phase differences, often resulting in negative superposition with the LOS link (if delay differences between the LOS and NLOS links are smaller than the system resolution for path separation) as well as positive superposition between NLOS links. If PD measurements of NLOS links are mistakenly treated as corresponding to delay in the LOS link, e.g., in node pairs \((u, a_2)\) and \((u, a_3)\), ranging accuracy will significantly be degraded. Clearly, ranging accuracy affects localization performance. For example, using basic trilateration, the
Figure 1.4: Illustration of various types of communication links: LOS, SNLOS and ONLOS links.

localization error grows quadratically with ranging offset, and a zero-mean Gaussian distributed offset with a standard deviation of only 2 msec (which is quite common in UWL [50, 2, 46]) would cause an average error of 6 m error.

**Acceleration Errors**

Tracking the location of a node usually involves the use of acceleration measurements produced by an INS. By integrating INS measurements the distance traveled by the TN can be estimated. This process is referred to as dead-reckoning (DR). The main challenge in DR navigation is the possible drift and measurement noise of inertial sensors, which may lead to errors in the order of 10% of the traveled distance, depending on the technology employed [51, 34]. For pedestrian applications, using the fact that velocity and orientation can be set to zero when the foot is on ground, it is customary to mount inertial sensors on foot or hip and estimate distance and orientation separately for each step, e.g., [52]. However, while at sea we can identify time instances where the vessel pitch angle is zero, i.e., at the top or bottom of the ocean wave (see Figure 1.5), velocity of the vessel cannot be assumed zero at these points. Instead, reference measurements are used, e.g., DVL, and measurement drifts are controlled through fusion of large number of inertial sensors [53]. For ships, DR navigation involves dynamic positioning, heading autopilots, and thruster-assisted position mooring [54]. However, for small AUV-type TNs with strict energy-constraints, these options are not available.

Another challenge in DR navigation is to determine the orientation of the inertial
sensor with respect to a reference coordinate system, usually using a gyrocompass [55]. Only then the distance traveled by the tracked object can be estimated. However, when the vessel is located close to or on the sea surface such that its motion is affected by the ocean waves, the vessel pitch angle is fast time-varying and orientation measurement may be too noisy to use directly [54]. For this reason, traditionally, DR navigation at sea involves integrating a large number of inertial sensors and applying Bayesian filtering methods, e.g., EKF or particle filters (cf. [51, 56]), to mitigate oscillatory wave-dependent components and reduce measurement drifts [53].

1.2 Spatial Reuse Scheduling in UWANs

Apart from navigation purposes, location information can greatly improve throughput in the setting of a wireless sensor network like a UWAN. For example, location-aware MAC protocols, for which scheduling of nodes’ transmissions are set according to their geographical or relative location in the network (e.g., [57]), can greatly improve throughput and/or latency by utilizing network resources more efficiently while maintaining scheduling limitations. Another example are adaptive coding techniques, where due to the usually low reliability of communication in UWANs, the transmission scheme changes adaptively as a function of the transmitter-receiver distance. In this section, we review current scheduling algorithms and adaptive transmissions techniques for UWANs.

1.2.1 Scheduling Algorithms for UWANs

Scheduling transmissions in UWANs is required for applications with wide range of requirements, e.g., latency, size of information packets, traffic rates, and reliability. Since UWANs are relatively small (usually in the order of tens of nodes), centralized scheduling approaches are considered and both contention-based and contention-free scheduling algorithms are in use [58]. The need to develop reliable communication links and the high cost of retransmissions due to the long transmission delay and large

Figure 1.5: Illustration of the vessel’s wave-induced motion.
energy consumption for transmission [59] make the handshake-based multiple access with collision avoidance (MACA) protocol the method of choice for contention-based scheduling of long unicast transmissions in UWANs [60, 8]. MACA was inspired by the carrier sense multiple access/collision avoidance (CSMA/CA) technique, standardized in IEEE 802.11a, where a communication session (CS) is established by exchanging request-to-send (RTS) and clear-to-send (CTS) packets. Differently, due to the narrow bandwidth available for UWAC, contention-free scheduling of UWANs relies on time-division multiple-access (TDMA) algorithms, where each node is assigned a unique time slot and each time slot includes guard interval to compensate for the (long) propagation delay and (short) time-synchronization offset [61]. Such scheduling is used for broadcast communication or short unicast transmissions in heavy load networks. By scheduling transmissions to avoid packet collisions, both handshake- or TDMA-based scheduling benefit from location information. In fact, in [62] it was shown that by carefully scheduling transmissions, unlike for terrestrial radio-frequency networks where network throughput decreases with the number of nodes, the optimal network throughput achieved for UWANs is $\frac{1}{2}$.

**Handshake-based Scheduling**

Due to the long propagation delay in the UAC the exposure time to packet collisions is long [60], and a modification to the basic handshake protocol is required. Considering this problem, [60] suggested a slotted handshake protocol in which globally established time slots, the size of which is comparable to the propagation delay, are used, and transmissions are restricted to the beginning of these time slots. The authors of [63] suggested improving channel utilization by employing separate time slots for control and data packets, and in [64] MAC throughput was further improved by allowing the receiver to warn the transmitter of expected interferences.

In handshake-based scheduling, channel resources are allocated to the transmitter whose RTS packet was the first to arrive at the receiver, whereas other nodes need to try again to gain channel access after waiting for a certain backoff period. This might lead to an increased delay in packet transmission as the probability of successfully reserving the channel is inversely proportional to the transmission distance [65]. In [60], this problem was resolved by using a fixed backoff-window size instead of an adaptive one as standardized in IEEE 802.11a. However, determining the size of the fixed backoff-window is difficult as it has opposite effects on throughput and transmission delay [66]. For this reason, [66] suggested an algorithm to bring randomness to channel reservation by allocating transmitters with time-varying random ranks and giving priority to the transmitter with the currently highest rank. Unfortunately, they assumed control packets arriving almost simultaneously to the receiver, which does not hold true in UWANs.

In addition to the problem of increased delay, traditional handshake-based MAC protocols require nodes in the proximity of a CS to remain silent, which limits channel
utilization. This effect is even more noticeable in UWANs, where longer silence periods are imposed by the long propagation delay [60]. This also leads to the exposed terminal problem\(^1\), which further decreases channel utilization. In [67], the long propagation delay in the channel is utilized to allow simultaneous transmissions in exposed terminal links. Upon detecting a packet from node \( p \) directed to node \( \tilde{p} \), a node \( j \) schedules its transmissions such that its packets arrive at \( p \) before the response of node \( \tilde{p} \). One way to increase channel utilization is to use timing-advance techniques, often called time reuse [67], such that more nodes can transmit. In UWAC, time reuse is related to the utilization of the long propagation delay in the UAC. In terrestrial radio-frequency networks, performance is inversely proportional to the network size. However, by utilizing the long propagation delay, in UWAC performance is potentially fixed for different network sizes [62]. In [68], time reuse is applied by allowing nodes to initiate another CS while waiting for a CTS response. [69] suggested a distance-aware protocol where channel utilization is improved by allowing both nodes involved in a handshake CS to transmit simultaneously. Alternatively, in [70], time reuse was applied by allowing nodes to opportunistically transmit data packets to a node \( j \), such that the packets arrive at \( j \) upon completion of its own transmissions. However, both in [68] and [69], nodes located within the interference range of a transmitter or receiver should remain silent, and in [70] simultaneous transmission in different (connected) CSs is not allowed, and thus the above mentioned limitation of traditional handshake-based protocols remains.

Since network nodes at different locations experience different channel-access limitations, applying spatial reuse on top of time reuse can further improve channel utilization. Spatial reuse refers to simultaneous CSs in different parts of the network, and it is specifically applicable to UWANs, since low-power half-duplex transceivers, range and frequency dependent absorption loss [5], and acoustic NLOS scenarios lead to sparse network topologies. The spatial-reuse handshake protocol suggested in [71] identifies exposed terminal links when a node overhears RTS packets which are not followed by CTS responses. This node can then transmit simultaneously. [72] suggested using control gaps in each exposed terminal link, allowing nodes to schedule their transmissions via RTS/CTS packet exchange during those gaps. Alternatively, in [73] exposed terminal links are identified by building a conflict map using a trial and error procedure without the need to exchange RTS/CTS control packets.

**TDMA-based Scheduling**

When broadcast communication is required, or when transmitted messages are short compared to the propagation delay in the channel, the overhead of RTS/CTS packets becomes significant due to the collision probability being comparable to that for...  

\(^1\)The exposed terminal problem occurs when a node, upon detecting other transmissions, decides not to transmit, even though these transmissions may not interfere with its own transmission, and vice versa [60].
payload packets. Furthermore, since nodes detecting an RTS or CTS packet should be kept silent, channel utilization decreases. As shown in [74], the number of silenced nodes grows quadratically with the communication range and the CSMA/CA algorithm becomes more and more conservative. Therefore, CSMA/CA is not suitable to meet high network traffic demands for broadcast communication or for short unicast information packets [74]. Moreover, channel reservation cost using MACA is high in UWANs, where the long propagation delay necessitates longer silence periods [60]. In fact, when considering high traffic networks, the conventional TDMA algorithm seems to outperform most of the existing random-access algorithms [74].

In TDMA, besides packet duration, time slots include the expected propagation delay at the maximal transmission range as well as a guard interval to compensate for possible clock drifts between periodic time synchronization. To quantify the latter, consider a clock skew $S$, and guard-interval of $\Delta$ sec. Then, time-synchronization is required every $\frac{k \Delta}{S}$ sec, where the value of $k$ depends on the time it takes to re-synchronize. Since in UWAC propagation delay is long and message rate is low (transmission rates on the order of a few kbit/s are common [8]), the latter is not expected to affect performance much. However, due to the long time-slots, end-to-end transmission delay in TDMA scheduling might be too large in practice even for small number of network nodes [75]. A different approach is to apply spatial-reuse techniques where significant improvement in channel utilization is possible. Spatial reuse allows several nodes to share network resources such as frequency bands (in multicarrier systems, e.g. [76]) or time slots (in TDMA systems, e.g. [77]), increasing channel utilization [78]. Spatial-reuse TDMA is an appealing technique in UWANs where low power transceivers, range and frequency dependent absorption loss [5], and acoustic NLOS scenarios lead to sparse network graphs. The concept of spatial reuse in UWANs was first introduced in [75], where a scheduling algorithm that improves channel utilization by clustering the network was suggested. Assuming short intra-cluster distances, the communication within clusters is based on TDMA, while each cluster is assigned a unique pseudo-random spreading sequence used for signal modulation to reduce interferences between adjacent clusters. In [79] the long propagation delay in the channel was utilized to allow staggered packet transmission. Unfortunately, the scheduling algorithm is based on the propagation delay of specific node-to-node links, which cannot be exploited in node-to-multiple-nodes transmission. Assignment of network resources (i.e., time slots) to maximize channel utilization is known as the broadcast scheduling problem (BSP) [80], [81]. The BSP can be formulated as a graph-coloring problem, and various heuristics to solve (variants of) the BSP have been proposed in e.g., [77, 82, 80]. However, these BSP formulations do not consider transmission of broadcast packets, which requires packet flow control via routing.
1.2.2 Adaptive Transmissions

Several approaches have been suggested to utilize location or propagation delay information by opportunistically transmitting more data in the channel (e.g., [83, 68, 70]). Another method to utilize location information is through transmitter-side adjustment of the transmission scheme or by receiver-initiated request of additional transmissions, i.e., automatic repeat request (ARQ), to ensure successful data delivery. Considering the benefit of channel-dependent adjustments of the transmission scheme, an adaptive modulation scheme was implemented in [84] to optimize transmission rate for time-varying channel conditions. With regards to reliability of transmissions, incremental redundancy hybrid ARQ (IR-HARQ) is particularly efficient as it does not suffer from a coding loss due to repetition of the same parity symbols [85]. Despite the coding efficiency of IR-HARQ, it suffers from high latency due to retransmission requests and retransmissions. Due to the long propagation delay of sound transmission and the low link reliability, this disadvantage is particularly pronounced in UWANs. In part addressing this problem, several adaptive transmission applications tailored to UWANs have been suggested. In [86], an HARQ using rateless codes has been suggested for transferring large files underwater. A rateless coding scheme is also used in [87] and [88] to optimize throughput of UWANs for broadcast communications. Recently, [89] offered to optimize the code rate for the current channel conditions by forming transmitter and receiver collaboration.

1.3 Open Problems Addressed in this Thesis

In the previous section, we have reviewed current approaches for UWL and UWAN scheduling. While for the former, challenges associated with deployment of anchor node, time-synchronization, and mobility have been addressed to some extent in the reviewed schemes, and for the latter collision avoidance scheduling algorithms that combat the long propagation delay in the channel have been suggested, in our view, the following challenges should be, but have not been, fully addressed.

1.3.1 Challenges for Underwater Localization

Sound Speed Variation While most range-based localization techniques assume a known speed of sound underwater, the dependency of the speed of sound with depth, temperature, and salinity, makes it challenging to pre-estimate. In this thesis we will show that a mismatch of roughly 10 m/sec in the assumed sound speed considerably affects localization accuracy. While some works suggested measuring the parameters affecting the sound speed, it is not an easy task for small and relatively simple vehicles, and a model-based approach may induce localization errors. We therefore believe that the sound speed should be estimated and tracked over time.
Chapter 1. Introduction

Inter-node Time Synchronization Localization schemes that rely on silent positioning to minimize communication overhead assume that nodes are time-synchronized. However, unlike surface nodes that can be time-synchronized via GPS updates, the clocks of submerged nodes are subject to skew as well as offset. Although time synchronization algorithms have been proposed for UWANs, to reduce latency and communication overhead they should be incorporated into localization schemes.

Node mobility model Node mobility due to ocean current, which presents one of the greatest challenges for UWL, has only been accounted for up to various degrees. Although some schemes assume a simple mobility model, either the anchor nodes or the UL nodes are always assumed fixed during the localization process. Furthermore, node mobility exhibits different characteristics and irregularities which makes it hard to determine the motion model. By using inertial systems, often available for navigation, and the (possible) spatial correlation of the ocean current, we believe that more information can be available to account for such motion irregularities.

Impact of MAC Delays Another important challenge that has not been fully addressed for UWL is MAC to resolve contention. MAC schemes inevitably introduce delays in transmission, and affect the accuracy of localization schemes that rely on immediate or scheduled responses (e.g., two-way messaging). Due to this delay and the constant motion of nodes in the channel, it is not possible to assume fixed nodes for UWL.

Impact of Channel Structure Since range is measured based on the ToF of the direct path or its received power, it is essential to lock on the location of the direct path of the received signal. Existing algorithms assume that the direct path is the strongest path and thus it’s location is easy to estimate. However, multipath fading can lead to destructive interference and as a result the energy of the direct path is not always the strongest. Moreover, the presence of structures and obstacles in the UAC may result in the loss of the direct signal. Hence, designated mechanisms to classify ToF measurements into LOS and NLOS are needed.

Effect of Ocean Waves Last, current UWL schemes assume capability to estimate or track the orientation angle of the vehicle, and thus project acceleration measurements to the horizontal plain. However, near the ocean surface the vehicle may experience time-varying pitch and roll angles, which may be too irregular to track and too rapid to directly measure. Therefore, a tailored solution to this specific case is required.
1.3.2 Challenges for Scheduling of UWANs

**Location-Dependent Handshake-based Scheduling** The problem of low channel utilization when small-to-moderate information unicast packets are sent in handshake-based scheduling can be overcome by utilizing the long propagation delay in the channel through location information. While some of the reviewed methods exploit exposed terminal links, this is mostly done opportunistically and performance are far from the optimal throughput for UWANs.

**Time-Varying Topology Changes** While spatial-reuse TDMA scheduling algorithms can improve network throughput, the problem of time-varying network topology together with the slow propagation of topology information in the UWAN, can greatly decrease performance of such scheduling algorithms. On the other hand, due to the long duration of the time-slots in TDMA UWAN scheduling, the low throughput of topology-transparent scheduling algorithms (like pure TDMA) may not be sufficient for network requirements. Considering the benefit of high throughput of topology-dependent scheduling and reliability of topology-transparent scheduling algorithms, a scheme combining both approaches is the natural next step.

**Utilizing Location Information for Adaptive Scheduling** The long propagation delay in the channel together with transmitter-receiver distance information offer great opportunities for improving performance by means of adaptive transmissions. Intuitively, by setting the transmission parameters according to the expected delay in the channel, throughput can be optimized. While some works consider adaptive coding for UWANs, a pure distance-dependent adaptive channel coding scheme has not been suggested.

1.4 Thesis Structure

In this thesis, we present UWL and spatial-reuse scheduling algorithms for UWANs that address the open problems identified in Section 1.3. As illustrated in Figure 1.6, the former serves as a building tool for the latter. We divide the thesis into two parts: I) UWL, and II) spatial-reuse scheduling for UWANs. In Chapter 2, we present a scheme for joint time-synchronization and localization, which considers sound speed uncertainties and utilizes motion in the channel to allow localization even when only a single anchor node is available. Using this location information, in Chapter 3 we propose a location tracking scheme that combats the effect of motion irregularities, tracks the sound speed, and incorporates Doppler shift measurements. Next, we consider the problem of ranging and INS measurement errors, which affect localization and tracking accuracy. The former problem is considered in Chapter 4, where we present a classification approach of ToF information into classes of NLOS and LOS. The latter challenge is the focus of Chapter 5, where, for the case of a TN whose
motion is affected by ocean waves, we suggest a machine-learning approach to project acceleration measurements into the horizontal plane and perform DR without using orientation measurements. The connections between the above chapters is illustrated in Figure 1.6.

In Part II, we present spatial-reuse MAC techniques that rely on either location or topology information available through the UWL capability developed in Part I. In Chapter 6, we show how location information can be used in a handshake-based scheduling algorithm to utilize all available network resources even in fully connected networks. For broadcast UWAC, in Chapter 7 we present an optimal spatial-reuse TDMA scheduling scheme to trade off robustness to topology changes and network throughput. Based on such TDMA scheme, in Chapter 8 we propose an adaptive channel coding technique that utilizes location information to greatly increase network throughput. Finally, in Part III we summarize our contributions and suggest topics for further research.
Part I

Underwater Localization and Tracking
Chapter 2

UWL with Time-Synchronization and Propagation Speed Uncertainties

Considering the challenges identified in Section 1.3.1, in this chapter we propose a new algorithm for UWL. In particular, our algorithm takes into account anchor and UL node mobility as well as propagation speed uncertainties, can function with only one anchor node, and includes time-synchronization of nodes. These abilities are important to enable localization under varying conditions, such as static or mobile nodes, in shallow or deep water, and when nodes are submerged for short or long periods of time. Our setting includes several anchor nodes at known locations and one or more UL node, whose location is estimated. We assume that UL nodes are equipped with means to self-evaluate their speed and direction such as accelerometer and compass. Since such inertial systems are relatively light weight and inexpensive, there is a large variety of applications that satisfy this assumption for UL nodes. These include AUV, remotely operated underwater vehicles (ROV), manned vehicles, and divers [34]. Operating in tandem with self-localization systems, our algorithm makes use of the permanent movements of underwater nodes. It also performs a self-evaluation of localization accuracy, by estimating the propagation speed and checking its validity, relying on known model boundaries for it. According to the structure of the proposed algorithm we refer to it as sequential time-synchronization and localization (STSL) algorithm. We demonstrate the advantages of the STSL algorithm by simulation comparisons with two benchmark localization methods, which reveal significant localization errors for the latter if nodes are not time-synchronized or the propagation speed is not accurately estimated. Furthermore, we formalize the Cramér-Rao lower bound for UWL and show that it is well approached by the STSL algorithm. Considering the problem of accurately modeling the UAC in a simulation environment, we also conducted a sea trial in August 2010 in Haifa, Israel, and present results that confirm the performance of the proposed algorithm under real conditions.

The remainder of this chapter is organized as follows. In Section 2.1, we briefly summarize the general structure of our algorithm and the intuition behind it. In Section 2.2, we introduce the system model, followed by a detailed description and discussion of our STSL algorithm in Section 2.3. Cramér-Rao lower bounds perti-
Chapter 2. UWL with Time-Synchronization and Propagation Speed Uncertainties

...ent to our problem are derived in Section 2.4. Simulation and sea trial results are presented and discussed in Sections 2.5 and 2.6, respectively. Finally, conclusions are drawn in Section 2.7.

2.1 Intuition

The intuition behind our approach is the use of relative speed and direction information available at the mobile UL node to compensate for node mobility. In doing so, three or more range measurements obtained at different times and locations can be combined for 2-D localization. This approach allows us to readily include the localization procedure as part of the operation of a communication network. More specifically, instead of using designated localization packet exchange (which is necessary if node mobility is not compensated), we rely on periodic packet exchange between the network nodes. This characteristic renders our approach more flexible and easy to integrate into a UWAC system and also reduces communication overhead.

The STSL algorithm uses a two-step approach, in which first nodes are time-synchronized and then location is estimated. In both steps, the measured time of flight of packets exchanged between anchor and UL nodes and self-localization data obtained at UL nodes are linked to the unknown location, synchronization (clock skew and offset), and propagation speed parameters through linearized matrix equations. Our algorithm is modular in the sense that both time-synchronization and localization steps can be readily replaced with alternative solutions (as we do in this chapter to benchmark the STSL performance).

Before describing the STSL algorithm in detail, we next present the system model and assumptions used in this work.

2.2 System Setup and Assumptions

Our setting includes one or more UL nodes directly connected to \( L \geq 1 \) anchor nodes, which have means to accurately measure their time-varying 2-D location and transmit it to the UL node. Both UL and anchor nodes operate in a time-slotted UWAC network, where nodes transmit at the beginning of globally established time slots as in, for example TDMA, slotted handshake [60] or slotted Aloha [90] transmission\(^2\). Since usually UL nodes perform localization independently of each other, we consider localization of one UL node in the following.

We are interested in estimating the 2-D location of the UL node in terms of the universal transverse mercator (UTM) coordinates \( j_N = [j^x_N, j^y_N]^T \) (the subscript \( N \) becomes clear below) after a pre-defined localization window of duration \( W \) time-slots, which without loss of generality starts at the UL node local time \( t_{UL} = 0 \). We

\(^2\)We note that a relaxation of this assumption is possible by having anchor nodes time-stamp their packets, thereby informing the UL node of the transmission time.
assume that nodes are not time-synchronized. Suppose that the local time at the UL node $t_{UL}$ corresponds to the time $t_l$ according to the local clock of anchor node $l$, and let $S_l$ and $O_l$ denote the clock skew and offset of the UL node relative to node $l$, $l = 1, \ldots, L$, which are constant within the localization window. Then,

$$t_{UL} = t_l \cdot S_l + O_l .$$  

We assume that the time-synchronization error is small relative to the globally established time-slot duration, such that a node can match a received packet with the time slot it was transmitted in. Thus, local transmission times of received packets are known and time stamps of packets are not required. We note that long time slots are common in UWAC [8] due to the low propagation speed underwater, which is modeled between 1420 m/sec and 1560 m/sec [5].

For localization we rely on a two-way packet exchange between the anchor nodes and the UL node. Let $N$ be the total number of packets exchanged between the UL node and the $L$ anchor nodes during the localization window $W$. For convenience, we define the sets $N^a$ and $N^b$ for enumerating the packets to and from the UL node, respectively, such that $N^a \cup N^b = \mathcal{N} = \{1, \ldots, N\}$. Denote $l_i$ the index of the anchor node that transmits ($i \in N^a$) or receives ($i \in N^b$) the $i$th packet, and $T_i$ and $R_i$ the transmission and reception local times of this packet. Also denote $T_i^{pd}$ the propagation delay for packet $i$ according to local clock of anchor node $l_i$. (Note that the propagation delay according to the clock of the UL node is $T_i^{pd} \cdot S_i$). Consider packet $i \in N^a$ transmitted at local time $T_i$ and detected by the UL node at anchor node $l_i$ local time $T_i + T_i^{pd} + \gamma_i$, where $\gamma_i$ is a propagation-delay-measurement-noise sample. Also, consider packet $i \in N^b$ received by anchor node $l_i$ at local time $R_i + \gamma_i$ and transmitted at anchor node $l_i$ local time $R_i + \gamma_i - T_i^{pd}$. Then, following the relation in (2.1), the above time variables are related by

$$R_i = S_i(T_i + T_i^{pd} + \gamma_i) + O_i , \quad i \in N^a$$

$$T_i = S_i(R_i + \gamma_i - T_i^{pd}) + O_i , \quad i \in N^b .$$

Since for $i \in N^a$ the UL node measures $R_i$ and is aware of $T_i$ via packet association (recall we assume transmissions in globally established time slots), and since for $i \in N^b$ the UL node records $T_i$ and is informed of $R_i$ through communication with anchor node $l_i$, the UL node is able to construct equations (2.2a) and (2.2b). For mathematical tractability and formulating a practical algorithm, we assume that the noise $\gamma_i$ is a zero-mean i.i.d. Gaussian random variable with variance $\sigma^2$. Since more complicated models, such as mixture models with one component having non-zero mean, would likely be a more faithful noise representations, we study the effect of model mismatch in Section 2.5.

Considering a dynamic scenario in which all nodes permanently move either by own means or by ocean current, we assume that the UL node uses an inertial system to self-estimate its speed and direction. During the localization window, the inertial
system provides \( N \) position estimations \( \tilde{\mathbf{j}}_i = [\tilde{j}_i^x, \tilde{j}_i^y]^T \) for the true locations \( \mathbf{j}_i \) of the UL node at the time of transmission (\( i \in \mathcal{N}^a \)) or reception (\( i \in \mathcal{N}^b \)) of the \( i \)th packet. These locations are translated into a series of motion vectors, \( \mathbf{\omega}_{i,i'} = [\tilde{d}_{i,i'}, \tilde{\psi}_{i,i'}]^T \), where \( \tilde{d}_{i,i'} \) and \( \tilde{\psi}_{i,i'} \) are the distance and angle between two self-estimated locations \( \tilde{j}_i \) and \( \tilde{j}_{i'} \), respectively. More specifically, assuming depth differences to be small (extensions are straightforward but not included here for brevity), the elements of a single motion vector \( \mathbf{\omega}_{i,i'} \) are

\[
\tilde{d}_{i,i'} = \|\tilde{j}_i - \tilde{j}_{i'}\|_2, \quad \tan(\tilde{\psi}_{i,i'}) = \frac{\tilde{j}_i^y - \tilde{j}_{i'}^y}{\tilde{j}_i^x - \tilde{j}_{i'}^x}.
\] (2.3)

While we do not directly use the self-estimated locations \( \tilde{j}_i \), whose errors accumulate with time, we rely on the accuracy of the motion vectors for all packet pairs \( i, i' \) transmitted or received by the UL node during the localization window. That is, we assume that for \( i, i' \in \mathcal{N} \) the estimated distance \( \tilde{d}_{i,i'} \) equals the true distance \( d_{i,i'} \) and that \( \tilde{\psi}_{i,i'} \) equals the true angle \( \psi_{i,i'} \). We note that this assumption sets limits on the value of \( W \), which is determined by the specifications of the inertial system in use.

We are now ready to present the STSL algorithm for UWL.

### 2.3 The STSL Algorithm

We are interested in accurately estimating the position \( \mathbf{j}_N \) of the UL node at the end of the localization window. In this section we first formalize the optimization problem for estimating \( \mathbf{j}_N \), using ToA measurements obtained from received packets and taking into account inertial system information. Then, we derive a sub-optimal solution, namely the STSL algorithm, in which first nodes are time-synchronized and then localization is performed.

According to the system description in the previous section, the location \( \mathbf{p}_i \) of anchor node \( l_i \) when transmitting (\( i \in \mathcal{N}^a \)) or receiving (\( i \in \mathcal{N}^b \)) the \( i \)th packet, the transmission and reception local times \( T_i \) and \( R_i \), respectively, and the motion vector \( \mathbf{\omega}_{i,i'} \) between locations \( \mathbf{j}_i \) and \( \mathbf{j}_{i'} \) are available at the UL node. Hence, denoting the propagation speed \( c \), the UWL problem can be formulated as

\[
\hat{j}_N = \arg \min_{\mathbf{j}_N} \sum_{i \in \mathcal{N}} \left( s_i T_i^{\text{pd}} - \frac{1}{c} \|\mathbf{j}_i - \mathbf{p}_i\|_2 \right)^2
\] (2.4a)

s.t. 1420m/sec ≤ \( c \) ≤ 1560m/sec

\[
R_i = S_l(T_i + T_i^{\text{pd}}) + O_{li}, \quad i \in \mathcal{N}^a
\] (2.4b)

\[
T_i = S_l(R_i - T_i^{\text{pd}}) + O_{li}, \quad i \in \mathcal{N}^b
\] (2.4c)

\[
d_{i,i'} = \|\mathbf{j}_i - \mathbf{j}_{i'}\|_2, \quad i, i' \in \mathcal{N}
\] (2.4d)

\[
\tan(\psi_{i,i'}) = \frac{\tilde{j}_i^y - \tilde{j}_{i'}^y}{\tilde{j}_i^x - \tilde{j}_{i'}^x}, \quad i, i' \in \mathcal{N}
\] (2.4e)
where the factor $\varsigma_i$ in (2.4a) accounts for the difference between $T_{pd}^i$ (according to anchor node $l_i$ local clock) and the actual propagation delay of packet $i$. Note that problem (2.4) is different from conventional localization problems due to the time-synchronization constraints (2.4c) and (2.4d), and due to the unknown sound speed $c$. Since (2.4) is a non-convex problem, we device our STSL algorithm as a pragmatic solution to the localization problem at hand, and we will compare its performance to the Cramér-Rao lower bound (CRLB) associated with estimating the desired location $j_N$ as well as the unknown parameters $S_l$ and $O_l$, $l = 1, \ldots, L$, and $c$.

In the following we describe the details of our STSL algorithm starting from the time-synchronization step and followed by the localization step.

### 2.3.1 Step 1: Time-Synchronization

The objective of the time-synchronization step is to provide estimates of the propagation delays $T_{pd}^i$, $\forall i \in N$. This is accomplished by two-way packet exchange, obtaining equations of type (2.4c) and (2.4d). However, due to the permanent motion of nodes in the channel, propagation delays $T_{pd}^i$, $i \in N^a$, and $T_{pd}^j$, $j \in N^b$, might not be equal, and thus (2.4c) and (2.4d) cannot be readily compared. Common time-synchronization approaches for UWAC deal with this problem by letting the receiving node respond immediately to limit any possible movements (e.g., [14]). However, such a requirement limits the scheduling protocol. We choose a different approach and apply a quantization mechanism to allow for differences in the propagation delay of separate packets and to enable time-synchronization per anchor node making use of the ongoing network communications.

#### Quantized Representation of Node Movements

In the quantization step, the locations of the UL node and the anchor nodes are quantized so that multiple ToA measurements from two-way communication are associated with the same pair of quantized locations. More specifically, consider the two packets $n, m$, $n \in N^a$, $m \in N^b$. If the two sets of UL node locations $j_n$, $j_m$ and anchor node locations $p_n$, $p_m$ with $l_n = l_m = l$ are associated with the same quantized location $k_\rho$ and $u_{l,\nu}$ of the UL node and anchor node $l$, respectively, we assume that $T_{pd}^n = T_{pd}^m$ and (2.2a) and (2.2b) can be combined as we show further below. The variables $\nu$ and $\rho$ are used to enumerate quantized locations.

To quantize the locations of anchor nodes, we introduce subsets $U_{l,\nu} \in N$ including all packets associated with the same anchor node $l$ such that for each pair of packets $n, m \in U_{l,\nu}$, $\|p_n - p_m\|_2 < \Delta$, where $\Delta$ is a fixed threshold. Next, we associate location $p_i$, $i \in U_{l,\nu}$, with the quantized location $u_{l,\nu}$. Similarly, to quantize locations of the UL node we form subsets of packets $K_\rho \in N$ such that for each pair of packets
Chapter 2. UWL with Time-Synchronization and Propagation Speed Uncertainties

\( n, m \in K, d_{n,m} < \Delta \), and associate location \( \tilde{j}_i, i \in K \), with the quantized location \( k_{i} \). We note that a single packet can be associated with multiple subsets \( U_{l,\nu} \) and \( K \).

There is a tradeoff for choosing \( \Delta \). If \( \Delta \) is too large, the assumption of identical propagation delay is notably flawed and thus the accuracy of the time-synchronization process is low. If \( \Delta \) is too small, there might not be enough two-way ToA measurements associated with each pair of quantized locations \( u_{l,\nu} \) and \( k_{i} \), and again accuracy of the time-synchronization process is degraded, as we further discuss below.

Estimating the Clock Skews and Offsets

We now use the quantized locations to estimate clock skews \( S_l \) and offsets \( O_l \), \( l = 1, \ldots, L \). Let us define subsets \( N_t^a \subseteq N_t^a \) and \( N_t^b \subseteq N_t^b \), with cardinality \( N_t^a \) and \( N_t^b \), respectively, including all packets associated with anchor node \( l \). Consider the pair of packets \( n, m, n \in N_t^a_l, m \in N_t^b_l \), for which locations \( p_n \) and \( p_m \) are mapped onto the same quantized location \( u_{l,\nu} \), and locations \( \tilde{j}_n \) and \( \tilde{j}_m \) are mapped onto the same quantized location \( k_{i} \). We assume that for each anchor node \( l \), this mapping results into \( M_l \) pairs of equations (2.2a) and (2.2b). Clearly, \( M_l \) increases with the quantization threshold, \( \Delta \). As stated above, we neglect the differences between the propagation delays \( T_{pd}^n \) and \( T_{pd}^m \) in (2.2a) and (2.2b) and thus obtain \( M_l \) equations of the form

\[
\frac{R_n + T_m}{S_l} - \frac{2O_l}{S_l} = T_n + R_m + \gamma_n + \gamma_m, \quad n \in N_t^a, \ m \in N_t^b.
\]

Note that equations of type (2.5) are introduced separately for each anchor node \( l \). This is because the estimated clock skew and offset are different for each \( l \).

Introducing the variable vector \( \theta_l = [\theta_l(1), \theta_l(2)]^T = [\frac{1}{S_l}, \frac{O_l}{S_l}]^T \), we express (2.5) as the linear matrix equation

\[
B_l\theta_l = b_l + \epsilon_l
\]

for each anchor node \( l \), where \( B_l \) is an \([M_l \times 2]\) matrix with rows \([R_n + T_m, -2]\), and \( b_l \) and \( \epsilon_l \) are column vectors of appropriate length with elements \( T_n + R_m \) and \( \gamma_n + \gamma_m \), respectively, with \( n \in N_t^a, \ m \in N_t^b \). Next, we apply the LS estimator

\[
\hat{\theta}_l = (B_l^T B_l)^{-1} B_l^T b_l
\]

for each anchor node \( l \). By (2.7), the covariance matrix of \( \hat{\theta}_l \) is [91]

\[
Q_{\theta} = \sigma^2 (B_l^T B_l)^{-1},
\]

whose main diagonal elements are proportional to \( \frac{1}{M_l} \) and \( \frac{1}{M_l^2} \), respectively. Hence, for large \( M_l \) the estimates \( \hat{\theta}_l(1) \) and \( \hat{\theta}_l(2) \) are expected to have much smaller variance than \( \sigma^2 \).
Chapter 2. UWL with Time-Synchronization and Propagation Speed Uncertainties

Estimating Propagation Delays

After estimating $\theta_l(1)$ and $\theta_l(2)$, the quantized locations are no longer in use and we return to our initial objective, which is to estimate the propagation delay. Thus, localization accuracy of the STSL algorithm is not limited to $\Delta$. Considering (2.2), for packets $n \in N_i^a$ and $m \in N_i^b$, the UL node estimates the propagation delay as

$$\hat{T}_{pd}^n = R_n \hat{\theta}(1) - \hat{\theta}(2) - T_n, \ n \in N_i^a$$

$$\hat{T}_{pd}^m = \hat{\theta}(2) - T_m \hat{\theta}(1) + R_m, \ m \in N_i^b. \quad (2.9)$$

We observe from (2.9) that the propagation delay estimation error is a function of both ToA measurement error, $\gamma_i$, and clock skew and offset estimation errors. However, since during the localization window $R_n$ and $T_n$ are bounded by $W$, the variances of $R_n \hat{\theta}(1) - \hat{\theta}(2), \ n \in N_i^a$, and $\hat{\theta}(2) - T_m \hat{\theta}(1), \ m \in N_i^b$, are expected to be much smaller than $\sigma^2$. Thus, we use the approximation $\hat{T}_{pd}^i = T_{pd}^i + \gamma_i$ in the following.

2.3.2 Step 2: Localization

We now introduce the localization step of the STSL algorithm. This step is performed immediately after time-synchronization, using propagation delay estimations (2.9). The objective of the localization step of the STSL algorithm is to estimate the UL node UTM coordinates $j_N^x$ and $j_N^y$ at the end of the localization window $W$. For this purpose, we adopt the common approach to linearize the estimation problem [92], and first estimate the transformed variable vector $\zeta_N = [(j_N^x)^2 + (j_N^y)^2, j_N^x, j_N^y]^T$.

Define

$$\alpha_{i,i'} = \frac{\tilde{d}_{i,i'}}{\sqrt{1 + \tan(\tilde{\psi}_{i,i'})^2}}, \quad \beta_{i,i'} = \alpha_{i,i'} \tan(\tilde{\psi}_{i,i'}), \quad (2.10)$$

and assume $\tilde{d}_{i,i'}$ and $\tilde{\psi}_{i,i'}$ in (2.3) to be equal to $d_{i,i'}$ and $\psi_{i,i'}$ from (2.4), respectively (recall that we rely on the accuracy of the motion vectors during the localization window). Thus,

$$j_N^x = j_i^x - \alpha_{i,i'}, \quad j_N^y = j_i^y - \beta_{i,i'}, \quad i, i' \in N. \quad (2.11)$$

Furthermore, we have from (2.4) that

$$\hat{T}_{pd}^i = \frac{1}{\varsigma_i c} \|j_i - p_i\|_2 + \gamma_i, \quad i \in N. \quad (2.12)$$

Since $c$ is unknown, and assuming small differences between $\varsigma_i$ such that $\frac{\varsigma_i}{\varsigma_{i'}} \cong 1, \ l, l' = 1, \ldots, L$ (a relaxation of this assumption is given further below), we reduce the set of $N$ equations (2.12) to $N - 1$ equations, which together with (2.11) can be written as

$$\mu_{N,i} \cdot \zeta_N = a_{N,i} + \epsilon_{N,i}, \quad i = \ldots, N - 1 \quad (2.13)$$
Chapter 2. UWL with Time-Synchronization and Propagation Speed Uncertainties

with vector $\mathbf{\mu}_{N,i} = [\mu_{N,i}(1), \mu_{N,i}(2), \mu_{N,i}(3)]$, where

$$
a_{N,i} = \left( \hat{T}_{tpd}^i \right)^2 \left( (p_{N_i}^x)^2 + (p_{N_i}^y)^2 \right) - \left( \hat{T}_{tpd}^N \right)^2 \left( (p_{i}^x + \alpha_{N,i})^2 + (p_{i}^y + \beta_{N,i})^2 \right),$$

$$
\mu_{N,i}(1) = \left( \hat{T}_{tpd}^N \right)^2 - \left( \hat{T}_{tpd}^i \right)^2,
$$

$$
\mu_{N,i}(2) = 2 \left( \hat{T}_{tpd}^i \right)^2 p_{N_i}^x - 2 \left( \hat{T}_{tpd}^N \right)^2 (p_{i}^x + \alpha_{N,i}),
$$

$$
\mu_{N,i}(3) = 2 \left( \hat{T}_{tpd}^i \right)^2 p_{N_i}^y - 2 \left( \hat{T}_{tpd}^N \right)^2 (p_{i}^y + \beta_{N,i}),
$$

and $\epsilon_{N,i}$ is the noise component originating from the noisy estimations (2.9). For the localization window $W$, we construct an $[(N-1) \times 3]$ matrix $A$ with rows $\mathbf{\mu}_{N,i}$ and vectors $\mathbf{a}$ and $\mathbf{\epsilon}$ with elements $a_{N,i}$ and $\epsilon_{N,i}$, respectively. Then, the $(N-1)$ equations (2.13) are arranged in

$$
A\mathbf{\zeta}_N = \mathbf{a} + \mathbf{\epsilon}.
$$

The elements of the error vector $\mathbf{\epsilon}$ depend on the elements of $\mathbf{\zeta}_N$. Thus, direct estimation of $\mathbf{\zeta}_N$ from (2.15) will result in low accuracy. Hence, we follow [92] and offer a two-step heuristic approach in which first we get a coarse estimate of $\mathbf{\zeta}_N$, and then we perform a refinement step. The coarse estimate is given by

$$
\hat{\mathbf{\zeta}}_{N}^{LS} = (A^T A)^{-1} A \mathbf{a}.
$$

We note that $\epsilon_{N,i}$ from (2.13) can be formalized as $\gamma_i f_{N,i}$, where $f_{N,i}$ is a function of the elements of $\mathbf{\zeta}_N$, not given here for brevity. Thus, $\epsilon_{N,i}$ are i.i.d random variables and the covariance matrix $\sigma^2 Q_N$ of $\mathbf{\epsilon}$ is a diagonal matrix whose $i$th diagonal element equals $\sigma^2 f_{N,i}^2$. Using $\hat{\mathbf{\zeta}}_{N}^{LS}$ from (2.16) to estimate the elements of $f_{N,i}$, $i = 1, \ldots, N-1$, we estimate $Q_N$ as $\hat{Q}_N$. The refined estimate of $\mathbf{\zeta}_N$ follows as

$$
\hat{\mathbf{\zeta}}_{N}^{WLS} = \left( A^T \hat{Q}_N^{-1} A \right)^{-1} A \hat{Q}_N^{-1} \mathbf{a},
$$

with the error covariance matrix [91]

$$
\hat{Q}_N' = \left( A^T \hat{Q}_N^{-1} A \right)^{-1}.
$$

Finally, we use the inner connection of the elements of $\mathbf{\zeta}_N$ to estimate the location vector $\mathbf{j}_N$. Defining $G_N = \begin{bmatrix} \hat{\mathbf{\zeta}}_{N}^{WLS}(2) & \hat{\mathbf{\zeta}}_{N}^{WLS}(3) \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$, where $\hat{\mathbf{\zeta}}_{N}^{WLS}(i)$ is the $i$th element of $\hat{\mathbf{\zeta}}_{N}^{WLS}$, we obtain

$$
G_N \mathbf{j}_N = \hat{\mathbf{\zeta}}_{N}^{WLS} + \mathbf{\epsilon}_N.
$$
where $\epsilon_N$ is a $[3 \times 1]$ estimation noise vector of $\hat{\zeta}_{N}^{\text{WLS}}$. Using (2.19), $\hat{Q}'_N$ from (2.18) and $\hat{\zeta}_{N}^{\text{WLS}}$ from (2.17), the WLS estimator of $j_N$ is

$$
\hat{j}_N = \left( G_N^T \hat{Q}'_N^{-1} G_N \right)^{-1} G_N^T \hat{Q}'_N^{-1} \hat{\zeta}_{N}^{\text{WLS}},
$$

(2.20)

whose elements $\hat{j}_N(1) = \hat{j}_N^x$ and $\hat{j}_N(2) = \hat{j}_N^y$ are the desired location coordinates.

We would like to mention that if assumption $\varsigma_l \varsigma_l' \sim 1$ used to obtain (2.13) does not hold, the localization process can be performed on a per-anchor-node basis. To this end, packet index $i$ in (2.13) is limited to packets transmitted or received by the same anchor node, and the number of equations (2.13) is reduced to $N - 1$ (assuming equal number of transmissions per anchor node in the network). Since $L$ is expected to be small (we use $L = 2$ in our simulations and sea trial described below), the accuracy of the localization process is not expected to deteriorate much. Then, the UL node location at the end of the localization window can be estimated by combining per-anchor-node based estimations $\hat{j}_N$ from (2.20) using data fusion techniques, cf., [93]. Since per-anchor-node based estimations $\hat{j}_N$ are independent of $\varsigma_l$, such combination is not affected by mismatch of $\varsigma_l$ between anchor nodes.

### 2.3.3 Extensions

In this section we introduce two extensions for the above location estimation. The first is a refinement step in which we iteratively improve the location estimation (2.20). The second is a self-evaluation process to test the accuracy of the localization process.

#### Iterative Refinement

The accuracy of estimation (2.20) depends on the quality of the coarse estimate $\hat{\zeta}_{N}^{\text{LS}}$ from (2.16), used to construct the error covariance matrix, $\hat{Q}_N$. We now follow [94] and propose an iterative refinement procedure in which the accuracy of $\hat{Q}_N$ is improved.

In the $k$th step of our iteration, vector $\hat{j}_{N,k}$ is estimated using (2.20) from which the vector $\hat{\zeta}_{N,k}$ is constructed. Next, in the $(k + 1)$th step $\hat{\zeta}_{N,k}$ replaces $\hat{\zeta}_{N}^{\text{LS}}$ in the construction of $\hat{Q}_N$. As a stopping criterion, we use the covariance matrix of the $k$th estimation (2.20),

$$
\hat{Q}''_{N,k} = \left( G_N^T \hat{Q}'_{N,k}^{-1} G_N \right)^{-1}.
$$

(2.21)

Since the determinant, $|\hat{Q}''_{N,k}|$, is directly proportional to the estimation accuracy [91], the iteration stops when the absolute value of $|\hat{Q}''_{N,k} - |\hat{Q}''_{N,k-1}|$ is below some empirically chosen threshold, $\Delta_{\text{iter}}$, or if the number of iterations exceeds its maximum, $N_{\text{iter}}$. While we could not prove the convergence of this process, we demonstrate it by means of numerical simulations in Section 2.5.
Self-Evaluation of Localization Performance

In this section, we describe a binary test for self-evaluating localization accuracy. It can be used to adjust STSL parameters, such as the localization window $W$, for refining the localization procedure, such as data fusion of per-anchor-node based localization (see discussion after (2.20)), or to decide whether an UL node can be used as a new reference node. For the latter application localization should be extended to tracking though, to make sure that location estimates remain accurate when nodes move. Our self-evaluation test relies on a widely used model that bounds propagation speed underwater between 1420 m/sec and 1560 m/sec [5]. In particular, given an estimate of the propagation speed, $c_{\text{est}}$, the binary test output $\xi$ is computed as

$$
\xi = \begin{cases} 
1, & \text{if } 1420 \leq c_{\text{est}} \leq 1560 \\
0, & \text{otherwise}
\end{cases},
$$

(2.22)

and $\xi = 1$ and $\xi = 0$ indicate accurate and non-accurate localization, respectively. Using (2.12) and since $\varsigma_{\text{UL}}$ are expected to be close to 1, we obtain the propagation-speed estimate as

$$
c_{\text{est}} = \frac{1}{N} \sum_{i=1}^{N} \frac{\|\hat{j}_i - p_i\|_2}{\hat{T}_{pd_i}},
$$

(2.23)

where $\hat{j}_i, i = 1, \ldots, N - 1$, follow from $\hat{j}_N$ in (2.20) using relation (2.11). Different from traditional self-evaluation techniques involving the broadcast of a confidence index obtained from comparing the measured propagation delay to the estimated one (e.g. [95, 30]), the advantage of the our method lies in the comparison of $c_{\text{est}}$ to a given model of propagation speed, which is independent of the estimation. Our numerical results (see Section 2.5) show that when localization error is accurate (i.e., below 10 m), we obtain $\xi = 1$ in more than 99% of the cases, and when localization error is non-accurate (i.e., above 10 m), $\xi = 0$ results in 90% of the cases. If more reliable evaluation performance is needed, the proposed test could be combined with other self-evaluation tests.

2.3.4 Scalability

When more network nodes are added, often fewer packets are transmitted per node. Hence, performances of both time-synchronization and localization degrade with increasing number of UL nodes, $N_{UL}$. Furthermore, since time-synchronization is performed per-anchor node, its performance degrades with increasing number of anchor nodes, $L$. However, since the number of propagation delay measurements, available for localization, increases with $L$, performance of the localization step by itself improves with $L$. However, since localization depends on the output of the synchronization step, the overall performance may improve or degrade with increasing number of anchor nodes. In summary, scalability of the STSL algorithm is closely related to the scalability of the underlying communications protocol.
2.3.5 Pseudo-Code for STSL

The operation of the STSL algorithm is summarized in the pseudo-code in Algorithm 1. For simplicity, the quantization mechanism introduced in Section 2.3.1 is not included, and we start when positions are already quantized into locations $u_{l,\nu}$ and $k_{\rho}$. First, equations (2.5) are formed, and an LS estimator is used to estimate clock skew and offset for each anchor node $l$ (lines 2-9). Then, the time-synchronization step is concluded by estimating propagation delays for each transmitted or received packet (line 12). The localization step begins with forming equations (2.13) (line 13), followed by an initial LS estimator (line 15). Then, an iterative procedure begins where in each step the covariance matrix $\hat{Q}_N$ and location $\hat{j}_{N,k}$ are estimated (lines 18-19). The latter is then used to refine the initial estimation by iteratively forming matrix $\hat{Q}_N$ till convergence is reached (lines 19-23). The algorithm performs a series of LS and WLS estimations with complexity of $O(N^3 + N)$ and is executed only once at the end of the localization window. A software implementation of the algorithm can be downloaded from [96].

2.4 Cramér-Rao Lower Bound

For the purpose of gauging the performance of the STSL algorithm, in this section we develop analytical expressions to lower bound the performance of any unbiased UWL estimator, assuming nodes not to be time-synchronized and propagation speed unknown. We start with general expressions for the CRLB, and then apply it to our specific localization problem.

2.4.1 General Cramér-Rao Lower Bound

Consider a measurement vector $y = h(\pi, \nu) + n$, where $n$ is a noise vector, and $h(\pi, \nu)$ is some function of a vector of wanted variables, $\pi$, and a vector of nuisance variables, $\nu$. For an unbiased estimator, the variance of the $n$th element of $\pi$, $\pi_n$, can be bounded by the CRLB [97]

$$E[(\hat{\pi}_n - \pi_n)^2] \geq \text{CRB}(\pi_n),$$

(2.24)

where $\text{CRB}(\pi_n) = \left(I^{-1}\right)_{n,n}$ and $I$ is the Fischer information matrix (FIM), whose $(n,m)$th element is

$$I_{n,m} = -E_y \left[ \frac{\partial^2 \ln P(y|\pi)}{\partial \pi_n \partial \pi_m} \right],$$

(2.25)

and $P(y|\pi)$ is the probability density function of $y$ given $\pi$. To calculate $P(y|\pi)$ one needs to average the nuisance variables, $\nu$, i.e., $P(y|\pi) = E_{\nu}[P(y, \nu|\pi)]$, which makes it hard to calculate (2.25), since often $P(y, \nu|\pi)$ cannot be expressed. Therefore,
Algorithm 1 Estimate $\hat{j}_N$

1: {Step 1: Time-synchronization}
2: for $(l = 1, \ldots, L)$ do
3:   for $(n \in \mathcal{N}_l^a, m \in \mathcal{N}_l^b)$ do
4:     if $(p_n, p_m \in \mathcal{U}_{\nu}) \cap (\hat{j}_n, \hat{j}_m \in \mathcal{K}_\rho)$ then
5:       Form equations (2.5) using $T_n, R_n, T_m$ and $R_m$
6:     end if
7:   end for
8:   Estimate $O_l$ and $S_l$ using (2.7)
9: end for
10: {Step 2: Localization}
11: for $(i = 1, \ldots, N)$ do
12:   set $\hat{T}_{pd}^i$ using (2.9)
13:   Form equations (2.13)
14: end for
15: Estimate $\hat{\zeta}_{N,1}$ using (2.16)
16: for $(k = 1$ to $N_{iter})$ do
17:   Estimate $\hat{Q}_N$ using $\hat{\zeta}_{N,k}$
18:   Estimate $\hat{j}_{N,k}$ using (2.20)
19:   Construct matrix $\hat{Q}_N''$ using (2.21)
20:   if $|\hat{Q}_{N,k}'' - \hat{Q}_{N,k-1}''| \leq \Delta_{iter}$ then
21:     Return
22: end if
23: Construct $\hat{\zeta}_{N,k+1}$ using $\hat{j}_{N,k}$
24: end for

instead of CRB($\pi_n$), the modified Cramér-Rao bound MCRB($\pi_n$) = $\left(\bar{I}^{-1}\right)_{n,n}$ is often used [98], where

$$\bar{I}_{n,m} = -E_{y,\nu} \left[ \frac{\partial^2 \ln P(y|\pi, \nu)}{\partial \pi_n \partial \pi_m} \right]. \quad (2.26)$$

In [98] it was shown that

$$\text{CRB}(\pi_n) \geq \text{MCRB}(\pi_n). \quad (2.27)$$

Hence, MCRB($\pi_n$) may be too loose to compare with.

A different approach would be to consider the nuisance variables $\nu$ as part of the estimation problem. That is, we consider a new variable vector $\Phi = [\pi^T, \nu^T]^T$ and formalize CRB($\Phi_n$) for

$$I_{n,m} = -E_y \left[ \frac{\partial^2 \ln P(y|\Phi)}{\partial \Phi_n \partial \Phi_m} \right]. \quad (2.28)$$
We note that CRB(\(\Phi_n\)) does not bound \(E[(\hat{\pi}_n - \pi_n)^2]\) but \(E[(\hat{\Phi}_n - \Phi_n)^2]\). Thus, it can only serve as a lower bound for estimators which estimate both \(\pi\) and \(\nu\).

2.4.2 Application to STSL

Since the STSL algorithm includes a sequence of LS and WLS estimators, it is an unbiased estimator. Thus, we next apply the MCRB(\(\pi_n\)) and CRB(\(\Phi_n\)) bounds for our STSL algorithm. We consider the measurement vector in (2.2) for which \(y = R_i, \pi = [j^x_N, j^y_N], \nu = [S_1, \ldots, S_L, O_1, \ldots, O_L, c]\), and \(n\) is as \(\gamma_i\) in (2.2). Then, we have

\[
E \left[ \left( \hat{j}^x_N - j^x_N \right)^2 + \left( \hat{j}^y_N - j^y_N \right)^2 \right] \geq \text{CRB}(\pi_1) + \text{CRB}(\pi_2). \tag{2.29}
\]

We note that the variance of \(y\) depends on the clock skew, \(S_i\). Thus, although \(\gamma_i\) is assumed Gaussian, the often used simplification for the CRB in the Gaussian case (cf. [97]) cannot be used to solve (2.26) and (2.28).

Following our discussion in Section 2.4.1, we consider the alternative CRLB formulations

\[
\text{CRB} = \text{CRB}(\Phi_1) + \text{CRB}(\Phi_2), \quad \text{MCRB} = \text{MCRB}(\pi_1) + \text{MCRB}(\pi_2), \tag{2.30}
\]

and compare \(\sqrt{\text{MCRB}}\) and \(\sqrt{\text{CRB}}\) to

\[
\rho_{\text{err}} = \sqrt{E \left[ \left( \hat{j}^x_N - j^x_N \right)^2 + \left( \hat{j}^y_N - j^y_N \right)^2 \right]}. \tag{2.31}
\]

2.5 Simulation Results

In this section, we present and discuss simulation and sea trial results demonstrating the performance of the STSL algorithm in different environments. We conducted 10,000 Monte-Carlo simulations of a scenario with two anchor nodes and one UL node, communicating in a simple TDMA fashion. The three nodes were placed uniformly in a square area of \(1 \times 1\) km\(^2\) and moved between two adjacent packet transmission times at uniformly distributed speed and angle between \([-5, 5]\) knots and \([0, 360]\) degrees, respectively. We added a zero mean i.i.d. Gaussian noise with variance \(\sigma^2\) to each of the ToA estimations [see (2.2)]. Furthermore, considering the results in [99] we added a zero mean i.i.d. Gaussian noise with variance 1 m\(^2\) to each of the distance elements of the motion vectors [see (2.3)] while regarding their angle components to be accurate. To simulate time-synchronization errors the clock of each of the three nodes had a Gaussian distributed random skew and offset relative to a common clock with mean values 1 and 0 sec and variances 0.001 and 0.5 sec\(^2\), respectively.
Figure 2.1: $\rho_{err}$ from (2.31) as a function of $1/\sigma^2$. Sound speed is known and all nodes are time-synchronized. Vertical bars show 95% confidence intervals of the simulation results for STSL.

We used a quantization threshold $\Delta = 38$ meters and a localization window of $W = 20$ time-slots. The time-slot duration was selected $T_{\text{slot}} = 5$ seconds, considering the long propagation delay in the UAC (e.g., 4 sec for a range of 6 km). We compare the performance of the STSL algorithm with those of the multilateration method [27] and the method proposed in [92], which we refer to as the joint localization and synchronization (JLS) algorithm. Both benchmark methods use an assumed propagation speed $\tilde{c}$. Furthermore, while the JLS algorithm performs joint time-synchronization and localization (assuming anchor nodes are time-synchronized but the UL node is not), the multilateration method assumes all nodes to be time-synchronized. Since both benchmark methods assume static nodes, we used a different simulation environment for them such that a fair comparison with the STSL algorithm is possible. The simulation environment for the benchmark methods considers fixed nodes and adds virtual anchor nodes according to node movements in the original simulation scenario (i.e., the one used to test the performance of the STSL algorithm). Consider, for example, an anchor node $l$ moving between locations $\mathbf{p}_{l_1}$ and $\mathbf{p}_{l_2}$ while communicating with a static UL node. To test the benchmark methods, such a scenario would change into a scenario where two static anchor nodes, $l^1$ and $l^2$, are located at $\mathbf{p}_{l_1}$ and $\mathbf{p}_{l_2}$, respectively. Allowing a fair comparison between the three tested localization methods, the virtual anchor nodes, $l^1$ and $l^2$, have the same local clock as that of the real anchor node $l$. The implementation code of the STSL algorithm can be downloaded from [96].

First, we consider a scenario where $c = \tilde{c} = 1500$ m/sec and all nodes are time-synchronized. Figure 2.1 shows $\rho_{err}$ from (2.31) as a function of $1/\sigma^2$ for the three
methods and the CRB and MCRB from (2.30). For clarity, here and in the following we show 95% confidence intervals in error bars only for the STSL algorithm. The results show that both benchmark methods achieve better performances than the STSL algorithm. This is mainly because STSL redundantly estimates \( c \) as well as clock offsets and skews, which introduces errors. This is also why the multilateration method achieves slightly better performance than the JLS protocol method. We note that the MCRB is slightly lower than the CRB and both bounds are quite close to the STSL error, which implies that although STSL is a heuristic estimator it achieves good localization results. To show the effect of possible mismatch of our model for the measurement noise \( \gamma_i \) (see (2.2)), Figure 2.1 also includes results for \( STSL\text{-}mix \), in which \( \gamma_i \) is modeled as a mixture of two distributions. The first distribution (with weight 0.9) is a zero mean Gaussian with variance \( \sigma^2 \) and the second (with weight 0.1) is a Rayleigh(\( \sigma \)) distribution, which accounts for multipath propagation [86]. From Figure 2.1, we observe that the performance of STSL-mix decreases compared to that in the Gaussian-noise case, which is mainly due to the non-zero mean of noise. However, this degradation is fairly moderate demonstrating some robustness of STSL to model mismatch.

In Figure 2.2 we compare \( \rho_{err} \) for the three methods when \( c = \bar{c} = 1500 \text{ m/sec} \), but nodes are not time-synchronized and \( \frac{1}{\sigma^2} = 46 \text{ dB} \), as a function of \( e_{sync} = \frac{\bar{S} + \bar{O} - W}{W} \), where \( \bar{S} \) and \( \bar{O} \) is the average of \( S_l \) and \( O_l \), \( l = 1, \ldots, L \), respectively. While the performance of the STSL algorithm is hardly affected by the synchronization error (compared to the results in Figure 2.1), the JLS protocol method, designed for time-synchronized anchor nodes, and the multilateration method suffer from significant estimation errors even for small synchronization errors.
Figure 2.3: $\rho_{err}$ from (2.31) as a function of $|c - \tilde{c}|$. All nodes are time-synchronized and $\frac{1}{\sigma^2} = 46$ dB. Vertical bars show 95% confidence intervals of the simulation results for STSL.

We now compare performance when $c$ is chosen with uniform distribution between the model boundaries, 1420 m/sec and 1560 m/sec, and the two benchmark methods were still given the nominal value $\tilde{c} = 1500$ m/sec. To understand the effect of mismatched propagation-speed information on localization accuracy, we compare $\rho_{err}$ from (2.31) when all nodes are time-synchronized. The results are shown in Figure 2.3 as a function of $|c - \tilde{c}|$, again for $\frac{1}{\sigma^2} = 46$ dB. We observe that for both benchmark methods, $\rho_{err}$ dramatically increases even for a small difference of $|c - \tilde{c}| = 10$ m/sec, which motivates the need to accurately estimate $c$ in UWL. Furthermore, compared to the results of Figure 2.1 the STSL is almost unaffected by the variations of $c$.

Next, we consider the practical case where all nodes are not time-synchronized (same scenario as for Figure 2.2) and $c$ is unknown (same scenario as for Figure 2.3). We study two of the properties of the STSL algorithm, namely the convergence of the refinement iterative process discussed in Section 2.3.3 and the self-evaluation process discussed in Section 2.3.3. In Figure 2.4, we demonstrate the convergence of the refinement iterative process by showing $\rho_{err}$ from (2.31), averaged over all clock offsets and skews and $c$ instances, as a function of the number of iteration steps and several values of $\frac{1}{\sigma^2}$. The results indicate that a significant performance improvement is achieved after only a few iteration steps. In Figure 2.5 we show the empirical probability density function (PDF) of the estimated propagation speed, $c_{est}$, from (2.23) when $c = 1500$ m/sec. The results are shown for two cases: 1) when $\rho_{err} \leq 10$ m and 2) when $\rho_{err} \geq 10$ m. The results show that for small values of $\rho_{err}$, in more than 99% of the cases, $c_{est}$ is inside the model boundaries (i.e., $1420 \leq c_{est} \leq 1560$), with a standard deviation of less than 10 m/sec. For large values of $\rho_{err}$, $c_{est}$ seems to be
almost uniformly distributed, with only 10% of the estimations being inside the model boundaries. However, for some applications (e.g., localization in sparse networks) this missed-detection probability may be too large. Thus, we conclude that \( c_{\text{est}} \) can serve as a good indicator to confirm accurate localization, but may be used to complement other self-evaluation techniques to identify non-accurate localization.

Finally, in Figure 2.6 we consider the same scenario as for Figure 2.4 and show \( \rho_{\text{err}} \) from (2.31) as a function of \( \frac{1}{\sigma^2} \). For clarity, since results for STSL are similar to those shown in Figure 2.1, error bars are omitted. We observe that while both benchmark methods suffer from a significant error floor, the error for the STSL algorithm decreases with \( \frac{1}{\sigma^2} \) and is the same as in Figure 2.1. Hence, the algorithm compensates for both synchronization and propagation speed uncertainties. To demonstrate the relation between the number of anchor nodes, \( L \), and the number of UL nodes, \( N_{\text{UL}} \) (see discussion in Section 2.3.4), in Figure 2.6 we also include results for \( L = 3, N_{\text{UL}} = 1 \) (STSL, \( L = 3 \)) and \( L = 2, N_{\text{UL}} = 2 \) (STSL, 2UL), which had similar standard deviation to the case of \( L = 2, N_{\text{UL}} = 1 \). We note that since multilateration does not include time-synchronization, and since JLS performs joint time-synchronization and localization, clearly their performance improves with \( L \) and degrades with \( N_{\text{UL}} \). Results show that, as expected, also performance of the STSL algorithm slightly improves with \( L \) and decreases with \( N_{\text{UL}} \).
Chapter 2. UWL with Time-Synchronization and Propagation Speed Uncertainties

2.6 Sea Trial Results

In this work we assumed 1) node’s clock skew and offset are time-invariant within the localization window, 2) propagation speed is time and space invariant for small depth differences, 3) propagation delay measurements are affected by a zero-mean Gaussian noise, and 4) node movements are relatively slow such that quantization of node locations is possible. While the first assumption depends on the system clock, the second and the third depend on the channel. To verify our assumptions and confirm our results we tested the STSL algorithm in a sea trial along the shores of Haifa, Israel in August 2010.

The sea trial included three drifting vessels, representing three mobile nodes, and lasted for $T_{\text{exp}} = 300$ minutes. In Figure 2.7, we show the UTM coordinates of the nodes during the sea trial. We note that node 3 needed to turn on its engines around time slot 150, which explains the sudden change in its direction and speed. Each node was equipped with a transceiver, deployed at 10 meters depth, allowing UWAC at 100 bps with a transmission range of 5 km. The nodes communicated in a TDMA network with a time-slot of $T_{\text{slot}} = 60$ seconds, allowing significant node motion between transmission of each packet. Time-slot management was performed at each node using an internal clock. These internal clocks were manually time-synchronized at the beginning of the experiment with an expected clock offset of up to one second. We also note that pre-testing of these clocks showed a clock skew of one second per day.

We used GPS receivers as reference for the location of each node as well as its inertial system to obtain motion samples [see (2.3)]. The localization error of the GPS-based reference locations was reported to be uniformly distributed between 0
Chapter 2. UWL with Time-Synchronization and Propagation Speed Uncertainties

and 10 m. To test the effect of this uncertainty in the anchor-node location we conducted simulations similar to the scenario considered in Figure 2.1 with error-free ToA measurement but with anchor-node location uncertainties similar to those of the GPS receivers in use. The results showed that using our STSL algorithm such uncertainty results in an average estimation error of 15 m. Thus, in the sea trial any location error below 15 m is considered accurate.

2.6.1 Channel and System Characteristics

At the beginning and end of the sea trial we measured the propagation speed in water using a measuring probe. Both measurements showed that the propagation speed $c$ was bounded in between 1552 m/sec (for depth of 40 m) and 1548 m/sec (for depth of 1 m) and was on average 1550 m/sec. The small variance of the measurements of $c$ confirms our assumption that the propagation speed can be considered fixed throughout the localization window. In the following, for performance evaluation we consider $c = 1550$ m/sec, which is within the boundaries of our model (see Section 2.2) but is different from the commonly used value of 1500 m/sec.

In Figure 2.8 for each pair of nodes we show $\hat{T}^{pd}_{diff}$, which is the time difference between propagation delay estimations at both sides of the communication link in a single set of receiver-transmitter quantized locations, measured directly from (2.2a) and (2.2b) neglecting the clock skew and offset. For example, for a two-way packet transmission between the quantized locations $k_\rho$ and $u_{l,\nu}$ with propagation delay estimations $\hat{T}^{pd}_1$ and $\hat{T}^{pd}_2$, $\hat{T}^{pd}_{diff} = |\hat{T}^{pd}_1 - \hat{T}^{pd}_2|$. We note that if nodes are time-synchronized, we would expect $\hat{T}^{pd}_{diff}$ to be on the same order of the length of the impulse response, which was measured as 20 msec on average and did not exceed
30 msec. However, the results show that $\hat{T}_{pd}^{\text{diff}}$ increases with time and is much greater than 30 msec. This implies that nodes suffered from considerable clock skew and offset. Furthermore, since the values of $\hat{T}_{pd}^{\text{diff}}$ are different for each pair of nodes, the nodes skew and offset are different, which confirms with our system model (see Section 2.2).

### 2.6.2 Results

In the following, we compare the performance of the STSL algorithm in the sea trial with that of a method aimed to solve a relaxed sequential time-synchronization and localization (R-STSL) problem, in which an a-priori propagation speed $\tilde{c}$ is given. In the R-STSL, time-synchronization is performed similar to the process described in Section 2.3.1, but the localization process is modified as $c$ is known. The results are shown for all three nodes, where each time a different node was considered as the UL node and the other two nodes were the anchor nodes. We measure the performance in terms of the Euclidean distance between the estimated location and the reference GPS location, averaged over a sliding localization window of $W$ time-slots, i.e.,

$$\bar{\rho}_{\text{err}} = \frac{1}{\frac{T_{\text{exp}}}{T_{\text{slot}}} - W + 1} \sum_{n=W}^{T_{\text{exp}}/T_{\text{slot}}} \| \hat{j}_n - \hat{j}_n \|_2,$$  \hspace{1cm} (2.32)

where for each UL node location estimation $\hat{j}_n$, we used ToA and inertial system measurements from time-slot $n - W + 1$ till $n$.

In Figure 2.9, we demonstrate the effect of mismatched propagation speed, i.e., $\tilde{c} \neq c$, by showing $\bar{\rho}_{\text{err}}$ from (2.32) for $W = 30$ time-slots as a function of $|c - \tilde{c}|$, where $c = 1550$ m/sec. We note that although such choice of $W$ seems large due
to the long time slot duration, the number of transmissions for each node was only 10, which is in the same order as considered in our simulations. The results show that $\bar{p}_{err}$ significantly increases with $|c - \tilde{c}|$ even for a relatively small difference of 10 m/sec.

This result, as well as the results in Figure 2.3, validate the need to consider the propagation speed as an additional variable in UWL. In the following we consider a matched version of R-STSL (MR-STSL), i.e., when $\tilde{c} = c$, which in the absence of benchmark localization methods that take into account time-synchronization uncertainties and availability of inertial system to track short-term node movements, can serve as a lower bound for the STSL.

Finally, in Figure 2.10 we show the empirical cumulative density function CDF of $\bar{p}_{err}$, averaged over the three nodes, for STSL and MR-STSL and $W = 10, 20, 30$ time-slots, i.e., in a single localization window an average number of packet transmissions of 3.3, 6.6, 10 for each node, respectively. We observe that both mean and variance of $\bar{p}_{err}$ improve with $W$, however, at a cost of delay. We observe that since STSL estimates an additional variable, its performance is worse than that of MR-STSL. However, the difference is not significant. We note that the average $\bar{p}_{err}$ for STSL and $W = 30$ time-slots is 21.5 m, which is close to the expected localization accuracy due to the GPS location uncertainties. Therefore, STSL fully compensates the large clock skew and offset shown in Figure 2.8, node movements and propagation speed uncertainty.

Figure 2.8: $\hat{T}_{d\text{iff}}$ for all communication links as a function of time slots.
Chapter 2. UWL with Time-Synchronization and Propagation Speed Uncertainties

Figure 2.9: $\bar{\rho}_{err}$ from (2.32) as a function of $|c - \tilde{c}|$ for $W = 30$ time slots. R-STSL method.

2.7 Summary

In this chapter, we considered UWL in the practical scenario where nodes are not time-synchronized and permanently moving, and where the propagation speed is unknown. We introduced a localization algorithm which uses existing self-estimations of motion vectors of nodes, assumed to be accurate for short periods of time. The algorithm utilizes the constant movements of nodes in the channel and relies on packet exchange to acquire multiple ToA measurements at different locations. We also presented a method to self-evaluate the localization accuracy of the node. In addition, we used the applicable Cramér-Rao lower bounds as references for the performance of STSL. Considering the problem of establishing a faithful simulation environment for the UAC, alongside simulations we tested our algorithm in a designated sea trial. Both simulations and sea trial results demonstrated that our algorithm can cope with time-synchronization and propagation speed uncertainties in a dynamic environment, and achieves a reasonable localization accuracy using no more than two anchor nodes.
Figure 2.10: Probability that $\bar{\rho}_{\text{err}} \leq x$ for STSL and MR-STSL.
Chapter 3

Spatially Dependent Underwater Navigation

Since nodes constantly move in the UAC, UWL is only the first step towards underwater navigation. At the presence of motion, underwater navigation must include a tracking scheme that uses UWL as an initial estimation and recursively updates the location of a TN. A key challenge in UT is motion irregularities, which makes it hard to determine the SSM. In this context, since the ocean current is usually spatially correlated [100], it is reasonable to assume spatial dependencies between the drift motions of network nodes in close proximity. In [101], it was argued that birds in a flock or school of fish obey simple rules for distance and speed relative to others in close proximity. With this idea in mind, [102] assumes temporal and spatial dependencies between the motion of nodes in an indoor environment and achieves good tracking performance. For UWL, [103] exploited spatial dependencies for the collaborative localization of fleets of vertically sinking drifters assuming equal speed. For tracking, an acoustic Doppler current profiler is used in [104] to measure ocean currents at different depths and update the SSM accordingly. Ocean current is treated as a separate state parameter in [105], and both the self-propelled and drift speed of the AUV are estimated using an EKF. However, while spatial dependencies between the drifts of the anchor nodes is observed from sea trial results in [105], to the best of our knowledge it has not been incorporated in a UT scheme yet.

In this chapter, we propose a UT scheme that accounts for the effect of ocean current, considers sound speed uncertainties, and incorporates Doppler shift measurements. To the best of our knowledge, neither of these three components has been considered before for UT. We exploit that the ocean current is spatially correlated, and causing correlated drift velocities of nodes participating in the tracking. In particular, by letting anchor nodes report their drift velocities through acoustic communication, we estimate the drift velocity of the TN as a combination of the former. We therefore refer to our proposed tracking solution as the drift dependent UT (DD-UT) scheme.

We offer two SSM-based tracking solutions, which are based on the EKF and the unscented Kalman filter (UKF), respectively. The EKF is a modification to the Kalman filter which linearizes the state-space and measurement model using the current predicted state. While the EKFs are extensively used in both GPS and underwater navigation [34, 106], they require knowledge of the probability density function of both the model and measurement noise and thus are sensitive to model
mismatch [107, 108]. Instead, the UKF approximates the probability density function by a deterministic sampling of points. If a large amount of data is available, the UKF tends to be more robust than the EKF in its estimation of error and has been proven superior to the EKF for complex cases such as time series modeling and neural network training [107]. However, no such comparison has been made for the case of UT.

Our tracking scheme starts from initial estimates of the sound speed, the location of the TN and its speed. Then, our SSM fuses INS measurements, drift-velocity information from anchor nodes, and ranging and Doppler shift estimates to anchor nodes, to provide timely estimates of the 2-D location of the TN. Considering the possibility of weak correlations between drift velocities (for example in the presence of turbulence), we present two types of confidence indices (CIs). The first one is based on the distance between TN and anchor node and the corresponding measured Doppler shift, and the second CI is based on the normalized variance of the anchors’ velocities. Since these CIs do not depend on the estimated location of the TN, we argue that they are unbiased. To evaluate the performance of our DD-UT scheme, we develop a hybrid simulator combining the shallow water hydrodynamic finite element model (SHYFEM) for ocean current [109] and the Bellhop ray-tracing numerical model for power attenuation of sound in water [2]. For a set of bathymetry maps, our simulation provides time-varying trajectories of drifting nodes along with power attenuation for all communication links. We compare the performance of our DD-UT scheme to benchmark solutions, as well as to the recursive Cramér-Rao lower bound applicable for tracking. To further verify our simulations, we also report results from two sea trials for different bathymetric channel structures, conducted in the Mediterranean Sea and in the Indian Ocean.

The remainder of this chapter is organized as follows. In Section 3.1, we introduce our system model, which is followed by the discussion of the state-space and measurement model in Section 3.2. Our DD-UT scheme is introduced in Section 3.3. Next, results from both simulations and sea trials are presented in Section 3.4. Finally, conclusions are drawn in Section 3.5.

### 3.1 System Model

Our system includes several anchor nodes at known locations and a TN, equipped with a depth sensor, an INS, and an acoustic modem. The TN is assumed moving with random acceleration in both the surge and angular direction. For simplicity, we neglect the pitch and roll Euler angles and assume that the vehicle is aligned with the horizontal plane or its measurements from on-board sensors can be projected to this plane. Furthermore, instead of the depth-dependent sound speed, $c$, for each communication link between the anchor node and the TN, we use the average sound speed $\bar{c}$ (see Section 1.1.1). We assume the anchor nodes remain at the same depth,
but the TN can rise or dive in water. Hence, at time instance \( t_k \), where \( k \) is the sampled time index of the INS system, \( \bar{c}_k \) is time-varying but similar for all anchor nodes. To estimate \( \bar{c}_k \) we adopt the widely used model for sound speed [5],

\[
c_k = 1449 + 4.6T_k - 0.055T_k^2 + 0.00037T_k^3 + (1.39 - 0.012T_k)(S_k - 35) + 0.017z_k, \tag{3.1}
\]

where \( T_k \) is the temperature in degree Celsius, \( S_k \) is the salinity in parts-per-thousand, and \( z_k \) is the depth in meters, and assume that \( \bar{c}_k \) is the mean of \( c_k \) from (3.1) over the water column from the transmitting anchor to the TN. We assume a prior UWL process (e.g., [110, 14]) that time-synchronizes the TN relative to the anchor nodes, and gives an initial estimation for \( \bar{c}_0 \) and for the TN location and heading.

We focus on 2-D location tracking. This is accomplished by the on-board INS giving timely estimates of the speed of the TN in both the surge and angular direction [34], and by packets transmitted by anchor nodes providing anchor-location and drift-velocity, ToF and Doppler shift information. The INS data rate (usually around 100 Hz) is assumed to be much faster than that of the packet exchange. Hence, we define intervals \( \tau_{\text{INS}} \) and \( \tau_{\text{range}} \), representing the time elapsed between two consecutive acceleration and ranging measurements, respectively, and for simplicity assume \( \tau_{\text{range}} = \tau_{\text{INS}} \). We define a reference grid system \([x, y, z]_k\), where \( x \) and \( y \) are UTM coordinates, and \( z \) is the depth in meters. The communication packets carry the 3-D coordinates of the anchor nodes in the reference grid system, \( \mathbf{r}_{\text{anc}}^k = [x_{\text{anc}}^k, y_{\text{anc}}^k, z_{\text{anc}}^k]^T \), and the anchor’s estimated drift vector, \( \mathbf{v}_{\text{anc,drift}}^k = [v_{x,\text{anc,drift}}^k, v_{y,\text{anc,drift}}^k, v_{\phi,\text{anc,drift}}^k, \phi_{\text{anc,drift}}^k]^T \), whose elements are the speed in the \( x, y \), and angular directions, and the anchor’s heading direction, respectively. We assume a slowly changing ocean current velocity field [100], such that \( \mathbf{v}_{\text{anc,drift}}^k \) and the drift vector of the TN are correlated. Vector \( \mathbf{v}_{\text{anc,drift}}^k \) is measured at the anchor nodes by subtracting the self-propelled motion, \( \mathbf{v}_{\text{thrust}}^k \), from the calculated location-based one. Following [39] (and references therein), the thrust velocity can be obtained by measuring the thrust force \( F_{\text{force}} \), and solving the differential equation

\[
m\ddot{v}_{\text{thrust}}^k = F_{\text{force}} - 0.5C_{\text{drag}}\rho A(v_{\text{thrust}}^k)^2, \tag{3.2}
\]

where \( m \) is the vehicle mass, \( C_{\text{drag}} \) is the vehicle drag coefficient, \( \rho \) is the density of the water, and \( A \) is the vehicle cross-section area. Then, \( \mathbf{v}_{\text{anc,drift}}^k \) can be estimated using a simple IIR filter, whose output is \( \alpha \mathbf{v}_{\text{anc,drift}}^k + (1 - \alpha)\mathbf{v}_{\text{anc,drift}}^{k-1} \) with parameter \( \alpha \) tuned to the expected rate of change of the drift velocity.

### 3.2 The SSM and Measurement Model

In this section, we describe our SSM and measurement model. We consider motion at fixed speed and allow for acceleration noise. Model mismatches are considered by
Chapter 3. Spatially Dependent Underwater Navigation

including drift velocity estimates of anchor nodes.

3.2.1 State Space Model (SSM)

Let \( \mathbf{r}_k = [x_k, y_k, z_k]^T \) and \( \phi_k \) be the 3-D coordinates of the TN in the reference grid system and its heading angle, respectively. Furthermore, let \( v^x_k, v^y_k \) and \( v^\phi_k \) be the velocity of the TN in the reference-grid \( x \) and \( y \) axes and in the angular direction, respectively. Since the TN is assumed to be moving with random acceleration in both the surge and angular direction, and since the average sound speed, \( \bar{c}_k \), is assumed time-varying, we choose the SSV as

\[
\mathbf{a}_k = [x_k, v^x_k, y_k, v^y_k, \phi_k, v^\phi_k, \bar{c}_k]^T.
\] (3.3)

To set up the SSM we assume a Gaussian distributed acceleration and mismatch for model (3.1), respectively. Recall that the sound speed in (3.1) depends on water salinity, temperature, and depth. For small depth changes of a few hundreds of meters, we can neglect changes in salinity. Furthermore, up to a water depth of 100 m, \( T_k \propto \frac{3}{100} z_k \) [5]. Thus, neglecting the second and third order terms of the temperature dependence in (3.1), we can linearly update the average sound speed, \( \bar{c}_k \), based on the depth change \( \Delta z_k = (z_k - z_{k-1}) + (z^\text{anc}_k - z^\text{anc}_{k-1}) \) at a rate of \( 0.017 + 4.6 \cdot \frac{3}{100} \). The assumed SSM is therefore

\[
\mathbf{a}_k = B \mathbf{a}_{k-1} + \mathbf{u} \Delta z_k + \mathbf{N} \mathbf{n}_k^a, \quad (3.4)
\]

where \( \mathbf{u} = [0, 0, 0, 0, 0, 0, 0.155]^T \), \( \mathbf{n}_k = \begin{bmatrix} n^x_k, n^y_k, n^\phi_k, n^\bar{c}_k \end{bmatrix}^T \) is a zero-mean Gaussian vector with covariance matrix \( \mathbf{R}^\text{model} \), and the advance and noise matrices are

\[
B = \begin{bmatrix}
1 & \tau_1^\text{INS} & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & \tau_1^\text{INS} & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & \tau_1^\text{INS} & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}, \quad N = \begin{bmatrix}
(\tau_1^\text{INS})^2 & 0 & 0 & 0 \\
0 & (\tau_1^\text{INS})^2 & 0 & 0 \\
0 & 0 & (\tau_1^\text{INS})^2 & 0 \\
0 & 0 & 0 & (\tau_1^\text{INS})^2 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\end{bmatrix},
\]

respectively.

3.2.2 Measurement Model

While AUVs can be equipped with a large number of on-board sensors, for long term missions, strict energy constraints may not permit the use of energy consuming sensors such as DVLs (e.g., [34]). We therefore consider UT using only an INS and occasional packet exchanges with anchor nodes. In the following, we state our measurement model.
INS

The on-board INS includes a 3-D accelerometer and angular sensors to measure the TN speed in the surge and angular direction [34],

$$\mathbf{m}^\text{INS}_k = \begin{bmatrix} v^x_k' \\ v^y_k' \\ v^\phi_k \end{bmatrix} + \mathbf{n}^\text{INS}_k,$$

(3.5)

where $v^x_k'$ and $v^y_k'$ are the speed of the TN in the $x$ and $y$ directions of the TN local coordinate system, such that $v^x_k = v^x_k' \cos \phi_k$, and $v^y_k = v^y_k' \sin \phi_k$. Vector $\mathbf{n}^\text{INS}_k$ in (3.5) is a 3-D measurement error vector with zero-mean Gaussian elements and covariance matrix $\mathcal{R}^\text{INS}$.

ToF

At a certain time sample $k$, the TN receives a packet from an anchor node in its communication range and estimates the ToF to the transmitting anchor. We assume that both the TN and the anchor move slowly relative to the propagation delay in the channel. Thus, the ToF is modeled by

$$m^\text{ToF}_k = \frac{\| \mathbf{d}_k \|}{c_k} + n^\text{ToF}_k,$$

(3.6)

where $\mathbf{d}_k = r_k - r^\text{anc}_k$ and $n^\text{ToF}_k$ is the ToF estimation error. While noise $n^\text{ToF}_k$ can be biased due to NLOS false identification, we rely on our method from Chapter 4 to classify NLOS and LOS ToF measurements, and assume $n^\text{ToF}_k$ is a zero-mean Gaussian with variance $\sigma^2_{\text{ToF}}$.

Doppler Shift

Along with estimating the ToF from received packets, communication signals can be used to evaluate the Doppler shift using, e.g., [111]. The Doppler shift is determined by the velocity difference of the anchor and the TN. Let $\mathbf{v}^\text{anc}_k = [v^x_k, v^y_k, v^\text{anc}_k]_T$ be the velocity$^3$ of the anchor node in the reference-grid $x$- and $y$-axis whose packet is received at time instance $t_k$. Velocity $\mathbf{v}^\text{anc}_k$ can be either reported by the anchor, or, assuming slow changes relative to $\tau^\text{range}$, be calculated by the TN using the anchor’s previous reported location. For slow motion, the Doppler shift can be approximated in terms of the frequency offset

$$\Delta f_k = \frac{f_c}{c_k} \| \mathbf{v}^\text{rel}_k \| \cos \theta_k,$$

(3.7)
where \( f_c \) is the carrier frequency, \( \mathbf{v}_{k}^{\text{rel}} = [v_{kx}, v_{ky}]^T - \mathbf{v}_{k}^{\text{anc}} \), and \( \theta_k \) is the angle between vectors \( \mathbf{v}_{k}^{\text{rel}} \) and \( \mathbf{d}_k \) (see (3.6)), such that

\[
\cos \theta_k = \frac{\mathbf{d}_k^T \cdot \mathbf{v}_{k}^{\text{rel}}}{\|\mathbf{v}_{k}^{\text{rel}}\| \|\mathbf{d}_k\|}.
\]  

(3.8)

By (3.7) and (3.8),

\[
\frac{\Delta f_k}{f_c} = \frac{(v_{kx}^x - v_{k,\text{anc}}^x)(x_k - x_{k,\text{anc}}) + (v_{ky}^y - v_{k,\text{anc}}^y)(y_k - y_{k,\text{anc}})}{\|\mathbf{d}_k\| \bar{c}_k}.
\]  

(3.9)

We model the Doppler shift measurement by

\[
m_{k,\text{Doppler}} = \frac{\Delta f_k}{f_c} + n_{k,\text{Doppler}},
\]  

(3.10)

where \( n_{k,\text{Doppler}} \) is assumed zero-mean Gaussian noise with variance \( \sigma^2_{\text{Doppler}} \).

**Velocity**

The spatial dependencies between the drift motions of the TN and the anchor nodes allow us to estimate the drift velocity of the TN by superimposing the drift velocities of the anchor nodes. In particular, by letting anchor nodes report their drift velocities and heading directions, \( \mathbf{v}_{k}^{\text{anc,drift}} \), we obtain an estimate for the TN drift velocity and heading direction

\[
\hat{\mathbf{v}}_{k,\text{drift}} = \begin{bmatrix}
\hat{v}_{kx,\text{drift}} \\
\hat{v}_{ky,\text{drift}} \\
\hat{v}_{k,\phi,\text{drift}} \\
\end{bmatrix}.
\]  

(3.11)

This process will be discussed in detail in Section 3.3.

We treat \( \hat{\mathbf{v}}_{k,\text{drift}} \) as a measurement, modelled as

\[
\hat{\mathbf{v}}_{k,\text{drift}} = \gamma_{k}^{d} \begin{bmatrix}
v_{kx} \\
v_{ky} \\
\phi_k \\
\end{bmatrix} + \mathbf{n}_{k,\text{drift}},
\]  

(3.12)

where \( \mathbf{n}_{k,\text{drift}} \) is an error vector whose elements are zero-mean Gaussian with covariance matrix \( \mathcal{R}_{\text{drift}} \), and \( \gamma_{k}^{d} \) is a CI whose purpose is to limit the use of anchor velocities when spatial dependencies between the motions of the TN and the anchor nodes are deemed to be weak (a method to determine \( \gamma_{k}^{d} \) will be presented in Section 3.3). We also assume that the TN is drifting or that, if self-propelled, it can compensate for its self-propelled motion using (3.2).
Chapter 3. Spatially Dependent Underwater Navigation

Measurement Vector

Using measurements $m^\text{INS}_k$ from (3.5), $m^\text{ToF}_k$ from (3.6), $m^\text{Doppler}_k$ from (3.10), and $\hat{v}_k^\text{drift}$ from (3.12), we form the measurement vector

$$y_k = \begin{bmatrix} m^\text{INS}_k \\ \gamma^p_k m^\text{ToF}_k \\ \gamma^p_k m^\text{Doppler}_k \\ \gamma^p_k \hat{v}_k^\text{drift} \end{bmatrix} = h(a_k) + n^\text{measure}_k,$$

(3.13)

where $\gamma^p_k = 1$ if a packet is received at time instance $t_k$ and 0 otherwise, $h(a_k)$ is the measurement model vector, and $n^\text{measure}_k$ is a zero-mean Gaussian measurement noise with covariance matrix $R^\text{measure}$. For the SSV $a_k$ in (3.3), the SSM (3.4), and the measurement vector $y_k$ from (3.13), we next derive the CRLB for UT.

3.2.3 Cramér Rao Lower Bound (CRLB)

Let $\hat{A}_k = \{\hat{a}_1, \ldots, \hat{a}_k\}$ and $\hat{Y}_k = \{\hat{y}_1, \ldots, \hat{y}_k\}$ be the set of SSV estimates (3.3) and measurement vectors (3.13) obtained till time instance $t_k$, respectively. For any unbiased estimator, the CRLB gives the lower bound on the variance

$$E \left[ (\hat{A}_k - A_k) (\hat{A}_k - A_k)^T \right] \geq J^{-1}(k),$$

(3.14)

where

$$J(k) = E \left[ -\frac{\partial^2}{\partial^2 A_k} \log P(A_k, Y_k) \right]$$

(3.15)

is the inverse of the Fisher information matrix with elements $J_{i,j}(k)$, $P(\cdot)$ denotes the probability density function, and $E[\cdot]$ denotes expectation. In [112], it was shown that if only estimation of $a_k$ is of interest, (3.15) can be formulated recursively such that

$$J(k) = J_{1,k} - J_{2,k} (J(k-1) + J_{3,k})^{-1} J_{2,k},$$

(3.16)

where

$$J_{1,k} = -E \left[ \frac{\partial^2}{\partial^2 a_k} \log P(a_k|a_{k-1}) \right] - E \left[ \frac{\partial^2}{\partial^2 a_k} \log P(y_k|a_k) \right],$$

(3.17a)

$$J_{2,k} = -E \left[ \frac{\partial^2}{\partial a_{k-1} \partial a_k} \log P(a_k|a_{k-1}) \right],$$

(3.17b)

$$J_{3,k} = -E \left[ \frac{\partial^2}{\partial^2 a_{k-1}} \log P(a_k|a_{k-1}) \right].$$

(3.17c)

Since both $n^a_k$ from (3.4) and $n^\text{measure}_k$ from (3.13) are modeled to be zero-mean Gaussians with corresponding covariance matrices $R^\text{model}$ and $R^\text{measure}$, respectively,
and since $\frac{\partial a_k}{\partial a_{k-1}} = B$ and introducing $H_k = \frac{\partial h(a_k)}{\partial a_k}$, (3.17) becomes

$$J_{1,k} = \left( N^T R_{\text{model}} N^T \right)^{-1} + E \left[ H_k^T \left( R_{\text{measure}} \right)^{-1} H_k \right]$$  \hspace{1cm} (3.18a)

$$J_{2,k} = B_k^T \left( N^T R_{\text{model}} N^T \right)^{-1}$$  \hspace{1cm} (3.18b)

$$J_{3,k} = B_k^T \left( N^T R_{\text{model}} N^T \right)^{-1} B_k$$  \hspace{1cm} (3.18c)

### 3.3 The DD-UT Scheme

In this section, we describe our DD-UT scheme. We start with the tracking algorithm and then proceed with the process of estimating the drift velocity of the TN.

#### 3.3.1 Tracking

Our tracking scheme is based on the SSM (3.4) with measurement vector $y_k$ and measurement model (3.13). For the purpose of tracking, we consider the use of the KF which is an optimal solution for Gaussian model and measurement noise. Since our measurement model is not linear, we adopt the EKF and the UKF. The EKF is formalized by

$$a_{k|k-1} = B a_{k-1|k-1} + u \Delta z_{k-1}$$  \hspace{1cm} (3.19a)

$$P_{k|k-1} = B P_{k-1|k-1} B^T + N^T R_{\text{model}} N^T$$  \hspace{1cm} (3.19b)

$$K_k = P_{k|k-1} H_k^T \left( H_k P_{k|k-1} H_k^T + R_{\text{measure}} \right)^{-1}$$  \hspace{1cm} (3.19c)

$$e_k = y_k - h(a_{k|k-1})$$  \hspace{1cm} (3.19d)

$$a_{k|k} = a_{k|k-1} + K_k e_k$$  \hspace{1cm} (3.19e)

$$P_{k|k} = (I - K_k H_k) P_{k|k-1}$$  \hspace{1cm} (3.19f)

where $I$ is the identity matrix, $P_{0|0}$ is taken as $R_{\text{model}}$, and $a_{0|0}$ is obtained from a prior UWL process (see Chapter 2). The UKF requires more regression steps, and for brevity we refer the reader to [108].

We initiate $R_{\text{model}}$ and $R_{\text{measure}}$ as diagonal matrices. For the former, since slow acceleration is expected and assuming no bathymetric layer exists (i.e., model (3.1) holds), we use small variance elements relative to the initial SSV (3.3) given by the UWL process. Similarly, since both ToF and Doppler shift measurements are expected to be fairly accurate, small values are used for $\sigma^2_{\text{ToF}}$ from (3.6) and $\sigma^2_{\text{Doppler}}$ from (3.10) relative to the initial estimates of range and velocity difference to anchor nodes, respectively. To determine $R_{\text{INS}}$ we use the INS specifications, and to determine $\sigma^2_{\text{drift}}$, we use a self-estimation of the error in the drift velocity estimation introduced next.
3.3.2 Drift Velocity Estimation

In this section we describe the process of obtaining estimates $\hat{v}^\text{drift}_k$ as a combination of $v^\text{anc,drift}_k$ measurements, which is then integrated into the measurement vector (3.13). Our drift velocity estimator requires ranging measurements $\|d_k\| = m_k^\text{ToF} \hat{c}_{k-1}$, assumed to be correct. However, we require only a coarse range estimation, mainly to weight drift velocities from different anchor nodes.

Considering the possibility of weakly correlated motion of nodes, we restrict the use of drift velocities reported from anchor nodes. First, accounting for time changes in the ocean current, temporal restriction is performed by defining a time window, $T^{\text{win}}$ (say 60 seconds), and considering only velocities of anchor nodes whose packets were received in the last $T^{\text{win}}$ sec. Second, spatial restriction is applied by using a CI, $\gamma_k^r$, indicating whether to use a velocity estimate received from a certain anchor node at time instance $t_k$. This is formalized through the set

$$\mathcal{K}_k = \{k' \mid t_k - T^{\text{win}} \leq t_{k'} \leq t_k \cap \gamma_{k,k'} = 1\}, \quad (3.20)$$

which contains the time instances for which drift velocities reported from anchor nodes can be used. Furthermore, we use a second CI, $\gamma_k^d$, which evaluates the accuracy of estimation $\hat{v}^\text{drift}_k$ and is incorporated into (3.12).

Next, we present three schemes for fusing $v^\text{anc,drift}_k$ measurements. Then, we formalize the CIs $\gamma_k^r$ and $\gamma_k^d$.

**Nearest Neighbor**

Assuming that the spatial correlation of ocean current decreases with range, the simplest method to obtain $\hat{v}^\text{drift}_k$ is by choosing it to be the velocity of the anchor nearest to the TN. Accordingly, we identify

$$\hat{k} = \arg\min_{k' \in \mathcal{K}_k} \|d_{k'}\|, \quad (3.21)$$

and set

$$v^\text{drift}_k = v^\text{anc,drift}_\hat{k}. \quad (3.22)$$

The main drawback of this method is that not all drift velocity information available from anchor nodes is used. Thus, it is sensitive to noise in estimations $v^\text{anc,drift}_k$, and to irregularities of the ocean current.

**Weighted Superposition**

In the weighted superposition method, velocities of anchor nodes are combined such that

$$\hat{v}^\text{drift}_k = \frac{\sum_{k' \in \mathcal{K}_k} \omega_{k'} v^\text{anc,drift}_k}{\sum_{k' \in \mathcal{K}_k} \omega_{k'} }, \quad (3.23)$$

where $\omega_{k'}$ is a weight function that depends on the distance between anchor nodes and the TN.
where \( \omega_{k'} \) is a weight function.

Since spatial correlation of ocean current depends on both range and depth (cf. [100]), a pragmatic choice is a normalized weight function

\[
\omega_{k'} = \left( \frac{\|d_k\|}{\max_{k \in K} \|d_k\|} \right)^{-\xi},
\]

where \( \xi \) is a pre-determined exponent. However, since ocean current irregularities (e.g., turbulence) occur at small regions [100], anchor nodes located in close proximity to each other should be given smaller weights. Thus, the spatial density of anchor nodes should also have an impact. Following [113], we suggest the alternative weighting function

\[
\hat{\omega}_{k',k} = \left( \frac{\|d_k\|}{\max_{k \in K} \|d_k\|} \right)^{-\xi} (1 + h_{k',k}),
\]

which takes into account the locations of the TN and the anchor nodes. In (3.25),

\[
h_{k',k} = \frac{1}{\sum_{j \in K, j \neq k'} \|d_j\|} \sum_{j \in K, j \neq k'} \frac{1}{\|d_j\|} \left( 1 - \frac{\|r_{k',j} - r_k\|}{\|d_{k'}\| \|d_j\|} \right).
\]

The exponent \( \xi \) in (3.24) and (3.25) is adapted to the expected noise level of \( v_{anc, drift}^k \). Our simulation results showed that when noise in estimates \( v_{anc, drift}^k \) is high, good performance is obtained for \( \xi = 0 \), whereas a value \( \xi = 4 \) should be used for relatively accurate \( v_{anc, drift}^k \) estimations.

**Least Squares**

Assuming that ocean current changes linearly in space, estimating the drift velocity of the TN can also be interpreted as estimating two planes, one each for the \( x \)- and \( y \)-coordinates of the velocity. Enumerating the elements in \( K_k \) from (3.20) as \( k_1, k_2, \ldots, k_{L_k} \) where \( L_k = |K_k| \), we formulate the problem as the linear combination

\[
\begin{pmatrix}
  x_{anc, k_1}^1 & y_{anc, k_1}^1 & \phi_{anc, k_1}^1 & 1 \\
  x_{anc, k_2}^1 & y_{anc, k_2}^1 & \phi_{anc, k_2}^1 & 1 \\
  \vdots & \vdots & \vdots & \vdots \\
  x_{anc, k_{L_k}}^1 & y_{anc, k_{L_k}}^1 & \phi_{anc, k_{L_k}}^1 & 1 \\
\end{pmatrix}
\begin{bmatrix}
  a_x & a_y & a_\phi \\
  b_x & b_y & b_\phi \\
  c_x & c_y & c_\phi \\
  d_x & d_y & d_\phi \\
\end{bmatrix}
= \begin{pmatrix}
  x_{anc, drift, k_1} & y_{anc, drift, k_1} & \phi_{anc, drift, k_1} \\
  x_{anc, drift, k_2} & y_{anc, drift, k_2} & \phi_{anc, drift, k_2} \\
  \vdots & \vdots & \vdots \\
  x_{anc, drift, k_{L_k}} & y_{anc, drift, k_{L_k}} & \phi_{anc, drift, k_{L_k}} \\
\end{pmatrix},
\]

where the coefficients of \( C \) describe the two planes. Solving for \( C \) yields

\[
C = (P^T WP)^{-1} P^T WV,
\]

where

\[
\begin{align*}
\text{Chapter 3. Spatially Dependent Underwater Navigation}
\end{align*}
\]
where $W$ is a diagonal weighting matrix whose $k^{th}$ coefficient is $\omega_k$ from (3.25). We use a coarse estimate of the location of the TN to calculate the velocity, such that

$$\hat{\nu}_k^{\text{drift}} = (x_{k-1}, y_{k-1}, \phi_{k-1}, 1) \cdot C,$$

(3.29)

and $x_{k-1}, y_{k-1},$ and $\phi_{k-1}$ are the previous course estimation of the location of the TN.

### 3.3.3 Confidence Index (CI)

Utilizing the spatial dependencies between the motions of the TN and anchor nodes has a great potential for improving UT. However, since in some cases the spatial correlation of ocean current may be weak, there is a need to limit the use of $\hat{\nu}_k^{\text{drift}}$ through a CI. Considering the likely large errors in the TN velocity estimation if spatial correlation is weak, we use two types of CIs. The first, $\gamma_r^k$, identifies anchor nodes with whom spatial dependency of motion is weak, and is incorporated in the estimation methods using the set $K_k$ from (3.20). The second, $\gamma_d^k$, is a measure for the homogeneity of the current velocity field around the TN.

#### CI for Correlation with Drift Velocity of a Single Anchor

In the above three schemes to evaluate $\hat{\nu}_k^{\text{drift}}$, only velocities of anchor nodes whose last packet was received at time instance $t_k$, $k \in K_k$, are considered for estimating $\hat{\nu}_k^{\text{drift}}$ (see (3.21), (3.23), and (3.27)). Assuming that spatial correlation of ocean current degrades with range, relative depth, and relative velocity of the anchor and TN, $\gamma_r^k$ is a function of the distance of the TN to the anchor node, the difference between the distances of the TN and the anchor to the sea bottom, $\Delta z_{\text{bottom}}$, and the measured Doppler shift, $\Delta f_k$ from (3.10). To formalize this, we use three threshold values $\text{Th}_r$, $\text{Th}_z$, and $\text{Th}_D$, and set

$$\begin{align*}
\gamma_r^k &= \begin{cases} 
1 & \text{if } |d_k| \leq \text{Th}_r \cap |\Delta z_{\text{bottom}}| \leq \text{Th}_z \cap \frac{\Delta f_k}{f_c \cos \theta_{k-1}} \leq \text{Th}_d \\
0 & \text{otherwise}
\end{cases}.
\end{align*}$$

(3.30)

We suggest determining thresholds $\text{Th}_d$ and $\text{Th}_z$ by training on a set of trajectories obtained from real data, or, if not available, simulated by a numerical ocean current model (e.g., SHYFEM [109]) for the expected bathymetry of the environment. However, a more educated guess can be made for $\text{Th}_r$. In case the bathymetry is known, we can calculate the range for which the velocity of the water current can be considered dependent. Otherwise, we use the Rossby radius of deformation, $D_r$, which is the range from a certain location at which the gravitational tide and low-related ocean current considerably change [100]. Since the effect of the gravitational force on ocean current is a dominant factor in determining the spatial correlation between the motion of the TN and the anchor, we set $\text{Th}_r = D_r$. The Rossby radius
of deformation is a function of the water depth, \( p \), earth gravity, \( g \), and the Coriolis force parameter, \( f_0 \), at the current latitude. For shallow water it is calculated as \[ D_r = \frac{\sqrt{gp}}{f_0}. \] (3.31)

Since neither \( D_r \) from (3.31) nor \( T_h d \) and \( T_h z \) are directly related to the velocity of the TN, and so are \( T_h d \) and \( T_h z \), and since only a coarse estimation of the distance \( d_k \) is needed, we consider \( \gamma_k^r \) from (3.30) to be an unbiased CI.

**CI for Detecting Uncorrelated Motion**

Our second type of CI enables self-evaluation of the estimation \( \hat{v}_k^{\text{drift}} \). Let \( \mathbf{v}_k = [v_k^x, v_k^y]^T \) be the true velocity of the TN, and \( \hat{v}_k^{\text{drift}}(1-2) \) be the first two elements of vector \( \hat{v}_k^{\text{drift}} \). To evaluate the validity of the assumption of spatially correlated ocean current for both speed and direction, we use the \( p \)-relative distance,

\[
e_v(k) = \frac{\| \mathbf{v}_k - \hat{v}_k^{\text{drift}}(1-2) \|^p}{\| \mathbf{v}_k \|^p + \| \hat{v}_k^{\text{drift}}(1-2) \|^p},
\] (3.32)

with \( p = 2 \), as the figure of merit. Error (3.32) is a normalized measure combining errors in both speed and direction, whose value is 1 if the vectors point in opposite directions and 0 if the vectors are the same.

Our aim is to obtain a CI that detects large errors (3.32). To this end, we use previous location estimates and anchors’ drift velocities and find an error prediction for a newly drift velocity information \( \mathbf{v}_k^{\text{anc,drift}} \). Since a large variability of the drift velocity field around the TN reflects poor spatial dependency of the ocean current, for such error prediction we use the variance of anchor nodes drift velocities. The weighted variance of velocities is calculated in two steps. First, the weighted mean velocity vector is calculated by

\[
\mu_k = \frac{1}{L_k} \sum_{i=1}^{L_k} \mathbf{v}_k^{\text{anc,drift}} \omega_{k_i},
\] (3.33)

where \( \omega_k \) is the weight function from (3.24) or (3.25). Then, the weighted variance vector is computed by

\[
\sigma_k^2 = \frac{1}{L_k} \sum_{i=1}^{L_k-1} \left( \mathbf{v}_k^{\text{anc,drift}} \omega_{k_i} - \mu_k \right)^2,
\] (3.34)

where the square is performed element-wise. The normalized variance is then

\[
\tilde{\sigma}_k^2 = \frac{\| \sigma_k^2 \|}{\| \mu_k^2 \|}.
\] (3.35)
Since spatial dependency of ocean current should be fixed or change slowly over time [100], we combine previous variances $\tilde{\sigma}_k^2$, $k' \in K_k$, in vectors

$$\lambda_{k'} = \left[ \tilde{\sigma}^2_{k'} \cdot \|d_k\|, \|d_k\|, \sqrt{\|d_k\|}, \tilde{\sigma}^2_{k'}, \sqrt{\tilde{\sigma}^2_{k'}} \right]^T,$$

(3.36)

where $\|d_k\|$ is used to match the contribution of the different velocities to the CI with their contribution to $\hat{v}_{k}^{\text{drift}}$. Using vectors $\lambda_{k'}$, we form a matrix of reliability indicators, $\Lambda_k = \left[ \lambda_{k_1}, \ldots, \lambda_{k_{L_k-1}} \right]^T$. Then, introducing $e_v = [e_v(k_1), \ldots, e_v(k_{L_k-1})]^T$ we calculate

$$\beta_k = \left( \Lambda \Lambda_k^T \right)^{-1} \Lambda_k e_v,$$

(3.37)

and form the error prediction

$$\hat{e}_v(L_k) = \lambda_{L_k}^T \beta_k.$$

(3.38)

Prediction $\hat{e}_v(L_k)$ is compared with a threshold $\text{Th}_e$ to form the CI

$$\gamma_{d,k} = \begin{bmatrix} 1 & \hat{e}_v(L_k) \leq \text{Th}_e \\ 0 & \text{otherwise} \end{bmatrix}.$$

(3.39)

Similarly to thresholds $\text{Th}_d$ and $\text{Th}_z$ from (3.30), we suggest calculating $\text{Th}_e$ by training on a set of modeled trajectories.

### 3.4 Results

In this section, we show and discuss the performance of our DD-UT scheme in simulations and two sea trials. We compare the following velocity estimation methods: 1) nearest neighbor (NN) from Section 3.3.2, 2) weighted superposition (WSP) from Section 3.3.2 with weighting function (3.24), 3) weighted superposition with spatial direction consideration (WSPS) from Section 3.3.2 with weighting function (3.25), 4) least squares (LS) from Section 3.3.2 by replacing $W$ from (3.28) with the identity matrix, 5) and weighted least square (WLS) from Section 3.3.2. We also compare performance of the EKF and the UKF when only INS and ToF measurements are in use and when the sound speed is considered fixed, e.g., [39]. The simulation results are also compared with the CRLB (3.18). We consider both $e^d_k$ from (3.32) and the distance of the estimated location $(\hat{x}, \hat{y})$ to the true location,

$$e^d_k = \sqrt{E \left[ (\hat{x}_k - x_k)^2 + (\hat{y}_k - y_k)^2 \right]},$$

(3.40)

for tracking performance. Note that $e^d_k$ from (3.40) can readily be compared with the square root of $J_{1,1}(k) + J_{3,3}(k)$ from (3.15).
3.4.1 Simulations

Our simulation environment includes a single TN and $L$ anchor nodes. To simulate the drift motion of nodes as well as the communication link between the anchor nodes and the TN, we have developed a MATLAB-based hybrid simulator combining numerical models for both ocean current and power attenuation in water. In this work we are mostly concerned with tracking in regions of shallow water and in time steps of a few seconds. For this reason, we have chosen the SHYFEM [109], which is designed to resolve the hydrodynamic equations for coastal seas and other small basins. To model power attenuation in communication links, we use the Bellhop ray-tracing numerical model (cf. [2]). The simulator requires a bathymetry map, latitude information, SSP, and initial location of nodes, and produces time-varying trajectories of nodes based on a generated current velocity field, as well as time-varying transmission loss values for the channel between the different nodes. For simplicity, we assume nodes are not self-propelled\(^4\). Example for the current velocity field and the resulting speed of four network nodes are shown in Figures 3.1a and 3.1b, respectively.

The structure of the simulated channel is a square area of $4 \times 4$ km\(^2\) with water depth of 100 m, and the bathymetry map includes one or two underwater hills of uniformly randomly generated width and depth of 3000 m and 2000 m, respectively. Initially, we place nodes uniformly at random in a smaller region of $1 \times 1$ km\(^2\) on the water surface and let them drift for 20 min according to the simulated current velocity field, where the TN dives at a speed of 0.05 m/sec. We use two sound speed

\(^4\)For a more complex motion pattern, equation (3.2) can be used.
profiles: one that matches model (3.4) and has a value of 1512 m/sec on the water surface, and another which has a fixed value of 1512 m/sec. We consider a carrier frequency of 15 kHz, and the model is set for location 49°16’13.33”N, 126°16’6.4”W (i.e., near Vancouver, BC). We let the anchor nodes transmit in a simple TDMA network, with a time frame of $L$ time slots of duration 10 sec. In its designated time slot, each anchor transmits a data packet including its 3-D coordinates and its drift velocity. We assume packets are received error-free as long as the signal-to-noise ratio (SNR) is above 15 dB. The SNR is calculated based on the output transmission loss from our simulator for a common power source level of 140 dB/µPa@1m, bandwidth of 1 kHz, symbol duration of 1 msec, and an ambient noise level of 50 dB/µPa/Hz.

We first investigate the performance of our velocity estimate $\hat{v}_k^{\text{drift}}$ from (3.12) and the CI $\gamma_{\text{CI}}^d$ from (3.39). Tracking performance follows next.

**Velocity Estimate (simulations)**

In Figure 3.2a, we show the effect of the number of anchor nodes on the velocity estimation. Results are averaged over 81 scenarios of different bathymetry maps and 1000 simulation runs for each scenario. For all methods, performance improves with the number of anchor nodes. As expected, for large $L$, the performance of the WLS is better than that of the LS method. This improvement confirms our assumption of range related spatial correlation of the ocean current. However, when $L \leq 5$ no improvement is observed. One reason for this is the sensitivity of the WLS method to scenarios where a cluster of anchor nodes is formed and the velocity plane can be tilted further due to noises at the measured anchors’ locations. The range related spatial correlation is also the reason for the better performance of the WSP and WSPS methods compared to that of the NN. However, the spatial direction considered in
the weighting function of the WSPS method does not seem to have a large impact on performance. We observe that for \( L < 8 \), the performances of the NN, WSP, and WSPS methods are better than those of the WLS or LS. This is because of the larger amount of information required for the latter to estimate matrix \( C \) from (3.27). In the following, we show results only for the WSP method.

In Figure 3.2b, we show the detection probability, \( P_{D} \), vs. the false alarm probability, \( P_{F} \), for the CI \( \gamma_{k}^{d} \) from (3.38). To calculate \( P_{D} \) and \( P_{F} \), we set a boundary \( e_{v}(k) = 0.1 \), such that a correct detection is declared when \( e_{v}(k) \leq 0.1 \) and \( \varepsilon_{v} \leq \Theta_{e} \), or when \( e_{v}(k) \geq 0.1 \) and \( \varepsilon_{v} \geq \Theta_{e} \). We compare results for three threshold values, \( \Theta_{e} \) (see (3.38)). Based on the results of Figure 3.2b, in the following we use \( \Theta_{e} = 0.2 \).

**Tracking (simulations)**

For tracking purposes, we follow the generated velocity field to calculate the time-varying location of the TN. To test the convergence of the tracking scheme, at the beginning of each simulation run the TN is given an initial SSV estimate, \( \hat{a}_{0} \) (see (3.3)), which includes zero-mean Gaussian noise whose covariance matrix is such that the SNR is 10 dB and the signal is regarded as the elements of the true SSV \( a_{0} \). We compare performances of our DD-UT tracking method, tracking without drift velocity estimation (No Drift), tracking without drift velocity estimation and Doppler shift measurements (No Drift+Doppler), and the latter scheme but without tracking the sound speed (No Drift+Doppler+Speed). We measure performance using the Euclidian distance \( e_{k}^{d} \) from (3.40). Since in our system measurement noise components in \( y_{k} \) from (3.13) have different units, to show effect of measurement error we set the elements of \( R_{\text{measure}} \) based on a measurement SNR defined as

\[
\text{MSNR} = \frac{E[|h(a)|_{i}^{2}]}{(\sigma_{i_{\text{measure}}}^{2})^{2}},
\]

where \( h(a) \), and \( \sigma_{i_{\text{measure}}}^{2} \) are the \( i \)th element of \( h(a) \) from (3.13) and the diagonal of \( R_{\text{measure}} \), respectively. We note that the values considered for the MSNR are taken from real INS systems and the results obtained in our sea trials (they are on the order of 50 dB, see [34] and Chapter 4).

We start by showing a tracking example for the simulated scenario presented in Figures 3.1a and 3.1b for MSNR = 40 dB. In the simulated scenario, anchor nodes 4 and 3 are located roughly 200 m and 700 m away from the TN, respectively, while anchor nodes 2 and 1 are located more than 1.5 km away. As a result, we observe similarities between the speeds of the TN and anchor nodes 4 and 3. The scenario includes two sudden changes for the speed of the TN in the \( x \) direction at time instances 760 sec and 990 sec. To show the effect of link communication errors, we simulated failures for all packets received by the TN between time instances 400 sec and 500 sec. In Figure 3.3b, we show tracking performance using the UKF as a function of time,
where we used the WSP drift velocity estimation method for our DD-UT scheme. When a change in the velocity of the TN occurs, an increase in $e^d_k$ is observed, followed by a recovery process. Since the DD-UT scheme uses more information, it converges roughly 30 sec before the other schemes. Moreover, the convergence is for a slightly lower error $e^d_k$ than that of the reference methods. Comparing the effect of adding Doppler shift information, larger errors and slower recovery is recorded when Doppler shift measurements are not in use (No Drift+Doppler) and the velocity changes. In addition, due to the change in the sound speed with depth (roughly 9 m/sec over the simulation time), a deterioration of performance of up to 5 m exists when the sound speed is considered fixed (No Drift+Doppler+Speed). As expected, errors accumulate in time when communication link failure occurs and tracking is performed solely based on INS measurements. Also here, recovery is faster using the DD-UT scheme. To comment on the sensitivity of our protocol to a mismatched model for the SSP, in Figure 3.3b we also show results (DD-UT (mismatch)) of the DD-UT scheme where the SSM is based on (3.4) but the actual SSP is fixed (which also affects the Bellhop-based simulated power attenuation). While slower recovery from velocity changes or communication link failures is observed, performance only slightly decreases and results are better than without tracking the sound speed. To emphasize this, in Figure 3.3a we show tracking performance of $\bar{c}_k$, where curves True 1 and True 2 represent the SSP used in (3.4) and the fixed one, respectively. We observe that when the SSP matches (3.4), the estimated $\bar{c}_k$ closely follows the true one. However, inaccuracies exists when Doppler shift measurements are not in use (No Drift+Doppler) and the additional information extracted from (3.9) is not available. When the SSP is fixed and the SSM is mismatched, performance decreases but the estimated sound speed is still within 1 m/sec from its true value.

Next, we show statistical tracking results. We generate 81 bathymetry maps...
Figure 3.4: Tracking performance (simulations): (a) empirical C-CDF for MSNR=50 dB, (b) \(e^d_k\) as a function of MSNR (thick line shows results for the CRLB).

and for each realization we conduct 1000 simulation runs. In each simulation, we determine the initial location of the network nodes uniformly at random, and generate new noise realizations. We note that since the scenarios generated include speed and direction changes (much like the example shown in Figure 3.1b), we cannot determine a point in time for which tracking converges. Instead, considering convergence time of up to 100 sec from the start of each simulation, we take the median of all \(e^d_k\) results obtained between time instances 100 sec and 1200 sec. In Figure 3.4b, we show the effect of the MSNR. The results indicate a relatively moderate effect of the measurement noise on performance for the MSNR range considered. Hence, as argued above, the dominant effect in tracking is the ability to accurately model the motion of the TN. As expected, performance improves the more information is used. We observe that estimating the drift velocity has more benefit than utilizing Doppler shift measurements. We note that performance of the UKF are notably better than that of the EKF, and that this performance gain is more significant for low MSNR. This is mainly due to the first order approximation of the EKF [108] and the fast changes in the TN speed and direction relative to the time step interval of the filter which are known to cause instabilities [107]. This is emphasized by the good performance of the UKF even for a mismatched SSP. The need to track the sound speed is demonstrated by the significant performance gain of the No Drift+Doppler scheme compared to that of the No Drift+Doppler+Speed. In Figure 3.4b, we also show the square root of the CRLB \(J_{1,1}(k) + J_{3,3}(k)\) from (3.15). We observe that the CRLB is well approached by our DD-UT scheme. To comment on the variations of the performance results, in Figure 3.4a we show the empirical complementary cumulative density function (C-CDF) of \(e^d_k\) for MSNR= 50 dB, as well as the CRLB for reference. We observe small variations of \(e^d_k\) for our DD-UT scheme using the UKF. The results
verify our conclusions from the average performance shown in Figure 3.4b.

3.4.2 Sea Trials

Due to the complex sea environment and our reliance on models for the SSM, sensor noise, and on the physical phenomena of spatial correlation of the ocean current, our simulation performance requires verification in a sea environment. For this reason, we have conducted two sea trials in the different bathymetry channel structures of the Mediterranean Sea and the Indian Ocean. The former was performed in August 2010 at Haifa, Israel, and included three vessels, representing three mobile nodes, which drifted for more than 2 hours. Water depth was around 40 m and the height of the sea waves was around 0.1 m. The three nodes deployed transducers at depth of 10 m, and transmitted data packets every 3 minutes using a TDMA protocol including the nodes’ 2-D UTM coordinates (which was measured at rate of 1 sec using GPS receivers). The received packets were used to calculate the ToF (applying the method described in Chapter 4), and the Doppler shift (using the method described in [111]). No acceleration measurements were taken during this sea trial, and instead we use the GPS readings to calculate $m_k^{INS}$ (see (3.5)). The time-varying location of the three nodes is shown in Figure 3.5b. The second sea trial was conducted in November 2011 at the Singapore strait with water depths of 15 m and ocean wave height of approximately 0.5 m. The experiment lasted for more than one hour and included two drifting boats. At each boat, we obtained 3-D acceleration measurements at a rate of $f = 4.8$ Hz using an Libelium Wasp Mote’s on-board accelerometer, and ToF measurements were taken at each node every 20 sec using underwater acoustic modems, manufactured by Evologics GmbH, which were deployed at a depth of 5 m. Throughout the experiment, the locations of the boats were monitored using GPS receivers at rate of 3 sec. The boats’ locations are shown in Figure 3.5a.
Chapter 3. Spatially Dependent Underwater Navigation

Table 3.1: Drift velocity estimation results from the Israel and Singapore sea trials.

<table>
<thead>
<tr>
<th>Sea Trial</th>
<th>Measure</th>
<th>NN N-I</th>
<th>WSP N-I</th>
<th>LS N-I</th>
<th>NN W-I</th>
<th>WSP W-I</th>
<th>LS W-I</th>
</tr>
</thead>
<tbody>
<tr>
<td>Israel</td>
<td>$E{</td>
<td>v_k - \hat{v}^{\text{drift}}_k (1 - 2)</td>
<td>}$</td>
<td>0.03</td>
<td>0.11</td>
<td>0.03</td>
<td>0.11</td>
</tr>
<tr>
<td></td>
<td>$E{</td>
<td>\phi_k - \hat{\phi}^{\text{drift}}_k</td>
<td>}$</td>
<td>0.16</td>
<td>0.37</td>
<td>0.09</td>
<td>0.25</td>
</tr>
<tr>
<td>Singapore</td>
<td>$E{</td>
<td>v_k - \hat{v}^{\text{drift}}_k (1 - 2)</td>
<td>}$</td>
<td>0.09</td>
<td>0.16</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>$E{</td>
<td>\phi_k - \hat{\phi}^{\text{drift}}_k</td>
<td>}$</td>
<td>0.17</td>
<td>0.22</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

For both sea trials, we use the GPS readings to calculate the drift velocity of the anchor nodes, $v^{\text{anc,drift}}_k$, and as the ground truth for the position and velocity of the TN.

**Velocity Estimate (sea trial)**

In Table 3.1, we compare the mean of errors $|v_k - \hat{v}^{\text{drift}}_k (1 - 2)|$ (in m/sec) and $|\phi_k - \hat{\phi}^{\text{drift}}_k|$ (in radians) for the NN, WSP, and LS methods described in Section 3.3. To comment on the robustness to motion instabilities, we present results when all data is considered (W-I), and when disregarding data with motion irregularities (N-I). For the Israel sea trial, motion irregularities are considered around time instances 2400 sec, 5400 sec, and 7200 sec, and for the Singapore sea trial around time instances 100 sec, 1500 sec, and 2900 sec. We note that all the above motion irregularities are detected by our CIs, $\gamma^d_k$ and $\gamma^r_k$, and both $\gamma^d_k$ and $\gamma^r_k$ are 1 in other time instances. For both sea trials, we observe good estimation results when motion is stable, and more noisy estimations but still within acceptable range when motion irregularities are considered. The results imply that the WSP method outperforms both NN and LS methods.

**Tracking (sea trial)**

In this section, we show tracking performance for the two sea trials. Since we used commercial GPS receivers whose expected accuracy was $\pm 5$ m, we consider a tracking result for which $e^d_k \leq 10$ m as accurate. In both sea trials, the depth of the transducers was fixed. Thus, we do not track the sound speed and instead use a predefined sound speed and modify the SSM (3.4) accordingly. In the Israel sea trial, we have measured a sound speed of $c = 1550$ m/sec which reduced by 2 m/sec along the water column of 40 m. In the Singapore sea trial, we rely on yearly measurements showing a fixed sound speed of 1540 m/sec. In addition, in the latter sea trial we did not have access to Doppler shift measurements, and vector $y_k$ from (3.13) is modified to not include measurement $m^\text{Doppler}_k$ from (3.10). In the following, tracking starts given an initial estimate $a_0$ (e.g., from a UWL process) such that $e^d_0 = 10$ m, and for the Israel sea
Figure 3.6: $e^d_k$ from (3.40) vs. time: (a) Singapore sea trial, (b) Israel sea trial.

We next show results for all the acquired data (i.e., with motion irregularities) and incorporate both types of CIs introduced in Section 3.3.3. Tracking results for the Singapore sea trial are shown in Figure 3.6a. For all methods, convergence is reached after roughly 200 sec (i.e., using 10 received packets from the anchor). We observe several cases where performance degrades (e.g., around time instances 500 sec, 2000 sec, and after 2800 sec). As the effect is similar to the one observed in Figure 3.3b, these performance fluctuations are explained by both ToF inaccuracies (due to the shallow water environment) and velocity changes (as seen in Figure 3.5a). Similar to the results in Figure 3.4b, better performance is obtained by the UKF compared to the EKF. As seen in the figure between time instances 2000 and 2500, the No Drift scheme sometimes achieves better results than our DD-UT scheme. However, most of the time significant performance gain of up to 5 m is obtained by the DD-UT scheme compared to the No Drift one.

In Figure 3.6b, we show tracking performance for the Israel sea trial. For all methods, a sudden spike in tracking error occurs at around 2400 sec. This error is due to the loss of communication in the network that occurred between time instances 2400 sec and 3200 sec. The similar decrease in performance at around 4000 sec is explained by a sudden change of velocity. Both faster recovery and absolute performance improvement are observed when drift velocity estimation is used. Moreover, performance significantly improves when Doppler shift measurements are utilized. Comparing the three methods of velocity estimation, similar to our simulation results, best performance is obtained by the weighted superposition scheme for which the tracking error converges to less than 5 m.

\footnote{We note that the WLS method yielded almost the same results as the LS method}
3.5 Conclusions

In this chapter, we considered the problem of UT, where nodes experience irregular drifting motion. Assuming spatial correlation of the ocean current, we suggested three methods to estimate the drift velocity of the TN by superimposing drift velocities reported by the anchor nodes. This augments the amount of information available for the TN as an unbiased velocity estimate. To combat errors due to weakly correlated motion of nodes, we proposed two types of unbiased confidence indexes. Our scheme also incorporates sound-speed tracking and Doppler shift measurements available from received packets, which, due to the uncertainty of the sound speed and the continuous motion of nodes in the channel, is a challenging task. We implemented tracking using the UKF and the EKF, and evaluated performance using numerical models for ocean current and power attenuation in the ocean. The results indicate a large gain of our scheme compared to reference approaches which do not consider drift velocity estimation, Doppler shift measurements, or sound speed uncertainties, and showed that the CRLB is well approached by our scheme. The results from two sea trials for different bathymetry channel structure conducted in Israel and in Singapore support the findings from simulations.
Chapter 4

LOS and NLOS Classification for UWL

In the previous chapters, we have presented our approach for UWL and UT. As expected and observed in Figures 2.6 and 3.4b, localization accuracy depends on the measurement noise from internal and external sources. In this chapter, we focus on ranging noise, and specifically on the problem of mistaking NLOS-related ToF measurements as LOS measurements. Several works suggested ways to avoid misidentification of NLOS and LOS links. In [114], direct sequence spread spectrum (DSSS) signals, which have narrow auto-correlation, are transmitted to allow better separation of paths in the estimated channel response. Averaging ToF measurements from different signals is suggested in [115], where results show considerable reduction in measurement errors. In [15], NLOS-related noise in the UAC is modeled using the Ultra Wideband Saleh-Valenzuela (UWB-SV) model, and multipath noise is mitigated using the approach introduced in [116]. A way to identify NLOS measurements is presented in [95], where assuming that NLOS-based measurements have larger variance than LOS-based measurements, measurements which increase the global variance are rejected. In [117], localization accuracy is improved by selecting ToF measurements based on minimal statistical mode (i.e., minimal variance and mean). Alternatively, the authors in [118] suggested a method for reducing the effect of NLOS-based noise by assigning each measurement with a weight inversely proportional to the difference between the measured and expected distances from previous localization. However, no systematic way has been suggested to classify NLOS- and LOS-related ToF measurements. Instead, we take a more rigorous approach and estimate the distribution of the ToF measurements.

The intuition behind our approach is that the continuous motion of nodes in the UAC changes causes shift in the arrival times and energy of signals received through different propagation paths, causing link variations. This diversity can be exploited by obtaining multiple propagation delay (PD) measurements and classifying them into classes of LOS, SNLOS and ONLOS (referring to Figure 1.4, recall SNLOS refer to sea surface and bottom reflections and ONLOS to reflection off an object). In doing so, we also take into account the effect of mobility on distance. The proposed classification can improve the accuracy of UWL by rejecting or correcting NLOS-related PD measurements to obtain a single ToF estimation, or by using them to bound range estimation.

To implement our scheme we present a two-step classification algorithm. We
first identify ONLOS-related PD-measurements by comparing PD-based range estimations with range estimations obtained from received-signal-strength (RSS) measurements. Considering the difficulty in acquiring an accurate attenuation model, our algorithm requires only a lower bound for RSS-based distance estimations. After excluding PD measurements related to ONLOS, we apply a constrained expectation-maximization (EM) algorithm (cf. [91]) to further classify the remaining PD measurements into LOS and SNLOS. Through a clustering of PD measurements we mitigate changes in propagation delay due to mobility of nodes. The EM algorithm also estimates the statistical parameters of both classes, which can be used to improve the accuracy of UWL. Results from extensive simulations and three sea trial experiments in different areas of the world demonstrate the efficacy of our approach through achieving a high detection rate for ONLOS links and good classification of non-ONLOS related PD measurements into LOS and SNLOS. To the best of our knowledge, no prior work considered a machine learning approach for NLOS and LOS classification of multiple PD measurements. Moreover, link classification for ranging was not investigated for the special characteristics of the UAC.

It should be noted that while our approach can also be adapted to other types of fading channels, it is particularly suited for UWL for the following reasons. First, our algorithm relies on significant power absorption due to reflection loss in ONLOS links, which are typical in the underwater environment. Second, we assume that the difference in propagation delay between signals traveling through SNLOS and LOS links is noticeable, which is acceptable in the UAC due to the low sound speed in water (approximately 1500 m/sec). Third, our algorithm is particularly beneficial in cases where NLOS paths are often mistaken for the LOS path, which occurs in UWL, where the LOS path is frequently either not the strongest or non-existent. Last, we assume that the variance of PD measurements originating from SNLOS links is greater than that of measurements originating from LOS, which fits channels with long delay spread such as the UAC.

The remainder of this chapter is organized as follows. Our system model and assumptions are introduced in Section 4.1. In Section 4.2, we present our approach to identify ONLOS links. Next, in Section 4.3, we formalize the EM algorithm to classify non-ONLOS related PD measurements into LOS and SNLOS. Section 4.4 includes performance results of our two-step algorithm obtained from synthetic UAC environments (Section 4.4.1) and from three different sea trials (Section 4.4.2). Finally, conclusions are offered in Section 4.5.

4.1 System Setup and Assumptions

Referring to Figure 1.4, our system comprises of one or more transmitter-receiver pairs, \((u, a_j)\), exchanging a single communication packet of \(N\) symbols or impulse signals, from which a vector \(\mathbf{X} = [x_1, \ldots, x_N]\) of PD measurements \(x_i\), and corre-
sponding measured time $t_i$, is obtained using detectors such as in, e.g., [44, 46, 47]. We model $x_i$ such that

$$x_i = x_{\text{LOS}} + n_i,$$  \hspace{1cm} (4.1)

where $x_{\text{LOS}}$ is the PD in the LOS link, and $n_i$ is the measurement noise. We note that due to the mobility of nodes, $x_{\text{LOS}}$ is not strictly constant during the measurement interval, and this motion bounds the accuracy of ranging.

We focus on transmission over short range (on the order of a few km), for which refraction of acoustic waves is negligible and propagation delay in the LOS link is on average shorter than in the NLOS links. We assume successively transmitted signals for PD measurements are separated by guard intervals such that $n_i$ in (4.1) can be assumed i.i.d. For each measurement $x_i$, a PD-based estimate, $d_{\text{PD}}^i$, is obtained by multiplying $x_i$ with an assumed propagation speed, $c$. In addition, based on an attenuation model for an LOS link, we obtain RSS-based range estimates, $d_{\text{RSS}}^i$, $i = 1, \ldots, N$, from the received signals.

In the following, we introduce our system model for obtaining RSS-based range measurements as well as the assumed PDF for PD measurements.

### 4.1.1 RSS-Based Range Measurements

Let $d_{\text{LOS}}$ denote the distance corresponding to $x_{\text{LOS}}$, i.e., $d_{\text{LOS}} = x_{\text{LOS}}c$. For the purpose of obtaining RSS-based range measurements, we use the popular model [5]

$$TL_{\text{LOS}}(d_{\text{LOS}}) = PL(d_{\text{LOS}}) + AL(d_{\text{LOS}}) + \epsilon,$$  \hspace{1cm} (4.2)

where $PL(d_{\text{LOS}}) = \gamma \log_{10}(d_{\text{LOS}})$ is the propagation loss, $AL(d_{\text{LOS}}) = \alpha \frac{d_{\text{LOS}}}{1000}$ is the absorption loss, $\gamma$ and $\alpha$ are the propagation and absorption coefficients, respectively, and $\epsilon$ is the model noise assumed to be Gaussian distributed with zero mean and variance $\phi$. Considering the simplicity of the model in (4.2), we do not directly estimate $d_{\text{RSS}}^i$ but rather estimate a lower bound $d_{\text{RSS, LB}}^i$, for which we apply upper bounds for $\gamma$ and $\alpha$ in (4.2) according to the expected underwater environment.

For an ONLOS link with distance $d_{\text{ONLOS}} = d_{\text{ONLOS,1}} + d_{\text{ONLOS,2}}$, where $d_{\text{ONLOS,1}}$ and $d_{\text{ONLOS,2}}$ are the distance from source to reflector and from reflector to receiver, respectively\(^6\), we assume that the power attenuation in logarithmic scale is given by [5]

$$TL_{\text{ONLOS}}(d_{\text{ONLOS}}) = TL_{\text{LOS}}(d_{\text{ONLOS,1}}) + TL_{\text{LOS}}(d_{\text{ONLOS,2}}) + RL,$$  \hspace{1cm} (4.3)

where RL is the reflection loss of the reflecting object, whose value depends on the material and structure of the object and the carrier frequency of the transmitted signals. Since RL is often large (see examples in [5]), and due to the differences between models (4.2) and (4.3), we further assume that

$$TL_{\text{ONLOS}}(d_{\text{ONLOS}}) \gg TL_{\text{LOS}}(d_{\text{ONLOS}}).$$  \hspace{1cm} (4.4)

\(^6\)Referring to the ONLOS link between node pair $(u, a_2)$ in Figure 1.4, $d_{\text{ONLOS,1}} = d_{21}$ and $d_{\text{ONLOS,2}} = d_{22}$. 

69
4.1.2 PDF for PD Measurements

Since we assume changes in $x_{\text{LOS}}$ are bounded by a small transmitter-receiver motion during the time $X$ is obtained, we can model the PDF of the noisy measurement $x_i$ as a mixture of $M = 3$ distributions, corresponding to LOS, SNLOS, and ONLOS links, such that (assuming independent measurement noise samples in (4.1))

$$p(X|\theta) = \prod_{x_i \in X} \sum_{m=1}^{M} k_m p(x_i|\omega_m),$$

where $\theta = [\omega_1, k_1, \ldots, \omega_M, k_M]$, $\omega_m$ are the parameters of the $m$th distribution, and $k_m \left( \sum_{m=1}^{M} k_m = 1 \right)$ is the a-priori probability of the $m$th distribution. Clearly, $p(X|\theta)$ depends on multipath and ambient noise in the UAC, as well as on the detector used to estimate $x_i$. While recent works used the Gaussian distribution for $p(x_i|\omega_m)$ (cf., [86] and [10]), since multipath and ambient noises are hard to model in the UAC, we take a more general approach and model it according to the generalized Gaussian PDF [119], such that

$$p(x_i|\omega_m) = \frac{\beta_m}{2\sigma_m \Gamma\left(\frac{1}{\beta_m}\right)} e^{-\left(\frac{|x_i - v_m|}{\sigma_m}\right)^{\beta_m}}$$

with parameters $\omega_m = [\beta_m, v_m, \sigma_m]$. We associate the parameter vectors $\omega_1$, $\omega_2$, and $\omega_3$ with distributions corresponding to the LOS, SNLOS, and ONLOS links, respectively. Thus, by (4.1), $v_1 = x_{\text{LOS}}$. Without prior knowledge of the actual distribution of PD in the LOS and NLOS links, the use of parameter $\beta_m$ in (4.6) gives our model a desired flexibility, with $\beta_m = 1$, $\beta_m = 2$, and $\beta_m \to \infty$ corresponding to Laplace, Gaussian, and uniform distribution, respectively. The flexibility and fit of model (4.6) is demonstrated using sea trial results in Section 4.4.2.

Following [95] and [117], we assume that PD measurements of NLOS links increase the variance of the elements of $X$. Thus, if $\varsigma_1$, $\varsigma_2$, and $\varsigma_3$ are the respective variances of measurements related to the LOS, SNLOS, and ONLOS links, then we have

$$\varsigma_1 < \varsigma_m, \ m = 2, 3.$$  

Since, for the PDF (4.6),

$$\varsigma_m = (\sigma_m)^2 \frac{\Gamma\left(\frac{3}{\beta_m}\right)}{\Gamma\left(\frac{1}{\beta_m}\right)},$$

and by (4.8), $\varsigma_m$ does not change much with $\beta_m$, constraint (4.7) can be modified to

$$\sigma_1 < \sigma_m, \ m = 2, 3.$$
Furthermore, let \( T_{\text{LIR}} \) be the assumed length of the UAC impulse response, which is an upper bound on the time difference between the arrivals of the last and first paths. Then, since \( \varsigma_1, \varsigma_2, \) and \( \varsigma_3 \) in (4.8) capture the spread of measurements related to the LOS, SNLOS, and ONLOS links respectively,

\[
\sqrt{\varsigma_m} < T_{\text{LIR}}, \quad m = 1, 2, 3. \tag{4.10}
\]

Moreover, the propagation delay through the LOS link is almost always shorter than those for any NLOS link\(^7\). Hence, we have

\[
v_1 < v_m < v_1 + T_{\text{LIR}}, \quad m = 2, 3. \tag{4.11}
\]

Together with assumption (4.7), assumption (4.11) manifests the expected differences between the PD in the LOS and NLOS links. Clearly, the more separable PD measurements from LOS and NLOS links are (i.e., the propagation delay difference is larger), the better the classification will be. Since the channel impulse response is longer for deeper channels, classification accuracy is expected to improve with depth.

### 4.1.3 Remark on Algorithm Structure

We offer a two-step approach to classify PD measurements into LOS, SNLOS, and ONLOS. First, assuming large attenuation in an ONLOS link, we compare PD-based and RSS-based range estimates to differentiate between ONLOS and non-ONLOS links. Then, assuming PDF (4.6) for PD measurements, we further classify non-ONLOS links into LOS and SNLOS links.

The reason for separating classification of ONLOS and SNLOS links is insufficient information about the distribution of the two link types. For example, delay in ONLOS links may be similar to or different from that of SNLOS links. In the former case, classification should be made for \( M = 2 \) states, while for the latter three states are required. Since a mismatch in determining the number of states may lead to improper classification, we rely on the expected high transmission loss in ONLOS links to first identify these links. Furthermore, a separate identification of ONLOS links can be used as a backup to our LOS/NLOS classifier. That is, we can still identify the link as ONLOS even when the channel is fixed, and thus PD measurements originate from a single link-type. In the following sections, we describe our two-step approach for classifying PD measurements.

### 4.2 Step One: Identifying ONLOS Links

Considering (4.4), we identify whether measurement \( x_i \in X \) is ONLOS-related based on three basic steps as follows:

---

\(^7\)We note that in some UWACs, a signal can propagate through a soft ocean bottom, in which case SNLOS signals may arrive before the LOS signal [5]. However, such scenarios are rare and we assume that on average relation (4.10) holds.
• **Estimation of** $d_i^{PD}$

We first obtain the PD-based range estimation as $d_i^{PD} = c \cdot x_i$.

• **Estimation of** $d_i^{RSS, LB}$

Next, assuming knowledge of the transmitted power level, we measure the RSS for the $i$th received signal/symbol, and estimate $d_i^{RSS, LB}$ based on (4.2), replacing $\gamma$ and $\alpha$ with upper bounds $\gamma_{UB}$ and $\alpha_{UB}$, respectively.

• **Thresholding**

Finally, we compare $d_i^{PD}$ with $d_i^{RSS, LB}$. If $d_i^{RSS, LB} > d_i^{PD}$, then $x_i$ is classified as ONLOS. Otherwise, it is determined as non-ONLOS.

Note that the use of the above step Thresholding relies on assumption (4.4). To clarify this, consider an ONLOS link. The RSS-based range estimation, $d_i^{RSS, LB}$, is obtained from an upper bound for an attenuation model (4.2), i.e., from applying an attenuation model for an LOS link to an ONLOS link. Since the latter is expected to have a much larger power attenuation than the used model, it follows that $d_i^{RSS, LB}$ would be much larger than $d_i^{PD}$. Similarly, consider a non-ONLOS link (i.e., LOS or SNLOS). Here, since we use an upper bound for the attenuation model, we expect $d_i^{RSS, LB}$ to be smaller than $d_i^{PD}$.

Next, we analyze the expected performance of the above ONLOS link identification algorithm in terms of (i) detection probability of non-ONLOS links, $P_{rd, non-ONLOS}$, and (ii) detection probability of ONLOS links, $P_{rd, ONLOS}$. To this end, since explicit expression for $d_{LOS}$ cannot be obtained from (4.2), in the following, we use the upper bound $d_i^{RSS, LB}$ such that

$$\log_{10}(d_i^{RSS, LB}) = \frac{TL}{\gamma_{UB}}.$$  \hspace{1cm} (4.12)

We note that (4.12) is a tight bound when the carrier frequency is low or when the transmission distance is small.

### 4.2.1 Classification of non-ONLOS Links

For non-ONLOS links, we expect $d_i^{RSS, LB} \leq d_{LOS}$. Thus, since by bound (4.12), $P_r(d_i^{RSS, LB} \leq d_{LOS}) \geq P_r(d_i^{RSS, LB} \leq d_{LOS})$, and substituting (4.2) in (4.12), we get

$$P_{rd, non-ONLOS} \geq 1 - Q \left( \frac{(\gamma_{UB} - \gamma) \log_{10}(d_{LOS}) - \alpha_{d_{LOS}}}{\phi} \right),$$ \hspace{1cm} (4.13)

where $Q(x)$ is the Gaussian Q-function.
4.2.2 Classification of ONLOS Links

When the link is ONLOS, we expect $d_i^{RSS, LB} \geq d_{ONLOS}$. Then, substituting (4.3) in (4.12), and since $P_r(d_i^{RSS, LB} \geq d_{ONLOS}) \leq P_r(d_i^{RSS, LB} \geq d_{ONLOS})$, we get

$$P_{rd,ONLOS} \leq Q\left(\frac{\gamma_{UB} \log_{10} (d_{ONLOS}) - \gamma \log_{10} (d_{ONLOS,1}d_{ONLOS,2}) - \alpha d_{ONLOS}}{RL} \right).$$  

(4.14)

Next, we continue with classifying non-ONLOS links into LOS and SNLOS links.

4.3 Step 2: Classifying LOS and SNLOS Links

After excluding ONLOS-related PD measurements in Step 1, the remaining elements of $X$, organized in the pruned vector $X^{ex}$, are further classified into LOS ($m = 1$) and SNLOS ($m = 2$) links and their statistical distribution parameters, $\omega_m$, are estimated. Before getting into the details of our LOS/SNLOS classifier, we first explain its basic idea.

4.3.1 Basic Idea

The underlying idea of our approach is to utilize the expected variation in link type of PD measurements due to mobility of nodes at sea. After identifying ONLOS links, this variation means that our set includes PD measurements of different values and link types. This allows us to use a machine learning approach to classify the measurements into two classes, LOS and NLOS. For this purpose, we use the EM algorithm. While using EM to classify measurement samples into distinct distributions is a common approach, here the distribution parameters in (4.6) should also satisfy constraints (4.10), (4.9), and (4.11), where the two latter constraints introduce dependencies between the parameters of the LOS and NLOS classes. Furthermore, we incorporate equivalence constraints to group measurements of similar values into clusters which elements are classed to the same link type, thereby mitigating shifts in the value of $x_{LOS}$ due to nodes’ mobility. As we show later on, this result in a non-convex maximization of the log-likelihood function. For this reason, we present a heuristic sub-optimal algorithm.

In the following, we start by formalizing the equivalence constraints, and formulating the log-likelihood function. Next, we formulate a constrained optimization problem to estimate the distribution parameters, and present our heuristic approach to efficiently solve it. Given this estimate, we calculate the posterior probability of each PD measurement belonging to the LOS and SNLOS class, and classify the elements of $X^{ex}$ accordingly. Finally, we describe how the initial solution, required for the EM algorithm, is obtained and calculate the hybrid Cramér-Rao bound to bound the performance of our classifier.
4.3.2 Equivalence Constraints

In setting equivalence constraints, we assume that the identity and delay of the dominant channel path, used for PD detection, is constant within a given coherence time, $T_c$, and that for a bandwidth $B$ of the transmitted signal, system resolution is limited by $\Delta T = \frac{1}{B}$. PD measurements satisfying equivalence constraints are collected into vectors $\Lambda_l, l = 1, \ldots, L$, where $L$ denotes the number of such equivalence sets. Each PD measurement is assigned to exactly one vector, i.e., $\Lambda_l$ have distinct elements. To formalize this, we determine $x_i$ (recall that measurement $x_i$ corresponds to measurement time $t_i$) and $x_j$ being equivalent, denoted as $x_i \Leftrightarrow x_j$, if

$$
|t_i - t_j| \leq T_c \quad (4.15a)
$$

$$
|x_i - x_j| \leq \Delta T. \quad (4.15b)
$$

To illustrate this, let $x_i$, $x_j$, and $x_n$ correspond to the same class (either LOS or NLOS), such that $x_i \Leftrightarrow x_j$ and $x_j \Leftrightarrow x_n$. Then, although due to motion of nodes, $x_i$ and $x_n$ may not satisfy condition (4.15b), since $\Lambda_l, l = 1, \ldots, L$, are distinct vectors, all three measurements $x_i, x_j,$ and $x_n$ are grouped into same vector $\Lambda_l$ and are further classified to the same state. That is, vectors $\Lambda_p$ and $\Lambda_q$ are merged if they have a common element. To form vectors $\Lambda_l, l = 1, \ldots, L$, we begin with $|X_{ex}|$ (where $|\Lambda|$ symbolizes the number of elements in vector $\Lambda$) initial vectors of single PD measurements, and iteratively merge vectors. This process continues until no two vectors can be merged. As a result, we reduce the problem of classifying $x_i \in X_{ex}$ into classifying $\Lambda_l$, which account for resolution limitations and node drifting.

4.3.3 Formalizing the Log-Likelihood Function

Let the random variable $\lambda_l$ be the classifier of $\Lambda_l$, such that if $\Lambda_l$ is associated with class $m, m \in \{1, 2\}$, then $\lambda_l = m$. Also let $\lambda = [\lambda_1, \ldots, \lambda_L]$. Since elements in $X_{ex}$ are assumed independent,

$$
P_l(\lambda_l = m| \Lambda_l, \theta^p) = \frac{k_m^p p(\Lambda_l| \omega_m^p)}{p(\Lambda_l| \theta^p)} = \frac{k_m^p \prod_{x_i \in \Lambda_l} p(x_i| \omega_m^p)}{\sum_{j=1}^2 k_j^p \prod_{x_i \in \Lambda_l} p(x_i| \omega_j^p)}. \quad (4.16)
$$

Then, we can write the expectation of the log-likelihood function with respect to the conditional distribution of $\lambda$ given $X_{ex}$ and the current estimate $\theta^p$ as

$$
L(\theta|\theta^p) = E \left[ \ln \left( P_l(X_{ex}, \lambda|\theta) \right) | X_{ex}, \theta^p \right] = \\
\sum_{m=1}^2 \left[ \sum_{l=1}^L P_l(\lambda_l = m| \Lambda_l, \theta^p) \sum_{x_i \in \Lambda_l} \ln p(x_i| \omega_m) + \sum_{l=1}^L P_l(\lambda_l = m| \Lambda_l, \theta^p) \ln k_m^p \right]. \quad (4.17)
$$

where $\ln x = \log_e x$ is the natural logarithmic function.
Chapter 4. LOS and NLOS Classification for UWL

Assuming knowledge of \( \theta^p \), \( \theta^{p+1} \) is estimated as the vector of distribution parameters that maximizes (4.17) while satisfying constraints (4.9), (4.10) and (4.11). This procedure is repeated for \( P_{\text{last}} \) iterations, and the convergence of (4.17) to a local maximum is proven [91]. Then, we calculate \( P_t(\lambda_t = m|\Lambda_t, \theta^{p_{\text{last}}}) \) using (4.16), and associate vector \( \Lambda_t \) with the LOS path if

\[
P_t(\lambda_t = 1|\Lambda_t, \theta^{p_{\text{last}}}) > P_t(\lambda_t = 2|\Lambda_t, \theta^{p_{\text{last}}}),
\]

or with an SNLOS path otherwise. Estimation \( \theta^{p_{\text{last}}} \) and classifications \( \lambda_t \) could be used further to improve the accuracy of UWL, e.g., [95, 23]. We observe that the two terms on the right-hand side of (4.17) can be separately maximized, i.e., given \( \theta^p \), we can obtain \( \omega_m^{p+1} \) from maximizing the first term, and \( k_m^{p+1} \) from maximizing the second term. Thus (see details in [91]),

\[
k_m^{p+1} = \frac{1}{L} \sum_{l=1}^{L} P_t(\lambda_t = m|\Lambda_t, \theta^p), \quad m = 1, 2.
\]

In the following, we describe the details of our classification procedure for the estimation of \( \omega_m \), followed by a heuristic approach for the initial estimates \( \theta^0 \).

### 4.3.4 Estimating the Distribution Parameters \( \omega_1 \) and \( \omega_2 \)

To estimate \( \omega_m \), we consider only the first term on the right-hand side of (4.17), which for the PDF (4.6) is given by

\[
f(v_m, \sigma_m, \beta_m) = \sum_{l=1}^{L} P_t(\lambda_t = m|\Lambda_t, \theta^p) \sum_{x_i \in \Lambda_t} \ln \beta_m - \ln(2\sigma_m) - \ln \Gamma\left(\frac{1}{\beta_m}\right) - \left(\frac{|x_i - v_m|}{\sigma_m}\right)^{\beta_m}.
\]

Then, considering constraints (4.9), (4.10) and (4.11), we find \( \omega_m^{p+1} \) by solving the following optimization problem:

\[
\omega_1^{p+1}, \omega_2^{p+1} = \arg\min_{\omega_1, \omega_2} \sum_{m=1}^{2} f(v_m, \sigma_m, \beta_m) \quad \text{s.t.} \quad v_1 \leq v_2 \leq v_1 + T_{\text{LIR}} \quad \sigma_m \left[\frac{\Gamma\left(\frac{3}{\beta_m}\right)}{\Gamma\left(\frac{1}{\beta_m}\right)} - T_{\text{LIR}}\right] \leq 0, \quad m = 1, 2.
\]

We observe that convexity of \( f(v_m, \sigma_m, \beta_m) \) depends on \( \beta_m \). In Appendix A, we present an alternating optimization approach (cf. [120]) to efficiently solve (4.21).

Next, we present an algorithm to obtain the initial estimation, \( \theta^0 \), whose accuracy affects the above refinement as well as the convergence rate of the EM algorithm.
4.3.5 Forming Initial Estimation $\theta^0$

Our algorithm to estimate $\theta^0$ is based on identifying a single group, $\Lambda^*_l$, whose elements belong to the LOS class with high probability, i.e., $P_r(\Lambda^*_l = 1) \approx 1$. This group is then used as a starting point for the K-means clustering algorithm [91], resulting in an initial classification $\lambda_l$ for $\Lambda_l$, $l = 1, \ldots, L$, to form two classified sets $X^e_m$, $m = 1, 2$. Finally, we evaluate the mean, variance, and kurtosis of the elements in vector $X^e_m$, denoted as $E[X^e_m], \text{Var}[X^e_m]$, and $K[X^e_m]$, respectively, to estimate $\theta^0$ using the following properties for distribution (4.6):

$$\frac{|X^e_m|}{|X^e|} = k_m, E[X^e_m] = \nu_m, \text{Var}[X^e_m] = \frac{\sigma^2_m \Gamma \left( \frac{3}{\beta_m} \right)}{\Gamma \left( \frac{1}{\beta_m} \right)} , K[X^e_m] = \frac{\Gamma \left( \frac{5}{\beta_m} \right) \Gamma \left( \frac{1}{\beta_m} \right)}{\Gamma \left( \frac{3}{\beta_m} \right) \frac{2}{2}} - 3.$$  

(4.22)

Since we assume that $\sigma_1 < \sigma_2$ (see (4.9)), we expect small differences between measurements of the LOS link, compared to those of SNLOS links. We use this attribute to identify group $\Lambda^*_l$ by filtering $X^e$ and calculating the first derivative of the sorted filtered elements. Group $\Lambda^*_l$ corresponds to the smallest filtered derivative.

4.3.6 Discussion

We note that the constraints in (4.21) do not set bounds on the values of $\omega_1$ and $\omega_2$, but rather determine the dependencies between them. This is because, apart from the value of $T_{LIR}$ and distribution (4.6), we do not assume a-priori knowledge about the values of $k_m$ and $\omega_m$, $m = 1, 2$. Without node motion, all elements of $X^e$ belong to one class, but our classifier might still estimate both $k_1$ and $k_2$ to be non-zero, resulting in wrong classification into two classes. In this case, using the average of the elements of $X^e$ might give a better estimation of $d_{LOS}$ than $\nu_1$.

To limit this shortcoming of our classifier, we assume that $\nu_1$ and $\nu_2$ are distinct if $X^e$ is indeed a mixture of two distributions. To this end, in the last iteration, $P_{\text{last}}$, we classify $X^e$ as a single class (of unknown type) if the difference $|\nu_1^{\text{last}} - \nu_2^{\text{last}}|$ is smaller than a threshold value, $\Delta \nu$ (determined by the system resolution for distinct paths). Then, if required, we find the distribution parameters of the (single) class by solving a relaxed version of (4.21), setting $k_1 = 1$ and $k_2 = 0$. Nevertheless, we motivate relevance of our classifier in Section 4.4.2 by showing that scenarios in which $X^e$ is indeed a mixture of two distributions are not rare in real sea environments.

4.3.7 Summarizing the Operation of the Classifier

We now summarize the operation of our classification algorithm, whose pseudocode is presented in Algorithm 2. First, we evaluate $d_i^{PD}$ and $d_i^{RSS, LB}$ (lines 1-2). If $d_i^{RSS, LB} > d_i^{PD}$, we classify $x_i$ as ONLOS; otherwise, we classify it as non-ONLOS (lines 3-5) and form the vector of non-ONLOS PD measurements, $X^e$, and groups
Algorithm 2 Classifying $X$

1: $d^\text{PD}_i := c \cdot x_i$
2: Calculate $d^\text{RSS,LB}_i$ using RSS measurements, $\gamma_{\text{UB}}, \alpha_{\text{UB}}$ and model (4.2)
3: **if** $d^\text{RSS,LB}_i > d^\text{PD}_i$ **then**
4: Classify $x_i$ as ONLOS link
5: **else**
6: Exclude ONLOS measurements to form vector $X^\text{ex}$ and groups $\Lambda_l$ satisfying (4.15)
7: Estimate $\theta^0$
8: **for** $p := 2$ to $P_{\text{last}}$ **do**
9: Calculate $k^p_m$, $m = 1, 2$ using (4.16) and (4.19)
10: **for** $i := 1$ to $N_{\text{repeat}}$ **do** {alternating maximization to solve (4.21) (see Appendix A)}
11: Estimate $\omega^p_{m,i}$, $m = 1, 2$ and set $v^p_{m,i+1} := v^p_{m,i}$, $\sigma^p_{m,i+1} := \sigma^p_{m,i}$, $\beta^p_{m,i+1} := \beta^p_{m,i}$
12: **end for**
13: $m = 1, 2$: $v^p_{m} := v^p_{m,N_{\text{repeat}}}$, $\sigma^p_{m} := \sigma^p_{m,N_{\text{repeat}}}$, $\beta^p_{m} := \beta^p_{m,N_{\text{repeat}}}$
14: **end for**
15: **if** $|v^{P_{\text{last}}} - v^{P_{\text{last}}}| > \Delta v$ **then**
16: Calculate $P_l(\lambda_l = m|\Lambda_l, \theta^{P_{\text{last}}})$ and $\lambda_l$, $m = 1, 2$, $l = 1, \ldots, L$ using (4.16), (4.18)
17: **else**
18: Vector $X^\text{ex}$ consists of a single class. Repeat steps 7-14 for $k_1 = 1$, $k_2 = 0$
19: **end if**
20: **end if**

$\Lambda_l$, $l = 1, \ldots, L$ (line 7). Next, we form the initial solution, $\theta^0_m$ (line 7), and run the EM algorithm for $P_{\text{last}}$ iterations (lines 8-14). The procedure starts with estimating $k^p_m$ (line 9), followed by an iterative procedure to estimate $\omega^p_m$ for a pre-defined number of repetitions $N_{\text{repeat}}$ (lines 10-13). After iteration $P_{\text{last}}$, we check if vector $X^\text{ex}$ consists of two classes (line 15), and determine classifiers $\lambda_l$, $l = 1, \ldots, L$ (line 16); otherwise $X^\text{ex}$ is classified as a single class (of unknown type), and, if estimating $\omega_m$ is required, we repeat the above procedure while setting $k_1 = 1$, $k_2 = 0$ (line 18). The software implementation of the above algorithm can be downloaded from [96].

Since the data for step 2 in Algorithm 2 is already available through the detection of each symbol, and since the complexity of the EM algorithm is $O(NP_{\text{last}})$ (cf. [91]), the complexity of our algorithm is $O(NP_{\text{last}}N_{\text{repeat}})$. We also note that the EM algorithm, as well as the alternate maximization process described in Appendix A, provably converge to a local maximum of the log-likelihood function (4.17). In the following, we provide the hybrid Cramér-Rao lower bound (HCRLB) as a benchmark for our classifier.
4.3.8 Deriving the HCRLB

Consider the vector of measurements $X^{ex}$ whose elements are drawn from distributions (4.6) with $M = 2$ classes (we assume that ONLOS measurements have correctly been identified). Our classifier estimates the vector $\theta = [\nu_1, \sigma_1, \beta_1, k_1, \nu_2, \sigma_2, \beta_2] = [\theta_1, \ldots, \theta_7]$. We observe that constraints (4.11), (4.9), and (4.10), introduce dependencies between pairs $(\theta_1, \theta_5)$, $(\theta_2, \theta_6)$, $(\theta_r, \theta_{r+1})$, $r = 2, 6$, respectively. Thus, we cannot use the conventional Cramér-Rao Bound to lower bound the variance of any unbiased estimator of $\theta$. Instead, we apply the HCRLB considering $\theta_1$ as a deterministic and $\theta_r = [\theta_2, \ldots, \theta_7]$ a vector of random variables having prior distributions, respectively. The HCRLB is given by [121]

$$E_{X^{ex}, \theta_r | \theta_1} \left[ (\theta^{P_{last}} - \theta) (\theta^{P_{last}} - \theta)^T \right] \geq H^{-1}(\theta_1), \quad (4.23)$$

where $H(\theta_1) \in \mathbb{R}^{7 \times 7}$ is the hybrid Fisher information matrix$^8$ (HFIM). Let $g_i$ be the classifier of $x_i$ (i.e., $g_i = \lambda_i$ if $x_i \in \Lambda_i$). Then, the $(j, q)$th element of the HFIM is

$$H(\theta_1)_{j,q} = E_{\theta_r | \theta_1} [F(\theta_r, \theta_1)_{j,q}] + E_{\theta_r | \theta_1} \left[ -\frac{\partial^2}{\partial \theta_j \partial \theta_q} \log p(\theta_r | \theta_1) \right], \quad (4.24)$$

where

$$F(\theta_r, \theta_1)_{j,q} = E_{X^{ex} | \theta_r, \theta_1} \left[ -\sum_{i=1}^{X^{ex}} \frac{\partial^2}{\partial \theta_j \partial \theta_q} \log k_i p(x_i | \omega_{\theta_1}) \right]. \quad (4.25)$$

Solving (4.24) requires the calculation of

$$p(\theta_r | \theta_1) = p(k_1)p(\nu_2 | \nu_1)p(\sigma_2 | \sigma_1, \beta_2)p(\sigma_1 | \beta_1)p(\beta_1)p(\beta_2) . \quad (4.26)$$

Since, as discussed in Section 4.3.6, we do not assume further knowledge about the values of $k_1$ and $\omega_m$, $m = 1, 2$, accounting for constraints (4.7)-(4.11) we assume $p(\nu_2 | \nu_1)$ is uniform between $\nu_1$ and $\nu_1 + T_{LIR}$, $p(\sigma_2 | \sigma_1, \beta_2)$ is uniform between $\sigma_1$ and $T_{LIR} \sqrt{\Gamma \left( \frac{1}{\beta_2} \right) / \Gamma \left( \frac{3}{\beta_2} \right)}$, $p(\sigma_1 | \beta_1)$ is uniform between 0 and $T_{LIR} \sqrt{\Gamma \left( \frac{1}{\beta_1} \right) / \Gamma \left( \frac{3}{\beta_1} \right)}$, and $p(\beta_m)$, $m = 1, 2$, is uniform between 1 and a deterministic parameter, $G$. Furthermore, we assume $p(k_1)$ is uniform between 0 and 1. Exact expressions for (4.24) are given in Appendix B. For the numerical results presented in the following section we evaluate the HCRLB through Monte-Carlo simulations considering the above uniform distributions.

$^8$Note that while the EM algorithm works on vectors $\Lambda_i$, the actual inputs to our classifier are PD measurements. Thus, in forming the HCRLB, we use $x_i$ rather than $\Lambda_i$. 

78
4.4 Performance Evaluation

In this section, we present results from both computer simulations and sea trials to demonstrate the performance of our classification algorithm. The results are presented in terms of detection probabilities of LOS, SNLOS, and ONLOS links. In addition, we measure estimation errors $|\nu_m^p - \nu_m|$, $|\sigma_m^p - \sigma_m|$, and $|\beta_m^p - \beta_m|$. We compare our results to the HCRLB presented in Section 4.3.8, as well as to several benchmark methods. The purpose of the simulations is to evaluate the performance of our classifier in a controlled environment, while results from sea-trial measurements reflect performance in actual UWACs.

4.4.1 Simulations

Our simulation setting includes a Monte-Carlo set of 10000 channel realizations, where two time-synchronized nodes, uniformly randomly placed into a square area of 1 km, exchange packets. The setting includes two horizontal and two vertical obstacles of length 20 m, also uniformly randomly placed into the square area, such that a LOS always exists between the two nodes. In each simulation, we consider a packet of 200 symbols of duration $T_s = 10$ msec and bandwidth $B = 6$ kHz transmitted at a propagation speed of $c = 1500$ m/sec. To model movement in the channel (dealt with by forming groups $\Lambda_l$), during packet reception the two nodes move away from each other at constant relative speed of 1 m/sec, and $d_{LOS}$ is considered as the LOS distance between the nodes when the 100th symbol arrives.

In our simulations, we use model (4.1) to obtain set $X$ as follows. For each channel realization and node positions, we find the LOS distance between the two nodes, and determine $\nu_1 = x_{LOS}$. Based on the position of nodes and obstacles, we identify ONLOS links as single reflections from obstacles and determine $\nu_3$ as the average delay of the found ONLOS links. We use $T_{LIR} = 0.1$ sec and based on constraint (4.11), we randomize $\nu_2$ according to a uniform distribution between $\nu_1$ and $\nu_1 + T_{LIR}$. For the other distribution parameters $\theta$, we determine $\beta_m$, $m = 1, 2, 3$ as an integer between 1 and 6 with equal probability (i.e., $G = 6$ in (4.26)), and $\sigma_m$, $m = 1, 2, 3$ according to (4.8) with $\varsigma_m$ uniformly distributed between 0 and $(T_{LIR})^2$, preserving $\varsigma_1 < \varsigma_2$. Based on model (4.5), we then randomize $x_i$, $i = 1, \ldots, 200$ using distribution (4.6) and a uniformly distributed $k_m$, $m = 1, 2, 3$ between 0 and 1, while keeping $\sum_{m=1}^{3} k_m = 1$ and setting $k_3 = 0$ if no ONLOS link is identified. Considering the discussion in Section 4.3.6, we use $\Delta \nu = \frac{1}{c}$ as a detection threshold to check if measurements in vector $X^e$ correspond to a single link. Additionally, for forming groups $\Lambda_l$ (see (4.15)), we use an assumed coherence time $\tilde{T}_c = a \cdot T_s$, where $a \in \{1, 5, 10\}$, and a quantization threshold $\Delta T = 0.16$ msec based on the bandwidth of the transmitted signals. Note that since the distance between nodes changed by 2 m during reception of the 200 symbols, if $a > \frac{2}{\Delta T \cdot c} \approx 8.3$, condition (4.15b) is irrelevant, whereas $a = 1$
Chapter 4. LOS and NLOS Classification for UWL

Figure 4.1: Classification results: (a) $P_{rd,\text{non-ONLOS}}$ and $P_{rd,\text{ONLOS}}$ vs. $\gamma_{ UB}$, (b) Empirical detection probabilities of LOS and SNLOS.

results into single-element vectors $\Lambda_i$.

To simulate channel attenuation (4.2), we use $\gamma = 15$, $\alpha = 1.5$ dB/km (considering a carrier frequency of 15 kHz [5]), and set $\epsilon$ to be zero-mean Gaussian with variance $5/\text{dB}^2/\mu\text{Pa}@1\text{m}$. We use a source power level of 100 dB/\mu Pa@1m and a zero-mean Gaussian ambient noise with power 20 dB/\mu Pa@1m, such that the SNR at the output of the channel is high. Attenuation in LOS and SNLOS links is determined based on (4.2), while for ONLOS links we use (4.3) and set $RL = 10$ dB/\mu Pa@1m.

To obtain the lower bound on RSS-based distance, $d_{RSS, LB}^i$, $i = 1, \ldots, 200$, we use the attenuation model in (4.2) with $\gamma_{ UB} = 20$ and $\alpha_{ UB} = 2$ dB/km. An implementation of the simulation environment can be downloaded from [96].

First, in Figure 4.1a we show empirical detection probabilities for ONLOS and non-ONLOS links as a function of $\gamma_{ UB}$, as well as corresponding results using bounds (4.14) and (4.13). We observe a good match between the empirical results and the analytical bound for $P_{\text{rd, ONLOS}}$, and that $P_{\text{rd, ONLOS}}$ is hardly affected by $\gamma_{ UB}$. However, $P_{\text{rd, non-ONLOS}}$ increases dramatically with $\gamma_{ UB}$, and the corresponding bound in (4.13) is less tight. This is because choosing $\gamma_{ UB} < \gamma$ might lead to $d_{RSS, LB} > d_{LOS}$ and neglectance of $\alpha$ in (4.13) causes analytical inaccuracies. For $P_{\text{rd, ONLOS}}$, however, the large RL is more significant than the effect of $\gamma_{ UB}$.

In Figure 4.1b, we show empirical detection probabilities\footnote{We note that detection probabilities are calculated only when vector $X$ consists of both LOS and SNLOS related PD measurements; classification cannot be made otherwise.} for LOS ($LOS (EM)$) and SNLOS ($SNLOS (EM)$) links, the total detection probability ($ALL (EM)$), which is calculated as the rate of correct classification (of any link). Also shown are classification performance using only the initialization process ($init$), i.e., before the EM algorithm is employed, as well as the results for classification without prior identification of ONLOS links ($No \text{ ONLOS ID}$), i.e., without the first step of our algorithm. For the latter, we consider two cases: i) $M = 2$ and ii) $M = 3$, where in the
second case ONLOS links are considered as a separate class. We observe that the constrained EM algorithm achieves a significant performance gain compared to the K-means algorithm, used in the initialization process. Results show that for the former, the detection rate is more than 92% for both LOS and SNLOS. We also observe a performance degradation when the first step for ONLOS identification is not performed. This degradation is more significant when ONLOS links are considered as SNLOS links, i.e., when $M = 2$. Please refer to the discussion in Section 4.1.3, for explanations of this result.

In Figure 4.2, we show the empirical complimentary cumulative distribution (C-CDF) of $\rho_{\text{err}}$ from (4.27).

$$\rho_{\text{err}} = |\hat{x} - d_{\text{LOS}}|,$$  \hspace{1cm} (4.27)

where $\hat{x}$ is i) $\upsilon_1 P_{\text{last}}$, ii) the average of the elements in $X$ ($E(x_i)$), iii) the minimum of $X$ ($\min(x_i)$), or iv) the average value of $X$ after removal of outliers, as suggested in [95] (Outlier). Results for $\hat{x} = \upsilon_1 P_{\text{last}}$ are shown for $\hat{T}_c \in \{T_s, 5T_s, 10T_s\}$. The results in Figure 4.2 are also compared with the HCRLB presented in Section 4.3.8. We observe that the Outlier method outperforms the naive approaches of using the average or minimum value of $X$, where the latter performs extremely poorly for large values of $\rho_{\text{err}}$. However, the use of our classifier improves results significantly. For example, the proposed classifier achieves $\rho_{\text{err}} \leq 7$ m in 90% of the cases, compared to 11.2 m when using the Outlier method, and the results are close to the HCRLB. Such an improvement immediately translates into better localization performance as PD estimation errors significantly decrease. Comparing results for different values of $\hat{T}_c$, we observe that using equivalence constraints (i.e., $\hat{T}_c > T_s$), performance slightly improves compared to the case of $\hat{T}_c = T_s$. However, a tradeoff is observed as results for $\hat{T}_c = 5T_s$ are marginally better than for $\hat{T}_c = 10T_s$. This is because of erroneous
4.4.2 Sea Trials

While our simulations demonstrate good classification performance for our algorithm, the tests relied on the distribution model (4.6), and upper bound on transmission loss models (4.2) and (4.3), which might not be faithful representations of realistic UWACs and PD estimators. Thus, we present classification results for UWACs measured during three sea trials conducted in Israel and Singapore. One of these experiments was conducted in a harbor environment to test only ONLOS classification, while the other two were in shallow water to test LOS and SNLOS classification. To acquire PD measurements from recorded sea-trial data, we used the phase-only-correlator (POC) detector as described in [47]. For the $i$th received signal, $x_i$ is estimated as the first peak at the output of the POC that passes a detection threshold.
Classifying ONLOS links

In this section, we show the performance of ONLOS link identification for an experiment conducted at the Haifa harbor, Israel, in May 2009. The experiment included four vessels, each representing an individual node in the network. Here we consider a subset of the recorded data for which nodes were static. In each vessel, a transceiver was deployed at a fixed depth of 3 m, at a water depth of around 6 m. The four nodes were time synchronized using GPS and transmitted with equal transmission power at a carrier frequency of 15 kHz. Referring to Figure 4.4a, node 2 was placed at a fixed location 2A, while nodes 1, 3 and 4 sent packets to node 2 while moving between various locations, creating a controlled environment of five non-ONLOS and four ONLOS communication links with a maximum transmission distance of 1500 m. For each link, \((2, j), j \in \{1,3,4\}\), we evaluated (i) \(d_{PD}\) as the product of an assumed propagation speed of 1550 m/sec and the position of the first peak of the POC for the synchronization signal of each received packet, and (ii) \(d_{RSS,LB}\), employing an energy detector over the synchronization signal and using (4.2) for \(\alpha_{UB} = 2\) db/km and \(\gamma_{UB} = 20\). We note that results only changed slightly when alternative methods for obtaining \(d_{RSS,LB}\) and \(d_{PD}\) were applied.

In Figure 4.4b, we present values of \(d_{RSS,LB}\) and \(d_{PD}\) for each of the 9 communication links. Applying our proposed ONLOS link identification method, all four ONLOS links were correctly classified and there was no false classification of non-ONLOS links. In particular, we observe that for all ONLOS links, \(d_{PD}\) is much lower than \(d_{RSS,LB}\). Since the latter was obtained from model (4.2) (i.e., a model for LOS) and we assumed spherical propagation (i.e., \(\gamma_{UB} = 20\) in (4.2)), which is a very loose upper bound on power attenuation in the UAC (e.g., [5], [86]), this difference validates our assumption that the RL level of the reflecting objects (which could have been harbor docks, ship hulls, etc.) are sufficiently high to satisfy assumption (4.4).

\[
\begin{array}{|c|c|c|c|c|}
\hline
\text{Link} & d_{RSS,LB} [\text{m}] & d_{PD} [\text{m}] & \text{Link} & d_{RSS,LB} [\text{m}] & d_{PD} [\text{m}] \\
\hline
(2A,3B) & 579 & 780 & (2A,1B) & 1957 & 1105 \\
(2A,4A) & 179 & 242 & (2A,3C) & 1639 & 740 \\
(2A,4D) & 343 & 415 & (2A,1D) & 1549 & 1254 \\
(2A,1A) & 428 & 610 & (2A,1C) & 1816 & 950 \\
(2A,3A) & 647 & 817 & & & \\
\hline
\end{array}
\]
Classifying non-ONLOS links

Next, we present results from two separate experiments conducted in open sea: (i) the first along the shores of Haifa, Israel, in August 2010 and (ii) the second in the Singapore straits in November 2011, with water depths of 40 m and 15 m respectively. This is done to demonstrate our classifier’s performance in different sea environments. As communication links were all non-ONLOS links in both experiments, $X^{\text{ex}} = X$ and we only present results for LOS/SNLOS classification.

The first sea trial included three vessels, representing three mobile nodes, which drifted with the ocean current at a maximum speed of 1 m/sec, and were time-synchronized using the method described in Chapter 2. Throughout the experiment, the node locations were measured using GPS receivers (whose accuracy was up to $\pm 5$ m), and the sound speed was measured to be $c = 1550$ m/sec with deviations of no more than 2 m/sec across the water column. Each node was equipped with a transceiver, deployed at 10 meters depth, and transmitted more than 100 data packets which were received by the other two nodes. Each packet consisted of 200 direct-sequence-spread-sequence (DSSS) symbols of duration $T_s = 10$ msec and a spreading sequence of 63 chips was used.

Table 4.1: Israel sea trial: distribution of estimated values of $\beta_{m,\text{last}}^1$, $m = 1, 2$.

<table>
<thead>
<tr>
<th>Measure</th>
<th>$\beta = 1$</th>
<th>$\beta = 2$</th>
<th>$\beta = 3$</th>
<th>$\beta = 4$</th>
<th>$\beta = 5$</th>
<th>$\beta = 6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta_{1,\text{last}}^1$, $T_c = T_s$</td>
<td>11%</td>
<td>7%</td>
<td>1.8%</td>
<td>1.8%</td>
<td>0%</td>
<td>77%</td>
</tr>
<tr>
<td>$\beta_{1,\text{last}}^1$, $T_c = 10T_s$</td>
<td>16%</td>
<td>11%</td>
<td>1%</td>
<td>1%</td>
<td>0%</td>
<td>67%</td>
</tr>
<tr>
<td>$\beta_{2,\text{last}}^2$, $T_c = T_s$</td>
<td>0%</td>
<td>54%</td>
<td>9%</td>
<td>5%</td>
<td>5%</td>
<td>24%</td>
</tr>
<tr>
<td>$\beta_{2,\text{last}}^2$, $T_c = 10T_s$</td>
<td>0%</td>
<td>57%</td>
<td>6%</td>
<td>5%</td>
<td>5%</td>
<td>25%</td>
</tr>
</tbody>
</table>

An estimation parameter of interest is $\beta_{m,\text{last}}^m$, which determines the type of distribution of the $m^{\text{th}}$ class. In Table 4.1, we show the distribution of $\beta_{1,\text{last}}^1$ and $\beta_{2,\text{last}}^2$ for different values of $T_c$. As the results are similar for both $T_c = T_s$ and $T_c = 10T_s$, this implies that clustering PD measurements in vectors $\Lambda_i$ does not affect the estimated type of distribution. We also observe that the LOS class seems to lean towards $\beta_{1,\text{last}}^1 = 6$, which implies a uniform distribution, while SNLOS measurements cluster around $\beta_{2,\text{last}}^2 = 2$, which corresponds to the normal distribution.

In Figure 4.5, we show the empirical C-CDF of

$$
\rho^{\text{err}} = |c\hat{x} - E(d_i)|, \tag{4.28}
$$

where $E(d_i)$ is the mean of the GPS-based transmitter-receiver distance during the reception of each packet, $\hat{x}$ is either $v_{1,\text{last}}^1$, the average of the PD measurement in $X_i$ ($E(x_i)$), the minimum of $X$ (min$(x_i)$), or the average of the obtained PD measurements after removal of outliers, i.e., the method described in [95] (Outlier).Results
are shown for $\hat{T}_c = aT_s$, where $a \in \{1, 10, 20\}$. Assuming GPS location uncertainties of 5 m, we require $\rho^{err}$ to be below 6 m. Results show that $\rho^{err}$ for $\hat{x} = \min(x_i)$ is lower than for $\hat{x} = E(x_i)$ and almost the same as the results for the Outlier method. However, proposed classifier achieves always the lowest error, which is smaller than 6 m in more than 90% of the cases (compared to 55% for $\hat{x} = \min(x_i)$). Comparing results for the different values of $\hat{T}_c$, we observe that a notable advantage for $\hat{T}_c = 5T_s$. Since in the sea trial, during packet reception nodes were almost static, this difference is due to the time varying channel conditions.

The second sea trial included two UWAC modems, manufactured by Evologics GmbH, which were deployed at a depth of 5 m. One of them was suspended from a static platform and the other from a boat anchored to the sea bottom. Throughout the experiment, the boat changed its location, resulting in three different transmitter-receiver distances which were monitored using GPS measurements (whose accuracy was on average $\pm 1$ m). Measurements $x_i \in X$ were obtained every 6 sec. For each transmission distance, the boat remained static for 20 min, allowing around 200 measurements $x_i$ at each node. In this experiment, a propagation speed of $c = 1540$ m/sec, as measured throughout the year in the Singapore straits [86], was considered.

In Figure 4.6, since in the second sea trial the boat moved around its anchor while $X$ was obtained, we show the histogram of

$$\rho_i^{err} = cx_i - d_i$$  \hspace{1cm} (4.29)

for a single vector $X$, where $d_i$ is the GPS-based transmitter-receiver distance measured at time $t_i$ (i.e., when $x_i$ is measured), with mean and variance of $E(d_i) = 324.1$ m and $\text{Var}(d_i) = 3$ m$^2$, respectively. We also plotted $cE(x_i) - E(d_i)$ and
c \min(x_i) - E(d_i) as well as PDFs (4.6) of the LOS and SNLOS classes for estimation \( \theta^{P_{\text{last}}} \), for which \( c v_1^{P_{\text{last}}} - E(d_i) = 0.1 \text{ m} \) and \( c v_2^{P_{\text{last}}} - E(d_i) = 18.4 \text{ m} \). We estimated \( \beta_1^{P_{\text{last}}} = 1 \) and \( \beta_2^{P_{\text{last}}} = 6 \), which matches the narrow and the near uniform distributions observed for the LOS and SNLOS classes, respectively. We note the good fit between the shape of the estimated PDF and the histogram for both classes. In addition, we observe that estimation \( v_1^{P_{\text{last}}} \) gives much better results than the naive approach of taking the average or minimum value of \( X \).

Table 4.2: Singapore sea trial: \( \frac{c \hat{x} - E(d_i)}{E(d_i)} \).

<table>
<thead>
<tr>
<th>Average distance [m]</th>
<th>( \hat{x} = v_1^{P_{\text{last}}} )</th>
<th>( \hat{x} = v_2^{P_{\text{last}}} )</th>
<th>( \hat{x} = E(x_i) )</th>
<th>( \hat{x} = \min(x_i) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>83</td>
<td>0.007</td>
<td>0.29</td>
<td>0.19</td>
<td>0.07</td>
</tr>
<tr>
<td>324</td>
<td>0.0017</td>
<td>0.05</td>
<td>0.01</td>
<td>0.15</td>
</tr>
<tr>
<td>584</td>
<td>0.0018</td>
<td>0.03</td>
<td>0.02</td>
<td>0.002</td>
</tr>
</tbody>
</table>

In Table 4.2 we present results of the ratio \( \frac{c \hat{x} - E(d_i)}{E(d_i)} \) for the three locations of the boat, averaged for the two nodes, for \( \hat{x} = v_1^{P_{\text{last}}}, v_2^{P_{\text{last}}}, E(x_i), \min(x_i) \). We observe that \( \min(x_i) \) usually, but not always, results in better propagation delay estimation than \( E(x_i) \), which in turn always results in better estimation than \( v_2^{P_{\text{last}}} \), as expected. However, best results are obtained for \( \frac{c v_1^{P_{\text{last}}} - E(d_i)}{E(d_i)} \) with average of 0.7 m compared to more than 10 m for the other methods.

The results in Figures 4.5, 4.6, and Table 4.2 show a significant differences of up to 30 m between range measurements. Since motion of nodes was limited during the time each set of measurements \( X \) was obtained in both sea trials, this variation is due to a large difference between the delays in the LOS and SNLOS links, and validates our
assumptions (4.7) and (4.11). Moreover, since model mismatch of (4.6) would have considerably affected the accuracy of estimating $v_1^{\text{last}}$, the good estimation results in Figure 4.5 and Table 4.2, as well as the distribution of $\beta$ in Table 4.1, validate the choice of model (4.6) in two different sea environments (Israel and Singapore).

Finally, we present the effect of our classifier on localization accuracy. We use recordings of PD measurements obtained from the first sea trial for the localization method described in Chapter 2. This method time-synchronizes and localizes nodes using PD measurements from anchors. The results are shown in terms of the distance error, $\rho_e$, of the estimated location for ranging using i) the estimated $v_1^{\text{last}}$ value (for $\bar{T}_c = 5T_s$), and ii) a single PD measurement obtained from the output of the POC for a chirp signal transmitted at the prefix of each packet ($\text{sync}$). The results are shown in Figure 4.7 for localization using $W = [10, 30]$ packet exchanges (see Chapter 2 for details). Note that the large errors are due to GPS location uncertainty, which we used as reference and which for this sea trial corresponds to a localization error of up to 15 m. Using our classification approach, we observe a significant improvement of up to 10 m in localization accuracy.

### 4.5 Conclusions

In this chapter, we utilize the variation of PD measurements due to continuous motion of nodes at sea and classify the former into three classes: LOS, SNLOS, and ON-LOS. We presented a two-step classifier which first compares PD-based and received signal strength based ranging to identify ONLOS links, and then, for non-ONLOS
links, classifies PD measurements into LOS and SNLOS paths, using a constrained expectation maximization algorithm. We also offered a heuristic approach to efficiently maximize the log-likelihood function, and formalized the Cramér-Rao Bound to validate the performance of our method using numerical evaluation. As our classifier relies on the use of simplified models, alongside simulations, we presented results from three sea trials conducted in different sea environments. Both our simulation and sea trial results confirmed that our classifier can successfully distinguish between ONLOS and non-ONLOS links, and is able to accurately classify PD measurements into LOS and SNLOS paths.
Chapter 5

A Machine Learning Approach for Underwater DR

Our UT scheme presented in Chapter 3 relies heavily on speed and orientation measurements from on-board INS producing acceleration measurements. To this end, we aim to obtain a set of distance and heading measurements between each two locations for which the tracked node receives a communication packet from its neighbor anchor nodes. Using acceleration measurements, this procedure is performed through DR. As illustrated in Figure 1.5, at the presence of ocean waves these measurements need to be projected to the horizontal plane by compensating for the time-varying pitch angle. The current methods discussed in Section 1.1.3 perform such projection through direct measurement of the TN orientation. However, this approach limits the resolution of DR navigation, as measurements are filtered over the period of several ocean waves [54]. Moreover, due to energy constraints, gyrocompass are not always in use and orientation measurements may not be available. A different approach may be to directly project acceleration measurements using the principal component analysis (PCA) (cf. [122, 123]). PCA gives the axis coordinate system (or the transformation matrix to that system) along whose axes the variation of change is smallest. If the vessel acceleration is constant in a certain plane, PCA projection aligns acceleration measurements with that plane. The underlining hard assumption of this method is a large variance of acceleration measurements in the $y$ and $z$ axes and a fixed acceleration in the $x$ axis. In the absence of alternative approaches to compensate for the time-varying pitch angle with no orientation measurements, we use the so-called PCA method as a benchmark.

In this chapter, we offer a machine learning approach to compensate for the pitch angle using a single 3-D acceleration sensor. Our method, denoted as the DR-navigation-using-a-single-accelerometer ($DR-A$), is based on the observation that, due to ocean waves, the vessel pitch angle is periodic in nature. The intuition behind our method is that, when the vessel pitch angle is fixed, direction can be determined without using orientation measurements. Hence, by identifying classes of acceleration measurements such that in each class the pitch angle is approximately the same, we can readily project acceleration measurements into a reference coordinate system. While our method can be used also for AUVs submerged deep at sea, it is mostly designated to compensate for wave induced motions. The main contribution of our work is therefore in obviating the need to compensate for the vessel time-varying pitch angle via direct measurement of the vessel orientation using dedicated hard-
Chapter 5. A Machine Learning Approach for Underwater DR

Figure 5.1: Flow chart for the DR-A algorithm.

ware, e.g., gyrocompass, without setting hard limitation on the vessel motion. Our DR-A method is executed sequentially. First, we estimate the vessel heading angle and project acceleration measurements such that these are aligned with the vessel motion. Next, we classify the projected measurements into pitch-states of similar pitch angles. This is performed using a constraint EM algorithm. Third, per pitch-state, we project acceleration measurements (whose pitch angles are similar) to compensate on the vessel pitch angle. After this step, the vessel’s projected local coordinate system is considered aligned with the the vessel heading direction and the East-North-Up coordinates system. Finally, we integrate the projected measurements to evaluate the distance traveled by the vessel. A flow chart of the above process is given in Figure 5.1.

The remainder of this chapter is organized as follows. Our system model and assumptions are introduced in Section 5.1. In Section 5.2, we present the steps of our DR-A method. Section 5.3 includes performance evaluation of our algorithm in simulations (Section 5.3.1) and using realistic data obtained during a designated sea trial (Section 5.3.2). Finally, conclusions are offered in Section 5.4.

5.1 System Model

Our system consists of a three-axis accelerometer device attached to a vessel, which produces $N$ acceleration measurements during a set period of time, $[t_{\text{start}}, t_{\text{end}}]$, where $t_{\text{end}} - t_{\text{start}}$ is expected to be in the order of 10 seconds. We assume that at time $t_{\text{start}}$, the vessel’s initial speed and heading with respect to a reference coordinate system (e.g. UTM) is given as $v_i$ and $\mathcal{Z}_{\text{init}}$, respectively. Our objective is to track the vessel heading angle between time instances $t_{\text{start}}$ and $t_{\text{end}}$, as well as to estimate
the distance traveled by the vessel during the considered time period\(^\text{10}\), \(\tilde{d}_{n,j}\). This is required for compensation of nodes’ mobility for localization purposes.

Let \(a_{n,x}\), \(\angle_n\), and \(\alpha_n\), \(n = 1, \ldots, N\) be the vessel’s acceleration along the vessel heading direction, the vessel heading direction with respect to the reference coordinate system, and the vessel orientation (i.e., the direction in which the vessel’s bow is pointing) with respect to its heading direction \(\angle_n\), respectively. We assume that the change in \(\alpha_n\) is only due to timely changes in \(\angle_n\). That is, that during period \([t_{\text{start}}, t_{\text{end}}]\) the vessel’s orientation with respect to the reference system does not change. In addition, we assume both \(a_{n,x}\) and \(\angle_n\) are slowly time-varying, such that the considered time period can be divided into \(W = \frac{t_{\text{end}} - t_{\text{start}}}{T_c}\) time slots of duration \(T_c\) (representing the coherence time of the system), in each of which \(N_{\text{slot}} = \frac{NT_c}{t_{\text{end}} - t_{\text{start}}}\) acceleration measurements are acquired, where in each time slot \(w\), \(\angle_n\) and \(a_{n,x}\) are assumed fixed and equal \(\angle_w\) and \(a_{w,x}\), respectively. For simplicity, we assume the vessel’s motion is perfectly correlated with ocean waves such that the vessel roll angle can be neglected. However, extension is suggested also for the case of time-varying pitch and roll angles (see Section 5.2.5).

In the \(w\)th time slot, on time \(t_n\), \(n = 1, \ldots, N_{\text{slot}}\), \(t_{\text{start}} \leq t_n \leq t_{\text{end}}\), the accelerometer device produces a three-axis acceleration measurement vector, \(\hat{a}_n = [\hat{a}_{n,x}, \hat{a}_{n,y}, \hat{a}_{n,z}]\), at local coordinates, which are grouped into a matrix \(\hat{A}\). Vector \(\hat{a}_n\) is a projection of a vector \(a_n = [a_{n,x}, a_{n,y}, a_{n,z}]^T\) representing the vessel’s acceleration in a 3-D coordinate system referred to as the horizontal plane. The coordinates of the horizontal plane are defined so that the \(x\) axis is aligned with the vessel’s heading direction, assuming there is no movement in the \(y\) axis (since within the time slot the heading is assumed fixed), and the \(z\) axis is as in the East-North-Up coordinates system. Thus, the horizontal plane is separately defined for each time slot of duration \(T_c\) with assumed fixed heading and acceleration. Referring to the illustration on Figure 1.5, the projection from \(a_n\) into \(\hat{a}_n\) is modeled by a rotating vector, \([a_{n,x}, a_{n,y}, a_{n,z}]^T\) about its \(y\)-axis by the vessel time-varying pitch angle, \(\rho_n\), then its \(z\)-axis by angle \(\alpha_n\), and adding a zero-mean Gaussian noise, \(e_n\), with a per-axis standard deviation of \(\frac{\xi}{2} \text{ m/(seconds)^2}\), so that:

\[
\begin{bmatrix}
\hat{a}_{n,x} \\
\hat{a}_{n,y} \\
\hat{a}_{n,z}
\end{bmatrix} =
\begin{bmatrix}
\cos \rho_n & 0 & \sin \rho_n \\
0 & 1 & 0 \\
-\sin \rho_n & 0 & \cos \rho_n
\end{bmatrix}
\begin{bmatrix}
\cos \alpha_n & -\sin \alpha_n & 0 \\
\sin \alpha_n & \cos \alpha_n & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
a_{n,x} \\
a_{n,y} \\
a_{n,z}
\end{bmatrix} +
\begin{bmatrix}
e_{n,x} \\
e_{n,y} \\
e_{n,z}
\end{bmatrix}.
\]

\(5.1\)

We note that while \(a_{n,y}\) is assumed zero, \(a_{n,z}\) can be non-zero. This is due to the "up and down" movements of the vessel with the ocean waves. However, we expect a small value for \(a_{n,z}\). This is required to solve an ambiguity in estimating the pitch angle\(^\text{11}\). In fact, we are interested in cases where |\(a_{n,x}\)| >> |\(a_{n,z}\)|. Otherwise, the

\(^{10}\)Throughout this chapter upper case bold symbols represent matrices, bold symbols represent vectors, \(\hat{\theta}\) represents the measurement of true parameter \(\theta\), and \(\tilde{\theta}\) represents the estimation of \(\theta\).

\(^{11}\)This assumption is validated using a realistic wave model as well as results from a sea trial.
vessel is moving in almost fixed speed, $v_i$, and projection is not required. However, it is important to note that by (5.1), even when $a_{n,z}$ is zero, compensation over both $\alpha_n$ and $\rho_n$ is required.

For each time instance $t_n$, we aim to estimate $\rho_n$ and $\alpha_n$, and to project the vessel’s local coordinate system about the $z$-axis followed by its $y$-axis. The first projection converts the set of measurements $\hat{A}$ into a projected set $\hat{A}^h$ with elements $\hat{a}_{n}^h$. The second projection further converts the set of measurements into a projected set $\hat{A}_{n}^{hp}$ with elements $\hat{a}_{n}^{hp}$. Since after the second projection the local coordinate system is expected to be aligned with the horizontal plane, we can readily integrate elements $\hat{a}_{n}^{hp}$ to estimate $\hat{d}_{i,j}$. Furthermore, since the vessel’s orientation with respect to the reference coordinate system is assumed fixed during $[t_{start}, t_{end}]$, such projection allows us to estimate the per-time slot change in the vessel’s heading direction with respect to the initial heading, $\angle^{\text{init}}$.

By (5.1), the set of measurements $\hat{A}$ are affected by the pitch angle of the vessel. Much like the ocean waves, the vessel pitch angle is periodic in nature and likely, $-\frac{\pi}{4} < \rho_n < \frac{\pi}{4}$. However, being induced from ocean waves, we do not assume a fixed period for the vessel pitch angle. Given this periodicity, we can identify time instances for which the vessel pitch angle is approximately the same. This observation sets the stage for classifying measurements into pitch-states of assumed fixed pitch angle, given the following model.

Let $\hat{A}(w)$ with elements $\hat{a}_n(w) = [\hat{a}_{n,x}(w), \hat{a}_{n,z}(w)]^T$, $n = 1, \ldots, N_{\text{slot}}$ be the set of 2-D acceleration measurements acquired during the $w$th time slot, $w = 1, \ldots, W$, and sorted by their measurement time. We consider a Gaussian mixture with $M$ pitch-states for the distribution of $\hat{A}(w)$ such that for $\Theta(w) = \{\omega_{1,x}(w), \omega_{1,z}(w), k_1(w), \ldots, \omega_{M,x}(w), \omega_{M,z}(w), k_M(w)\}$,

$$P(\hat{A}(w)|\Theta(w)) = \prod_{\hat{a}_n(w) \in \hat{A}(w)} \sum_{m=1}^{M} k_m(w) P_m(\hat{a}_{n,x}(w)|\omega_{m,x}(w)) P_m(\hat{a}_{n,z}(w)|\omega_{m,z}(w)),$$

(5.2)

where $\omega_{m,x}(w) = [\mu_{m,x}(w), \sigma_{m,x}^2(w)]$, and $\omega_{m,z}(w) = [\mu_{m,z}(w), \sigma_{m,z}^2(w)]$, are the distribution parameters of the $m$th pitch-state with mean $\mu_{m,x}(w)$ and $\mu_{m,z}(w)$, and variance $\sigma_{m,x}^2(w)$ and $\sigma_{m,z}^2(w)$, respectively. $k_m(w)$ ($\sum_{m=1}^{M} k_m(w) = 1$) is the a-priori probability of the $m$th pitch-state. Model (5.2) is used to classify acceleration measurements into pitch-states using a constraint EM algorithm. For clarity, in the following we drop the time slot subindex, $w$, and consider projection of measurements acquired during a single time slot.
5.2 The DR-A Method

Referring to model (5.1), in the DR-A method we estimate angles \(\alpha_n\) and \(\rho_n\) to project measurement in \(\hat{A}\) into the horizontal plane and obtain an estimation for \(a_{n,x}\), \(n = 1, \ldots, N_{\text{slot}}\). While the orientation angle, \(\alpha_n\), is assumed fixed during the time slot, it is not the case for the pitch angle, \(\rho_n\). For this reason, we separately compensate the vessel heading and pitch angles. The former is done jointly for all measurements acquired during a single time slot, resulting in a projected set \(\hat{A}^h\). For the latter, we use machine learning approach to classify acceleration measurements into states of similar pitch angles, and compensate the pitch angle to form a projected set \(\hat{A}^{h,p}\) for every state, considered aligned with the horizontal plane. Finally, we combine projected measurements of all time slots, and, using the initial (given) velocity \(v_i\), integrate them over the time period \([t_{\text{start}}, t_{\text{end}}]\) to obtain \(\tilde{d}_{i,j}\). In the following sections we describe in details the steps of our method.

5.2.1 Forming \(\hat{A}^h\): Estimation of the Heading Angle

Before forming classes of similar pitch angles, we project \(\hat{a}_n \in \hat{A}\), \(n = 1, \ldots, N_{\text{slot}}\) to form \(\hat{a}_n^h = [\hat{a}_{n,x}^h, \hat{a}_{n,y}^h, \hat{a}_{n,z}^h]\), for which \(\hat{a}_{n,z}^h = \hat{a}_{n,z}\) and (if the vessel’s pitch angle is zero) \(\hat{a}_{n,x}^h\) aligns with the heading of the vessel. This projection is performed as follows.

Referring to Figure 1.5, \(\alpha_n\) denotes the angle between the heading direction of the vessel, \(\angle_n\), and the accelerometer’s local \(x\)-axis. Since in every time slot the vessel is assumed moving in a fixed heading, without measurement noise, there should be no acceleration caused by the vessel motion along the axis which is perpendicular to the heading direction. Therefore, we expect

\[
\hat{a}_{n,x} \sin \alpha_n - \hat{a}_{n,y} \cos \alpha_n \approx 0 .
\]  

We look for estimation \(\hat{\alpha}_n = \tilde{\alpha}\) which minimizes \(\sum_{n=1}^{N} (\hat{a}_{n,x} \sin \alpha_n - \hat{a}_{n,y} \cos \alpha_n)\) by,

\[
\tan \tilde{\alpha} = \frac{\sum_{n=1}^{N} \hat{a}_{n,y}}{\sum_{n=1}^{N} \hat{a}_{n,x}} .
\]  

Once \(\tilde{\alpha}\) is determined through (5.4), we project measurements in \(\hat{A}\) and form matrix \(\hat{A}^h\). This is performed by multiplying \(\hat{a}_n \in \hat{A}\), \(n = 1, \ldots, N_{\text{slot}}\) with the rotation matrix

\[
\begin{bmatrix}
\cos \tilde{\alpha} & \sin \tilde{\alpha} & 0 \\
-\sin \tilde{\alpha} & \cos \tilde{\alpha} & 0 \\
0 & 0 & 1
\end{bmatrix} .
\]
In the $w$th time slot, the estimated angle $\tilde{\alpha}_w$ from (5.4) is compared with $\tilde{\alpha}_{w-1}$ to estimate the heading direction of the vessel, $Z_w$, assumed fixed in each time slot. Let $\Delta Z_w$ be the change in the heading direction of the vessel between the $w-1$th time slot and the $w$th one. Then, $Z_w = Z_{w-1} - \Delta Z_w$. Recall we assume the direction of the vessel’s bow with respect to the reference coordinate system (e.g. UTM) does not change during the considered time period, $[t_{\text{start}}, t_{\text{end}}]$. Thus, $\Delta Z_w = \tilde{\alpha}_w - \tilde{\alpha}_{w-1}$, and

$$Z_w = Z_{w-1} - \Delta Z_w,$$

where $Z_0 = Z_{\text{init}}$.

### 5.2.2 Forming $\hat{A}^{h,p}$: Estimating the Time-varying Pitch Angle

After forming matrix $\hat{A}^h$, we estimate the vessel pitch angle to form matrix $\hat{A}^{h,p}$, which will be used to calculate $\tilde{d}_{i,j}$. Since the vessel pitch angle is time-varying, different from estimation (5.4), elements of $\hat{A}^h$ cannot be directly projected to $\hat{A}^{h,p}$. For this reason, traditionally, a direct measurement of the vessel orientation is used via, e.g. gyrocompass, which may be unavailable or too noisy. In this section we describe an alternative approach, where based on the observation that the vessel pitch angle is periodic in nature, we form matrix $\hat{A}^{h,p}$ by grouping acceleration measurements to classes of similar pitch angles.

Based on model (5.2), to identify measurements of similar pitch angles we use a constraint EM algorithm and classify the elements in $\hat{A}^h$ into $\tilde{M}$ pitch-states, where $\tilde{M}$ is a pre-defined number of states. For an assumed coherence time, $T_c$, classification is performed separately for each time slot. Since the effect of the pitch angle is similar in the $x$ and $z$ axes, classification is performed jointly for both axes.

In classifying $\hat{A}^h$, we can use side information in the form of positive constraints for measurements which must belong to the same pitch-state, and negative constraint for measurements which must be classified into different pitch-states, as described in the following.

### Formulating Positive Constraints

Positive constraints allow us to mitigate measurement noise by relaxing the element-wise clustering problem, and instead clustering sets of measurements. Since the vessel pitch angle is continuous, we limit positive constraints to consecutive measurements, $\hat{a}_n^h, \hat{a}_{n+1}^h, \ldots, \hat{a}_{n+\varepsilon}^h$ of similar values, where $\varepsilon$ is limited. To formulate positive constraint, let $V_{\text{pos}}$ be a predefined multiplication of the sensitivity level of the accelerometer device being used. Also let $T_p$ be a pitch-coherence time used as an
upper bound for the time period over which positive constraints can be defined for
two acceleration measurements. Define relation $a_n \leftrightarrow a_{n'}$ if
\[
|a_{n,x} - a_{n',x}| < V_{pos} \quad (5.7a)
\]
\[
|a_{n,z} - a_{n',z}| < V_{pos} \quad (5.7b)
\]
\[
|t_n - t_{n'}| \leq T_p \quad , \quad (5.7c)
\]
otherwise $a_n \not\leftrightarrow a_{n'}$. We classify measurements $\hat{a}_n^h$ and $\hat{a}_{n'}^h$ into the same pitch-state if $\hat{a}_n^h \leftrightarrow \hat{a}_{n'}^h$, and if there exist no element $\hat{a}_j^h$, such that $t_n < t_{end} < t_{n'}$ and $\hat{a}_n^h \leftrightarrow \hat{a}_j^h$ or $\hat{a}_{n'}^h \leftrightarrow \hat{a}_j^h$. Since the vessel pitch angle is affected by ocean waves, we set $T_p = \frac{1}{2f_{wave}}$, where $f_{wave}$ is the fundamental frequency of the ocean waves, which can be estimated using, e.g. cyclostationary analysis (cf. [124]). Positive constraints are used to form matrices $\Delta_l$, $l = 1, \ldots, L$, each including a chain of consecutive measurements which pair-wise are classified into the same pitch-state.

The above definition of positive constrains guarantees that $\Delta_l$ are disjoint matrices. We obtain $\Delta_l$ by grouping consecutive acceleration measurements satisfying the above constraints in disjoint matrices such that a new matrix is formed every time positive constraints are not met for two consecutive measurements. For each vector $\Delta_l$ we assign classifier $\delta_l = \{1, \ldots, M\}$ such that all elements in $\Delta_l$ are classified into the same $\delta_l$ pitch-state. To this end, we reduce the problem of classifying $\hat{a}_n^h \in \mathcal{A}^h$ into classifying $\Delta_l$.

**Formulating Negative Constraints**

Negative constraints reflect the expected change in the vessel pitch angle. Since the last element of $\Delta_l$ and the first element of $\Delta_{l+1}$ do not satisfy (5.7), negative constraints are formed as the complementary of positive constraints. More specifically, negative constraints ensure that two consecutive subsets $\Delta_l$ and $\Delta_{l+1}$ cannot share the same pitch-state. Interestingly, since the rank of vectors $\Delta_l$, $l = 1, \ldots, L$ is limited by $T_p$, if $T_p < \tilde{T}_c$ (recall $\tilde{T}_c$ is the duration of each time slot) then negative constraints renders $L \geq 2$. We represent negative constraints by operator
\[
\eta_{\delta_l, \delta_{l+1}} = \begin{cases} 1 & \delta_l = \delta_{l+1} \\ 0 & \text{otherwise} \end{cases}, \quad l = 1, \ldots, L \quad . \quad (5.8)
\]

Shental [125] suggested formalizing negative constraints by a directed Markov graph, where in our case classifier $\delta_l$ is a hidden (decision) node, connected to both $\delta_{l-1}$ and $\delta_{l+1}$, as well as to an observation node, $\hat{a}_n^h \in \Delta_l$. Since the pitch angle associated with $\Delta_{l+1}$ depends on the pitch angle associated with $\Delta_l$, hidden nodes form a directed chain from parent, $\delta_l$, to child, $\delta_{l+1}$, and from hidden nodes to observation nodes, as illustrated in Figure 5.2. This directed chain is a form of a Bayesian belief
network [126], where for \( Y \) being the set of classifiers \( \delta_l, l = 1, \ldots, L \) (with \( \delta_0 = \emptyset \)), we obtain the joint distribution

\[
P(Y|\hat{A}_h, \Theta) = \prod_{l=1}^{L} P(\delta_l|\delta_{l-1}, \Theta, \hat{A}_h).
\] (5.9)

The relation in (5.9) would be used to formulate the likelihood function to determine \( \delta_l \) using a constraint EM algorithm, as described next.

Formulating the Likelihood Function

Considering both positive and negative constraints, the data likelihood function for the event of correct classification of the elements of \( \hat{A}_h \) is

\[
P(\hat{A}_h, Y|\Theta) = \frac{1}{c_1} \prod_{l=1}^{L-1} (1 - \eta_{\delta_l,\delta_{l+1}}) \prod_{l=1}^{L} \prod_{\hat{a}_{n,x}^h \in \Delta_l} P(\delta_l|\Theta)P(\hat{a}_{n,x}^h|\delta_l, \Theta)P(\hat{a}_{n,z}^h|\delta_l, \Theta),
\] (5.10)

where \( c_1 = \sum_{\delta_1=1}^{\hat{M}} \cdots \sum_{\delta_L=1}^{\hat{M}} \prod_{l=1}^{L-1} (1 - \eta_{\delta_l,\delta_{l+1}}) \prod_{l=1}^{L} P(\delta_l|\Theta). \)

For classifying subsets \( \Delta_l \), we use the EM algorithm to iteratively maximize (5.10). In the \( p \)th iteration, we obtain estimation \( \hat{\Theta}^p \) of \( \Theta \), with elements \( \hat{\mu}_{m,x}^p, \hat{\mu}_{m,z}^p, \hat{\sigma}_{m,x}^p, \hat{\sigma}_{m,z}^p, \) and \( \hat{k}_m^p, m = 1, \ldots, \hat{M} \). For the Gaussian mixture model (5.2), given \( \hat{\Theta}^p \) we
obtain [91]

\[
\tilde{\mu}_{m,\omega}^{p+1} = \frac{\sum_{l=1}^{L} P(\delta_l = m|\hat{A}^h, \tilde{\Theta}^p) \sum_{\hat{a}_{n,\omega} \in \Delta_l} \hat{a}_{n,\omega}^h}{\sum_{l=1}^{L} P(\delta_l = m|\hat{A}^h, \tilde{\Theta}^p)}, \quad \omega \in \{x, z\} \tag{5.11a}
\]

\[
\tilde{\sigma}_{m,\omega}^{p+1} = \frac{\sum_{l=1}^{L} P(\delta_l = m|\hat{A}^h, \tilde{\Theta}^p) \sum_{\hat{a}_{n,\omega} \in \Delta_l} (\hat{a}_{n,\omega}^h - \tilde{\mu}_{m,\omega}^{p+1})^2}{\sum_{l=1}^{L} P(\delta_l = m|\hat{A}^h, \tilde{\Theta}^p)}, \quad \omega \in \{x, z\} \tag{5.11b}
\]

Furthermore,

\[
\tilde{k}_{p+1}^{1}, \ldots, \tilde{k}_{p+1}^{\tilde{M}} = \arg\max_{k_1, \ldots, k_{\tilde{M}}} \log \left( \frac{1}{c_1} \right) + \sum_{m=1}^{\tilde{M}} \log(k_m) \sum_{l=1}^{L} P(\delta_l = m|\hat{A}^h, \hat{A}^z, \tilde{\Theta}^p) \tag{5.12a}
\]

\[
\text{s.t. } \sum_{m=1}^{\tilde{M}} k_m = 1 \tag{5.12b}
\]

Since \(c_1\) is a function of \(k_m\) (recall \(k_m = P(\delta_l = m|\tilde{\Theta}^p)\)), it is hard to maximize (5.12). Instead, we approximate \(c_1\). Assuming negative constraints are mutually exclusive, \(c_1 \approx (1 - \sum_{j=1}^{\tilde{M}} k_j^2)^L\) [125]. Clearly, this approximation does not hold in our case. However, since our Markov graph is sparse, this assumption has little effect on performance (as our numerical simulations show). From (5.12) and using a lagrange multiplier, we get

\[
P(\delta_l = m|\hat{A}_x^h, \hat{A}_z^h, \tilde{\Theta}^p) \left( 1 - \frac{\sum_{j=1}^{\tilde{M}} (\tilde{k}_j^{p+1})^2}{\tilde{k}_m^{p+1}} \right) - 2(L-1)\tilde{k}_m^{p+1} + (2L-1)\sum_{j=1}^{\tilde{M}} (\tilde{k}_j^{p+1})^2 - 1 = 0 \tag{5.13}
\]

The expression in (5.13) can be solved for \(\tilde{k}_m^{p+1}\) either by approximating \(\sum_{j=1}^{\tilde{M}} (\tilde{k}_j^{p+1})^2\) as \(\sum_{j=1}^{\tilde{M}} (\tilde{k}_j^p)^2\) (and \(\tilde{k}_j^p\) is known from previous iteration), or using alternating optimization technique (cf. [120]). Equations (5.11a), (5.11b), and (5.13), are used to determine \(\tilde{\Theta}^{p+1}\), which in turn is used for the next EM iteration till convergence of (5.10) is reached\textsuperscript{12}.

\textsuperscript{12}Note that the EM algorithm is proven to converge to a local maxima of the log-likelihood function [91].
In the last EM iteration, $\rho_{\text{last}}$, we determine classifiers $\delta_l$, $l = 1, \ldots, L$ of vectors $\Delta_l$ by numerically solving
\[
\delta_l = \arg\max_{\delta_l} \left[ P(\delta_l = m|\hat{A}_x^h, \hat{A}_z^h, \Theta^{\text{last}}) \right], \quad (5.14)
\]
and construct matrix $\hat{A}_m^h$, $m = 1, \ldots, M$ including elements $\hat{a}_n^h$ for which $\delta_l = m$ and $\rho_n = \rho_m$.

Considering model (5.1), to project the elements in matrix $\hat{A}_m^h$ project onto $\hat{A}_m^{h,p}$, we estimate a solution $\tilde{\rho}_m$ by multiplying $\hat{a}_n^h \in \hat{A}_m^h$, $n = 1, \ldots, N$ with the rotation matrix
\[
\begin{bmatrix}
\cos \tilde{\rho}_m & 0 & -\sin \tilde{\rho}_m \\
0 & 1 & 0 \\
\sin \tilde{\rho}_m & 0 & \cos \tilde{\rho}_m
\end{bmatrix}, \quad (5.15)
\]
Observing model (5.1), we note that when the noise vector $e_n$ is zero and heading compensation is ideal, multiplying $\hat{a}_n^h \in \hat{A}_m^h$ by the matrix in (5.15) gives the vector
\[
\begin{bmatrix}
a_{n,x}^h \cos (\tilde{\rho}_m - \rho_m) + a_{n,z}^h \sin (\rho_m - \tilde{\rho}_m) \\
0 \\
a_{n,x}^h \sin (\tilde{\rho}_m - \rho_m) + a_{n,z}^h \cos (\rho_m - \tilde{\rho}_m)
\end{bmatrix}. \quad (5.16)
\]
Since $a_{n,z}$ can be non-zero, $\tilde{\rho}_m$ cannot be estimated following the same approach as in (5.4). Unfortunately, (5.16) sets a degree of freedom in choosing $\tilde{\rho}_m$. Instead, since we assumed $a_{n,x} > a_{n,z}$ (see Section 5.1), the term $a_{n,x} \cos (\tilde{\rho}_m - \rho_m) + \sin (\rho_m - \tilde{\rho}_m)$ is approximated by $a_{n,x}^h \tilde{\rho}_m + a_{n,z}^h (\tilde{\rho}_m)$. Since $a_{n,x}^h (\tilde{\rho}_m) + a_{n,z}^h (\tilde{\rho}_m)$ achieves its maximum for $\tilde{\rho}_m - \rho_m = \frac{\pi}{4}$ when $a_{n,x} > 0$, and for $\tilde{\rho}_m - \rho_m = \frac{\pi}{4} + \pi$ when $a_{n,x} < 0$, we set
\[
\tilde{\rho}_{m,1} = \arg\max_{\tilde{\rho}_m} \sum_{n: \hat{a}_n^h \in \hat{A}_m^h} a_{n,x}^h (\tilde{\rho}_m) + a_{n,z}^h (\tilde{\rho}_m) - \frac{\pi}{4}
\]
\[
\tilde{\rho}_{m,2} = \arg\max_{\tilde{\rho}_m} \sum_{n: \hat{a}_n^h \in \hat{A}_m^h} a_{n,x}^h (\tilde{\rho}_m) + a_{n,z}^h (\tilde{\rho}_m) - \frac{\pi}{4} - \pi. \quad (5.17)
\]
From the two solutions in (5.17), we choose the one satisfying condition $-\frac{\pi}{4} < \tilde{\rho}_m < \frac{\pi}{4}$, and using (5.16) form projection $a_{n,x}^h$, considered aligned with the horizontal plane along the vessel heading direction. Finally, we combine projected measurements from all time slots and pitch-states in matrix $\hat{A}_m^{h,p}$.

Due to the dependency between hidden nodes $\delta_l$, the difficulty in (5.11), (5.13), and (5.14) lies in calculating $P(\delta_l = m|\hat{A}_x^h, \hat{A}_z^h, \Theta^{p})$. We next describe an efficient way to obtain the posterior probabilities.
Finding the Posterior Probability

Finding the posterior probability is simple when only positive constraints are imposed. Here, \( P(\delta_t = m|\mathcal{A}^h, \tilde{\Theta}^p) = P(\delta_t = m|\Delta_t, \tilde{\Theta}^p) \), and since measurements in \( \mathcal{A}^h \) are assumed independent (see (5.2)),

\[
P(\delta_t = m|\Delta_t, \tilde{\Theta}^p) = \frac{\bar{k}_m^p P(\Delta_t|\omega_m^p) P(\hat{a}_{n,x}^h|\omega_m^p) P(\hat{a}_{n,z}^h|\omega_m^p)}{\sum_{j=1}^{M} \bar{k}_j^p \prod_{\hat{a}_{n,x}^h \in \Delta_t} P(\hat{a}_{n,x}^h|\omega_j^p) P(\hat{a}_{n,z}^h|\omega_j^p)}. \tag{5.18}
\]

However, at the presence of negative constraints, \( \delta_t \) depends on other hidden nodes, and

\[
P(\mathcal{Y}|\mathcal{A}^h, \tilde{\Theta}^p) = \frac{\prod_{l=1}^{L-1} (1 - \eta_{\delta_l,\delta_{l+1}}) \prod_{l=1}^{L} \prod_{\hat{a}_{n,x}^h \in \Delta_t} P(\delta_l|\hat{a}_{n,x}^h, \tilde{\Theta}^p) P(\delta_l|\hat{a}_{n,z}^h, \tilde{\Theta}^p)}{c_2}, \tag{5.19}
\]

where

\[
c_2 = \sum_{\delta_1=1}^{\hat{M}} \cdots \sum_{\delta_L=1}^{\hat{M}} (1 - \eta_{\delta_l,\delta_{l+1}}) \prod_{l=1}^{L} \prod_{\hat{a}_{n,x}^h \in \Delta_t} P(\delta_l|\hat{a}_{n,x}^h, \tilde{\Theta}^p) P(\delta_l|\hat{a}_{n,z}^h, \tilde{\Theta}^p).
\]

Here, the posterior probability can be obtained by marginalizing the joint probability (5.19). However, in general, this is an NP-hard problem (with similarities to the graph coloring problem) with complexity \( \mathcal{O} \left( \hat{M}^{L-1} \right) \) [126]. Instead, since the Markov graph illustrated in Figure 5.2 is a directed chain, belief propagation techniques can be used to significantly reduce complexity to \( \mathcal{O} \left( LM^2 \right) \) [126]. The general idea in belief propagation is based on observation (5.9), that assuming \( P(\delta_l|\delta_{l-1}) \) is known, belief \( \text{BEL}_{\delta_l}(m) = P(\delta_l|\mathcal{A}^h, \tilde{\Theta}^p) \) can be exactly calculated by receiving belief \( \text{BEL}_{\delta_{l-1}}(m) \) from parent \( \delta_{l-1} \) and likelihood \( P(\mathcal{A}^h|\delta_{l-1}, \tilde{\Theta}^p) \) from child \( \delta_{l+1} \). Following [125], we use Pearl’s belief propagation algorithm for trees [126], as presented in the following.

In Pearl’s algorithm, each hidden node \( l \) exchanges ”messages” \( \lambda_{\delta_l}(m) \) and \( \pi_{\delta_l}(m) \) with his parents and children, respectively. We start by initializing lists \( \pi_{\delta_l}(m) = \lambda_{\delta_l}(m) = 1, \forall l, m, \ l = 1, \ldots, L, \ m = 1, \ldots, \hat{M} \). Upon activation, each hidden node, \( \delta_l \), receives \( \pi_{\delta_{l-1}}(m) \) and \( \lambda_{\delta_{l+1}}(m) \) from its parent and child, respectively. It then calculates

\[
\lambda_{\delta_l}(m) = \sum_{n=1}^{\hat{M}} \lambda_{\delta_{l+1}}(n) \prod_{\hat{a}_{n,x}^h \in \Delta_l} P \left( \delta_l = n|\delta_{l-1} = m, \hat{a}_{n,x}^h, \hat{a}_{n,z}^h, \tilde{\Theta}^p \right) \tag{5.20a}
\]

\[
\pi_{\delta_l}(m) = \sum_{n=1}^{\hat{M}} \pi_{\delta_{l-1}}(n) \prod_{\hat{a}_{n,x}^h \in \Delta_l} P \left( \delta_l = m|\delta_{l-1} = n, \hat{a}_{n,x}^h, \hat{a}_{n,z}^h, \tilde{\Theta}^p \right). \tag{5.20b}
\]
where $\pi_{\delta_0}(m) = 1$ (recall $\delta_0 = \emptyset$), and sets
\[
\text{BEL}_{\delta_l}(m) = \frac{\lambda_{\delta_{l+1}}(m) \pi_{\delta_l}(m)}{\sum_{j=1}^{M} \text{BEL}_{\delta_l}(j)}.
\]
(5.21)

Next, hidden node $\delta_l$ sends "message" $\frac{\pi_{\delta_l}(m)}{\sum_{j=1}^{M} \pi_{\delta_l}(j)}$ to its child, $\delta_{l+1}$, and $\frac{\lambda_{\delta_l}(m)}{\sum_{j=1}^{M} \lambda_{\delta_l}(j)}$ to its parent, $\delta_{l-1}$. Since our network is a directed Markov chain, we execute (5.20) and (5.21) for the sequence $\delta_1 \rightarrow \delta_2 \cdots \rightarrow \delta_L$, and reach convergence after $L$ steps.

**Initial Estimation $\tilde{\Theta}^0$**

The EM algorithm in (5.11) and (5.12) requires initialization of $\tilde{\Theta}^0$. With no prior knowledge of the effect of ocean waves on acceleration, we use the K-means clustering algorithm (cf. [91]) to initially classify elements in each time slot into $\tilde{M}$ pitch-states and form matrixes $\hat{A}^h_m$. Similar to the EM algorithm, the K-means algorithm is executed jointly for $\hat{A}^h_x$ and $\hat{A}^h_z$. After this initial classification, based on the statistical mean and standard deviation of elements $\hat{a}^h_{n,x}, \hat{a}^h_{n,z} \in \hat{A}^h_m$, we calculate $\tilde{\mu}_m^0, \tilde{\sigma}_m^0$, and $\tilde{k}_m^0$, $m = 1, \ldots, M$, respectively, and $\tilde{k}_m^0$ is estimated as the ratio between rank $|\hat{A}^h_m|$ and the number of elements in each time slot.

**5.2.3 Distance Estimation**

After projecting vector $\hat{A}$ into $\hat{A}^{h,p}$ with elements $\hat{a}^{h,p}_{n,x}, n = 1, \ldots, N$, considered aligned with the vessel heading direction (see Section 5.2.1) and horizontal plane (see Section 5.2.2), we can readily estimate the distance traveled by the vessel at time period $[t^{\text{start}}, t^{\text{end}}]$, $d_{i,j}$. Given the initial velocity $v_i$ at time instance $t^{\text{start}}$, we obtain estimation $\tilde{d}_{i,j}$ by integrating the projected acceleration measurements over the period $[t^{\text{start}}, t^{\text{end}}]$. First, we obtain the mean velocity by,
\[
\hat{v}_{i,j} = v_i + \frac{1}{2} \sum_{n=2}^{N} \hat{a}^{h,p}_{n,x}(t_n - t_{n-1}) .
\]
(5.22)

Then, we set $\tilde{d}_{i,j} = \hat{v}_{i,j}(t^{\text{end}} - t^{\text{start}})$.

**5.2.4 Summarizing the Operation of the DR-A Method**

We now summarize the operation of our DR-A method. Referring to the pseudo-code in Algorithm 3, we first form vectors $\hat{A}^w$, $w = 1, \ldots, W$ for $W = \left\lceil \frac{t^{\text{end}} - t^{\text{start}}}{T_c} \right\rceil$ time
slots (line 1). For each wth time slot, we estimate $\alpha_n$ and $\omega_w$, and project acceleration measurements in $\hat{A}^w$ into $\hat{A}^{h,w}$ (line 3). To assist classification of pitch-states, we set positive and negative constraints to form subsets $\Delta_l$ and $\Lambda_l$, respectively (lines 4-11). Next, we perform initial classification (line 12) to calculate $\tilde{\Theta}^0$ (line 13), and perform $p_{\text{last}}$ EM iterations to classify $\Delta_l$ (lines 14-18). For each wth time slot and $m$th pitch-state, we project elements in matrix $\hat{A}^{h,w}_m$ to compensate the vessel pitch angle (line 19), and group the projected elements to form vector $\hat{A}^{h,p}$ (line 21). Finally, we evaluate distance $d_{i,j}$ (line 22).

**Algorithm 3** Evaluate $\tilde{d}_{i,j}$ from vector $\hat{A}$

1: Divide $\hat{A}$ into $W$ time slots $\hat{A}^w$
2: for $w := 1$ to $W$ do \{do for each time slot\}
3: Estimate $\alpha_n$ using (5.4) and $\omega_w$ using (5.6), and project $\hat{A}^w$ into $\hat{A}^{h,w}$ using (5.5)
4: $l := 1$, $\Delta_l \leftarrow \hat{a}^{h,w}_1$, $\Lambda_l := \emptyset$
5: for $n := 1$ to $|\hat{A}^{h,w}_m| - 1$ do
6: if $\hat{a}^{h,w}_{n,x} \in \hat{A}^{h,w}_m$ and $\hat{a}^{h,w}_{n+1,x} \in \hat{A}^{h,w}_m$ satisfy positive constraints then
7: $\Delta_l \leftarrow \hat{a}^{h,w}_{n+1}$
8: else
9: $l := l + 1$, $\Delta_l \leftarrow \hat{a}^{h,w}_{n+1}$, $\Lambda_{l-1} \leftarrow (n, n + 1)$
10: end if
11: end for
12: classify $\hat{A}^{h,w}$ using the K-means algorithm to initially form matrixes $\hat{A}^{h,w}_m$
13: $\mu_{m,x}^0 := E[\hat{A}^{h,w}_{m,x}], \mu_{m,z}^0 := E[\hat{A}^{h,w}_{m,z}], \sigma_{m,x}^0 = E[(\hat{A}^{h,w}_{m,x} - \mu_{m,x}^0)^2], \sigma_{m,z}^0 = E[(\hat{A}^{h,w}_{m,z} - \mu_{m,z}^0)^2], k_m^n = |\hat{A}^{h,w}_m|, \text{initial estimation}$
14: for $p := 0$ to $p_{\text{last}} - 1$ do \{EM iterations\}
15: Iteratively calculate (5.21) to obtain $P(\delta_l = m | \hat{A}^{h,w}, \tilde{\Theta}^p)$
16: Solve (5.11) and (5.13) to obtain $\tilde{\Theta}^{p+1}$
17: end for
18: Classify subsets $\Delta_l$ using (5.14) and form $\hat{A}^{h,w}_m$, $m = 1, \ldots, \tilde{M}$
19: Project elements in matrixes $\hat{A}^{h,w}_m$ using (5.17)
20: end for
21: group all projected measurements to form vector $\hat{A}^{h,p}$
22: Evaluate $\tilde{d}_{i,j}$ using (5.22)

The DR-A algorithm is designed for the case where an estimate of the distance traveled by the vessel in the last $t_{\text{end}} - t_{\text{start}}$ seconds is required. However, a modification can be made to incorporate such recursive operation by re-estimating the distance and heading once a newly single or a small number of acceleration measurements are acquired. Here, the summations and multiplications in (5.4), (5.9), (5.11)
and (5.20) are stored and recalculated only for the newly acquired measurements.

5.2.5 Discussion

Time-varying Roll Angle

In this chapter, we specifically assumed the vessel roll angle is fixed. However, due to ocean waves, in certain cases the roll angle can also be time-varying. While the roll angle is expected to change much slower than the pitch angle, the former may still change within a single time slot. As a result, our assumption that (per time slot) changes in acceleration are only due to the time-varying pitch angle, does not always hold and classification to pitch-states cannot be made. A possible solution to this problem would be to identify the frequency of change of the vessel roll angle and to define shorter time slots for which both acceleration in the horizontal plane and vessel roll angle can be considered fixed. Classification of acceleration measurements is then performed for each of these (shorter) time slots. Since changes in the vessel roll angle also affects acceleration in the projected $y$-axis (which otherwise would be close to zero), the rate of change in the vessel roll angle can be estimated by observing periodic changes in the $y$ component of $\hat{A}$. Clearly, this approach introduces more noise in estimating $d_{i,j}$ since classification is based on fewer measurements.

Complexity

DR navigation is a task performed online. Thus, complexity is of interest. When a gyrocompass is used and both $\alpha$ and $\rho_m$ are directly measured, DR involves multiplications (5.5) and (5.15), and after accumulating $N$ measurements, equation (5.22) is executed. In our case, the heading angle is estimated through (5.4), and the pitch angles by solving (5.21), (5.11) and (5.13). Referring to the discussion in Section 5.2.2, the above procedure is performed for each $p$th EM iteration and $w$th time slot. Hence, the overhead complexity of the DR-A algorithm over using a gyrocompass is $O(LM^2p_{\text{last}}W)$, where $W = \lceil \frac{t_{\text{end}} - t_{\text{start}}}{T_c} \rceil$ and $L \leq \frac{N}{W}$. In our numerical simulations and sea trial, convergence of the EM algorithm was reached after roughly $p_{\text{last}} = 10$ iterations, and we used $M \sim 10$. Using an Intel Core Duo CPU with a 2.66 GHz processor, this allowed a processing time of less than a second.

Choosing the number of pitch-states $M$

The EM algorithm requires a pre-defined number of states, $M$. As the wave height and its effect on the vessel pitch angle are hard to evaluate prior to data collection, determining $M$ in an optimized fashion is a difficult task, which is beyond the scope of this work. However, an educated guess can be made using tree decision algorithms to determine the number of pitch-states as the one that maximises the amount of available information, i.e., the entropy [127]. This is evaluated using the observation
that projection accuracy increases with $M$, but as $M$ increases less information is available per-state and classification performance decreases. We next explore these tradeoffs.

5.3 Performance Evaluation

We now evaluate the performance of our DR-A algorithm. The results are presented in terms of $\rho_{err} = |\tilde{d}_{i,j} - d_{i,j}|$ and $\rho_{angle} = |\tilde{\alpha}_n - \alpha_n|$. Since the DR-A method is based on the Gaussian mixture model (5.2) as well as on the assumption that per-time slot acceleration in the horizontal plane, vessel heading direction, and vessel roll angle are fixed, to validate simulation results we also present results based on data collected from a real sea environment.

5.3.1 Simulation Results

Our simulation setting includes a Monte-Carlo set of 10000 channel realizations. In each simulation, a vessel moves for 60 seconds in the $x - y$ plane with initial speed uniformly distributed between $[0, 5]$ m/seconds. The vector of acceleration in the horizontal plane, $a_{n,x}, n = 1, \ldots, N$, is sampled at rate 0.1 seconds (i.e., $N = 600$), and generated as zero-mean colored Gaussian process with standard deviation of $1 \text{ m}/(\text{seconds})^2$ and a cross-correlation factor between adjacent samples of $e^{-\frac{t}{T_c}}$, where $T_c = 6$ seconds. Likewise, the vessel heading, $\angle_n$ is generated uniformly between $[0, 2\pi]$ with a similar cross-correlation factor between adjacent samples. Furthermore, for every simulation trial, the (fixed) vessel orientation angle with respect to the reference coordinate system is uniformly randomized within the interval $[0, 2\pi]$. Based on the latter and $\angle_n$, we form vector $\alpha_n, n = 1, \ldots, N$. Note that the initial speed and heading direction are known.

Let $h(x, y, t)$ be the time domain $t$ function for the height of the sea surface for a modeled three-dimensional ocean wave in the $x - y$ plane. To simulate wave-based acceleration in the vertical plane, for $t_n$ being exactly in the middle of the time period before a local maxima and after a local minima of $h(x, y, t)$, we uniformly randomize sample $a_{n,z}$ between $[0.01, 0.05]$ g. Similarly, $a_{n',z}$ is generated uniformly between $[-0.05, -0.01]$ g for $t_{n'}$ exactly in between a local maxima and a local minima. Acceleration then changes linearly with time such that it reaches zero at both local maxima and minima of $h(x, y, t)$. Since the derivative of $h(x, y, t)$ is also the slope of the wave surface, the pitch angle $\rho_n$ at time sample $t_n$ and coordinates $(x_n, y_n)$ is computed from $\tan \rho_n = \frac{\partial w(x, y, t_n)}{\partial x_n}$. Using $\alpha_n$ and $\rho_n$ and based on model (5.1), we form the vector of acceleration measurements, $\hat{\mathbf{A}}$.

Current literature offers multiple models for the wind-based ocean surface wave function $h(x, y, t)$ (e.g., [128, 129, 130],). In our simulations we use the analytical wave model offered in [130]. We note that similar results were obtained also for other
wave models. For the $i$th wave frequency and $j$th directional angle, let $\theta_i$, $\varphi_i$, and $\psi_{i,j}$ be the spreading directional angle, wave frequency, and an initial phase angle, respectively. The wave height is modeled as

$$h(x, y, t) = \sum_{i=1}^{I} \sum_{j=1}^{J} \sqrt{2S(\varphi_i)\Delta\varphi \Delta\theta_j \cos(c_i x \cos \theta_j + c_i y \sin \theta_j - \varphi_i t - \psi_{i,j})}, \quad (5.23)$$

where $c_i = \frac{\varphi_i^2}{g}$ is the wave number, $\Delta\varphi$ and $\Delta\theta$ are the increments of angles $\varphi$ and $\theta$, respectively, and we use the directional spectrum function $S(\varphi) = b_1 \frac{g^2}{\varphi^5} \exp \left[ -b_2 \left( \frac{g}{U\varphi} \right)^4 \right]$ with $b_1 = 8.1 \times 10^{-3}$, $b_2 = 0.74$, and a wind speed $U = 5$ knots. For each channel realization, parameters $\varphi_i$, $\theta_i$, and $\psi_{i,j}$ are uniformly randomized in intervals $[1, 5]$ rad/seconds, $[0, 2\pi]$ rad, and $[0, 2\pi]$ rad, respectively, and we use $I = 10$, $J = 10$, $\Delta\varphi = 0.4$ rad/seconds, and $\Delta\theta = \frac{\pi}{5}$ rad. An example for $h(x, y, t)$ for $t = 10$ seconds is shown in Figure 5.3.

Since current methods for DR navigation use both acceleration and orientation measurements, we benchmark our algorithm by showing results of $\rho_{err}$ when an ideal gyroscope is used, i.e., perfect compensation of the vessel pitch and heading angles (Ideal-Gyro), and when orientation measurements are noisy (Noisy-Gyro). In addition, we compare results of DR-A with a direct integration of acceleration measurements with no heading and pitch compensation (Naive), and with an alternative method in which after heading estimation, using the PCA method, per time slot we obtain measurements aligned with the horizontal plane (PCA). We also demonstrate tradeoffs between complexity and performance by replacing the constraint EM algorithm with i) the initial K-means classifier ($K$-means), ii) a simple slicing classifier ($Slice$), and iii) non-constraint EM ($NC-EM$). For the Ideal-Gyro and Noisy-Gyro methods, we estimate $a_{n,x}$ by multiplying $\hat{a}_n$ with matrices (5.5) and (5.15) for the measured heading and pitch angles, $\hat{\alpha}_{i,j}$ and $\hat{\rho}_n$, respectively. For the former we use
\[ \hat{\rho}_n = \rho_n \text{ and } \hat{\alpha}_n = \alpha_n, \text{ while for the latter we set } \hat{\rho}_n = \rho_n + \bar{\epsilon}_n \text{ and } \hat{\alpha}_n = \alpha_n + \bar{\epsilon}_n, \text{ where } \bar{\epsilon}_n \text{ is a zero-mean Gaussian noise with variance } \varsigma \text{ rad}^2, \text{ which is the same variance considered for the acceleration measurement noise. In the following, results for the DR-A method are shown for } p_{\text{last}} = 10 \text{ EM iterations and using } V_{\text{pos}} = 0.05 \text{ m/seconds}^2 \text{ for the positive constraints in (5.7).} \]

In Figure 5.4, we show the complementary cumulative density function (C-CDF) of \( \rho_{\text{angle}} \) using estimation (5.4) for different values of \( \frac{1}{\varsigma} \). We observe that reasonable performance are obtained for relatively low noise values (around 20 dB). Next, we show tradeoffs of performance vs. the number of pitch-states, \( \tilde{M} \), and the assumed coherence time, \( \tilde{T}_c \). In Figure 5.5a we show performance of the four considered classifying method in terms of \( \rho_{\text{err}} \) for \( \tilde{M} \in \{2, 10, 20\} \) and \( \frac{1}{\varsigma} = 20 \text{ dB} \). We observe that for the Slice method performance are not linear with \( \tilde{M} \). This is due to the fact that in the Slice method measurements are classified to all assumed \( \tilde{M} \) pitch-states, and thus, for each pitch-state, fewer measurements are available affecting accuracy of estimation (5.17). While accuracy increases with \( \tilde{M} \) for the DR-A, NC-EM, and Kmeans (which allow empty pitch-states), we observe that little is gained for \( \tilde{M} > 10 \).

Next, for \( \tilde{M} = 10 \), in Figure 5.5b we show \( \rho_{\text{err}} \) as a function of the assumed coherence time, \( \tilde{T}_c \), which is used to obtain \( W \) time slices of assumed fixed acceleration in the horizontal plane. Here we observe a slight performance degradation for mismatch coherence time (i.e., when \( \tilde{T}_c \neq T_c = 6 \text{ seconds} \)). When \( \tilde{T}_c < T_c \), this is because fewer measurements are available to estimate the pitch angle in (5.17), while for \( \tilde{T}_c > T_c \), performance degrade since our assumption of fixed acceleration in the horizontal plane (required for classification) does not hold. Interestingly, we observe that performance are less affected in the latter case. That is, having enough statistics to estimate the pitch angle is more important, as was also observed for the Slice method in Figure 5.5a. In the following we use \( \tilde{M} = 10 \) and \( \tilde{T}_c = T_c \).

In Figure 5.6, we show average results of \( \rho_{\text{err}} \) for the considered classification
methods and the benchmark methods as a function of $\frac{1}{\varsigma}$. We observe a significant performance degradation for the Noisy-Gyro method compared to the Ideal-Gyro one. We consider this performance gap as the maximal gain available for using our method. We also observe that for the Noisy-Gyro method, performance is not linear with $\frac{1}{\varsigma}$ (in the logarithmic scale). This is because the noisy orientation measurements introduce non-Gaussian noise to the projected acceleration measurements. As expected, performance for the Naive approach is poor, and is in fact fixed for different noise values. The latter is due to the periodic nature of the vessel pitch angle, which averages out positive and negative acceleration measurements in the $x$ axis. Comparing the performance of the PCA method to those of our method, we observe that PCA is better than the Slice method, mostly due to the naive classification performed in the Slice method which is largely affected by measurement noise. However, using the better classification capabilities of the EM and K-means algorithm, we observe considerable performance gain compared to the PCA method. This is due to the underlying assumption in PCA of the variance of acceleration measurements, which might not hold for all modeled ocean waves. As expected, performance of the EM algorithm, which is matched to the Gaussian mixture model in (5.2), outperforms that of the K-means algorithm, at a cost of complexity.\textsuperscript{13} Moreover, significant improvement is achieved using our DR-A method compared to that of the non-constraint EM algorithm, NC-EM. From Figure 5.6, we observe that the performance of our DR-A method is close to that of the (unrealistic) Ideal-Gyro method. Thus, we conclude that, without using orientation measurements, the DR-A method almost entirely compensates the vessel’s heading angle and time-varying pitch angle. To comment on the distribution

\textsuperscript{13}We note that both the EM and the K-means algorithms where processed in real-time.
Chapter 5. A Machine Learning Approach for Underwater DR

![Graph showing relationship between \(\rho_{\text{err}}\) and \(\frac{1}{\varsigma}\) with different noise levels.]

Figure 5.6: Results of \(\rho_{\text{err}}\) as a function of \(\frac{1}{\varsigma}\). \(\tilde{M} = 10, \tilde{T_c} = T_c\).

![Graph showing C-CDF results of \(\rho_{\text{err}}\) with \(\frac{1}{\varsigma} = 20\) dB.]

Figure 5.7: C-CDF results of \(\rho_{\text{err}}\). \(\frac{1}{\varsigma} = 20\) dB. \(\tilde{M} = 10, \tilde{T_c} = T_c\).

of the performance, in Figure 5.7 we show the C-CDF of \(\rho_{\text{err}}\) for \(\frac{1}{\varsigma} = 20\) dB. Results show that the above conclusions, drawn for the average \(\rho_{\text{err}}\) results, hold true for all simulated scenarios.

In the following we present results of offline processing of 3-D acceleration measurements obtained in a sea trial.

### 5.3.2 Sea-Trial Results

In this section, we describe results of our DR-A method obtained from real sea environment. The sea trial was conducted on Nov. 2011 in the Singapore Strait. The experiment lasted for two hours and included two boats. During the experiment, the waves height was about 0.5 m and the boats drifted with the ocean current, which was about 0.7 m/seconds. At each boat, we obtained around \(N = 33.5\) k 3-D acceleration measurements using the Libelium Wasp Mote’s on board accelerometer.
at rate of $f = 4.8$ Hz, and used its serial port for data logging purpose and offline decoding. Throughout the experiment, the location of the boats was monitored using GPS receivers at rate of 3 seconds and with expected accuracy of 5 m. The boats GPS-based location in the x and y axis are shown in Figures 5.8a and 5.8b, respectively. From the figures we observe that, as a result of the ocean current, the boats changed their heading direction. However, other than around time $t = 25$ min (where boat 2 had to maneuver around an obstacle), this heading change is slow and fits our underlying assumption. Furthermore, from the figures we observe a slow change in the boats’ speed (with variance of 0.1 m/seconds$^2$), which allows using time slices of assumed fixed acceleration in the horizontal plane. In Figure 5.9, we show the measured acceleration along the $x$ axis as a function of earth gravity, $g$ (the results for the projected accelerations are discussed further below). We observe that acceleration measurements follow a wave pattern, and that both frequency and amplitude of these waves are different for the two boats. The periodic nature of the measurements shown in Figure 5.9 emphasize the need for compensating on the vessel pitch angle, as direct orientation measurements may be too noisy. Moreover, due to the observed fast time-varying measurements (caused by ripples), mitigating the effect of the vessel’s pitch angle using filtering is not possible without loss of resolution.

First, in Figure 5.10a we show C-CDF performance of the heading estimate for the two boats, where the vessel heading direction with respect to the UTM coordinate system is calculated based on the boats’ GPS-based location. Results are compared for consecutive time slots of duration $T_{\text{slot}}$ such that $\frac{N}{T_{\text{slot}}}$ estimations of $\rho_{\text{angle}}$ are obtained. Since accuracy of estimation (5.4) improves with the number of measurements but decreases if the heading angle changes within the time slot, results tradeoff for $T_{\text{slot}}$. We observe for all cases best results are obtained for $T_{\text{slot}} = 200$ seconds, where the error is below 0.02 rad in more than 90% of cases. Due to the sensitivity
Figure 5.9: Sea trial: measured and projected acceleration along the $x$ axis relative to $g$ for $\bar{M} = 5$ and $\bar{T}_c = 40$ seconds. Boat 1.

Figure 5.10: Sea trial: (a) C-CDF results of $\rho_{\alpha}$ for the two boats, (b) $\rho_{\alpha}$ vs. the accumulated change of the boat’s heading angle within the time slot, $T_{\text{slot}} = 300$ seconds.
of our scheme to changes of the vessels’ heading angle within the time slot, further increase of $T_{\text{slot}}$ reduces accuracy. This is shown in Figure 5.10b, where $\rho_{\text{err}}$ is presented as a function of the accumulated change of the boat’s heading angle within the time slot for $T_{\text{mathrmslot}} = 300$ seconds. For both vessels we observe a considerable increase in $\rho_{\text{err}}$ whenever the accumulated change in heading direction is more than 5 degrees. A possible way to limit such inaccuracies is to detect significant heading changes using e.g. a compass, and to set uneven time slots such that the heading angle is roughly fixed within each time slot.

In Figure 5.9, for the first 50 seconds of the experiment, we show the measured and projected acceleration along the $x$ axis, where the latter are obtained using $\tilde{M} = 5$ and $\tilde{T}_c = 40$ sec. We observe that the projected measurements are almost constant, which matches the expected slow change of the boats’ acceleration in the horizontal plane. Finally, in Figure 5.11 we present C-CDF results of $\rho_{\text{err}}$, comparing estimation $\hat{d}_{i,j}$ to the GPS-based distance for time slots of duration $T_{\text{slot}} = 200$ sec and for different values of $\tilde{M}$ and $\tilde{T}_c$. For each time slot, the initial speed $v_i$ is calculated based on the first two GPS-based locations of the boats. We observe that in more than 96% of cases $\rho_{\text{err}}$ is lower than the expected error for $\tilde{M} = 5$ and $\tilde{T}_c = 40$ sec. Considering this result and the low estimation error observed for the heading angle, we conclude that when the boats’ heading is constant, our method fully compensates for the vessel’s time-varying pitch angle using only a single accelerometer device.

### 5.4 Conclusion

In this chapter, for DR navigation of a vessel whose motion is affected by the ocean waves, we proposed a method to estimate the vessel heading and the distance traveled by the vessel using only a single 3-D accelerometer. This is required when measurement of the vessel orientation using a gyrocompass for example, is unavailable or
too noisy to directly compensate for the vessel pitch and heading angles. Using only 3-D acceleration measurements, the major challenge in such calculation is the wave-induced vessel’s time-varying pitch angle. Considering this problem, based on the periodic nature of the vessel pitch angle, we described a machine learning classification approach that forms classes of acceleration measurements for which the pitch angle is similar. For each class, our method estimates the vessel pitch angle, and projects the available acceleration measurements into the horizontal plane. The projected measurements are then used to estimate the distance traveled by the vessel via simple integration. Since our method relies on models for the distribution of acceleration measurements, alongside simulations, we presented results from a sea trial conducted in the Singapore strait. Both our simulation and sea trial results confirmed that at a cost of increased complexity, our method accurately estimates orientation and distance using only a single accelerometer device.
Part II

Spatial Reuse MAC techniques for UWANs
Chapter 6

Time and Spatial Reuse Handshake Protocol for UWANs

Using the UWL and location tracking capabilities developed in Part I, we now move on to describe location-dependent MAC protocols for UWANs. In this chapter, we consider the case of UWANs which support peer-to-peer communication between any pair of one-hop neighbor nodes, i.e., nodes which are in the communication range of each other. Our main contribution is a distributed CA handshake-based MAC protocol that makes use of joint time and spatial reuse. This protocol will be referred to as the joint time and spatial reuse (TSR) handshake protocol. Our approach utilizes nodes’ location information and exploits the long propagation delay in UWAC channels as well as the (possible) sparsity of UWAN topologies to improve channel utilization in handshake-based MAC protocols. Our protocol is specific for UWANs for the following reasons: First, it utilizes the long propagation delay in the UAC. Second, while not limited to, our protocol works best in sparse networks, which in the UAC is often the case due to the high attenuation and the existence of obstacles. Third, our protocol considerably reduces packet collisions, which is of interest in UWAC where nodes have a limited energy supply. Last, our protocol relies on relatively moderate to long data messages, which due to the relatively low bit rates is the case in UWAC.

The remainder of this chapter is organized as follows. System model and objectives are introduced in Section 6.1. In Section 6.2 we formalize the problem of maximizing channel utilization in CA scheduling, and in Section 6.3 we describe the details of a distributed sub-optimal solution for this problem. Simulation results are presented in Section 6.4, and conclusions are offered in Section 6.5.

6.1 System Model and Objectives

We are interested in a CA handshake-based resource allocation protocol that achieves both high MAC throughput and low scheduling delay, while limiting primary conflicts, defined as simultaneously arriving (i.e., overlapping) packets from different senders at a common receiver. Considering the possible effect of remote interference on the signal-to-interference-plus-noise ratio (SINR), and thus on the effectiveness of RTS/CTS transmission [131], we assume a packet is lost if the SINR at the receiver is such that the packet error probability is above a required level. Since it is difficult to
online predict the SINR and since nodes may not be aware of transmissions outside
the communication range, in the TSR protocol, we avoid primary conflicts only with
nodes whose transmissions surely interfere with the reception and whose transmission
scheduling is known, and possible collisions with other sources of interferences need
to be tolerated. To formalize this, let the network be described by the undirected
graph $G(N, K)$ with the set $N$ of vertices, representing nodes, and the set $K$ of edges,
representing communication links. Also let $K_j$ be the list of nodes sharing a commu-
nication link with node $j$. For a node $j$, we avoid primary conflicts with the conflict
set $I_j$ of nodes located within the interference range of $j$, such that $I_j \subseteq K_j$. That
is, if the interference range is larger than the communication range, we have $I_j = K_j$. Set $K_j$ is obtained by including all nodes whose packets are successfully decoded by
$j$. Furthermore, using an attenuation model, the interference range, common for all
nodes, is a-priori calculated as the maximal range for which the interference-to-noise
ratio (INR) is above a threshold $T_{INR}$. Thus, upon obtaining $K_j$, node $j$ can calculate
its conflict set, $I_j$.

Let $T^{pd}_{j,j'}$ denote the propagation delay in link $(j, j')$\textsuperscript{14}. We assume that a node $j$
obreak\hspace{1em} obtains $T^{pd}_{j,p}$, $p \in K_j$, with a certain accuracy bounded by $t^{tol}$. Considering possible
outdated propagation delay information in the network, we focus on scenarios where
nodes are static or their motion is limited by $t^{tol}$ (for example, nodes which track their
time-varying locations and share them across the network). We consider a UWAN
in which a node $j \in N$ has a message $i$ of duration\textsuperscript{15} $T^{msg}_{i,j,j'} \gg T^{pd}_{j,j'}$ to transmit to
node $j' \in K_j$, and $j'$ may or may not respond with its own message to $j$. We consider
applications with heavy network load for short period of times in which several CS
can exist simultaneously. In our setting, nodes need not be aware of the number of
nodes in the network, packets arrive randomly, and the identity of the destination
node, $j'$, may change over time. Thus, although we consider static or slowly moving
nodes with respect to $t^{tol}$, the network topology may change dynamically.

The above setup restricts our protocol to a class of applications where nodes
are static or slowly moving with respect to $t^{tol}$, duration of transmitted message is
moderate to long, and communication can be either two-way (symmetric or non-
symmetric) or one-way. Such applications may include submerged buoys, divers, or
underwater structures, involving command and control or data retrieval. We identify
two existing applications supporting this kind of setup. The first is a study of the
migration and survival of marine animals, implemented by Kintama-Research based
in Vancouver Island, Canada [132]. Kintama has several acoustic arrays deployed
in the Pacific Ocean near and north of Vancouver Island. Each array is comprised

\textsuperscript{14}We indicate the communication partner of node $x$ as $\bar{x}$. Note that $\bar{x}$ can be different for a
different CS.

\textsuperscript{15}Note that the restriction on $T^{msg}_{i,j,j'}$ is because when messages are short, due to the long channel
reservation process, there is little benefit in using spatial reuse techniques for scheduling simultaneous
transmissions.
of tens of underwater buoys spaced by hundreds of meters. Kintama holds periodic maintenance missions, during which a separate connection to each acoustic sensor is setup in order to collect data and update its software. However, by forming a network between the array elements and routing data between sensors, efficiency may significantly increase, saving valuable time and effort. In this kind of setting, two-way or one-way communication of long packets is required between the network nodes.

A second practical example is a system called ”Deep-Link”, which includes up to 15 nodes and is used for command and control, surveillance, and diver-safety purposes in three navies. Nodes in the Deep-Link system are moving while continuously tracking their location using gyrocompass and DVL. This location is periodically shared across the network (see system specification in [133]). While often, due to nodes motion, location information of nodes in the Deep-Link system propagates too slowly in the network, there are scenarios where nodes remain static or slowly move while exchanging large image and voice files are exchanged (either one-way or two-way) between nodes, in possibly sparse network setting. The Deep-Link application sets limits on the end-to-end transmission delay, and energy supply is limited. Thus, throughput and delay are of interest. Moreover, in the Deep-Link system, the network may dynamically change (i.e., different transmitter-receiver pairs, change of destination nodes, nodes leaving or entering the network, and nodes rebooting). Therefore, in cases where moderate-to-large files are exchanged and propagation delay information is reliable, handshake-based approach for scheduling transmissions in two-way (symmetric or asymmetric) or one-way communication that utilizes network resources is required for the Deep-Link system.

In the following subsections, we formalize our objectives.

\section{MAC Throughput}

Denote $M_{\text{succ}}^{j,\hat{j}}(W)$ the set of indices of (original and relayed) unicast messages node $j \in \mathcal{N}$ successfully transmits to node $\hat{j} \in \mathcal{K}_j$ during the time interval of $W$ seconds, and let $L_i$ indicate the length of message $i$. The per-link MAC throughput is defined as

$$\rho_{\text{through,link}}^{j,\hat{j}}(W) = \frac{\sum_{i \in M_{\text{succ}}^{j,\hat{j}}(W)} L_i}{W}, \quad (j, \hat{j}) \in \mathcal{K}.$$  \hspace{1cm} (6.1)

Consequently, the average per-link MAC throughput is given by

$$\rho_{\text{through}}(W) = \frac{1}{|\mathcal{K}|} \sum_{(j, \hat{j}) \in \mathcal{K}} \rho_{\text{through,link}}^{j,\hat{j}}(W),$$  \hspace{1cm} (6.2)

where $|X|$ is the cardinality of set $X$. 
6.1.2 Fairness

Assuming equal message generation rate across the network, fairness can be measured by comparing the MAC throughput of nodes in (6.1). Applying the widely used Jain’s fairness index [134], we have the fairness measure

$$\rho_{\text{fair, through}}(W) = \frac{1}{|K|} \left( \sum_{(j, \hat{j}) \in K} \left( \frac{\rho_{\text{through, link}}(W)}{\rho_{\text{through, link}}(W)} \right) \right)^2.$$  

(6.3)

6.1.3 Scheduling Delay

We only consider the portions of link delay that are affected by the MAC protocol, which we denote as scheduling delay. The MAC protocol affects link delay by allowing nodes to transmit (original or relayed packets) at specific times. Denote $s_{\text{RTS,init}}^{j,\hat{j},i}$ the time node $j \in \mathcal{N}$ tries to reserve the channel to transmit message $i$ to node $\hat{j} \in K_j$, and $s_{\text{finish}}^{\hat{j},j,i}$ the time node $\hat{j}$ successfully received message $i$ from node $j$. Then, scheduling delay is defined as

$$\rho_{\text{delay}}(W) = \frac{1}{|K|} \sum_{(j, \hat{j}) \in K} \frac{1}{|M_{\text{succ}}(W)|} \sum_{i \in M_{\text{succ}}(W)} \left( s_{\text{finish}}^{\hat{j},j,i} - s_{\text{RTS,init}}^{j,\hat{j},i} - T_{\text{msg}}^{i,j,\hat{j}} \right).$$  

(6.4)

For clarity, in the following we drop the message subindex and consider the scheduling and transmission of a single message.

6.2 Maximizing Channel Utilization in Handshake Protocols

In the TSR protocol we make use of both time and spatial reuse techniques to maximize channel utilization. Following [69], if two-way communication is required, we achieve time-reuse by allowing both nodes $j \in \mathcal{N}$ and $\hat{j} \in K_j$ to simultaneously exchange messages. As illustrated in Figure 6.1, this is performed by dividing messages into a series of packets and utilizing the long propagation delay to allow simultaneously packet transmission, thus improving channel utilization as only a single initialization process is required per CS. Furthermore, by scheduling simultaneous transmissions from different CSs, we achieve spatial reuse even for one-way or asymmetric communication. In the following, we combine this time reuse with spatial-reuse techniques to enable simultaneous transmissions in neighbor CSs, defined as CSs whose nodes can construct a connected conflict sub-graph. In traditional handshake-based protocols, nodes detecting an RTS and its corresponding CTS packet should stay
silent for the entire CS. Furthermore, there are cases where a node should remain silent when detecting only the RTS or CTS packets, e.g., when the destination node is expected to transmit an acknowledgment packet. Thus, channel utilization greatly decreases with node density. Utilizing both time and spatial reuse allows us to alleviate this shortcoming.

A CS \( C_{j,\tilde{j}} \), \( (j, \tilde{j}) \in \mathcal{K} \), is defined by three characteristic parameters referred to as the CS parameters: 1) \( s_{j,\tilde{j}} \), 2) \( d_{j,\tilde{j}} \) and 3) \( t_{j,\tilde{j}} \), where \( s_{j,\tilde{j}} \) is the time node \( j \) transmitted its first data packet to node \( \tilde{j} \), \( d_{j,\tilde{j}} \) is the duration of a single packet transmitted from \( j \) to \( \tilde{j} \), and \( t_{j,\tilde{j}} \) is the time difference between the starting transmission times of consecutive packets, referred to as the cycle time of the message. Consequently,

\[
\hat{s}_{j,\tilde{j}} = s_{j,\tilde{j}} + (N_{j,\tilde{j}}^{\text{cycle}} - 1)t_{j,\tilde{j}} + d_{j,\tilde{j}}, \ (j, \tilde{j}) \in \mathcal{K}, \tag{6.5}
\]

where \( N_{j,\tilde{j}}^{\text{cycle}} = \lceil \frac{T_{j,\tilde{j}}^{\text{msg}}}{d_{j,\tilde{j}}} \rceil \) is the number of cycles in the CS \( C_{j,\tilde{j}} \). Our protocol is based on scheduling CSs. To form the CS \( C_{j,\tilde{j}} \), nodes \( j \) and \( \tilde{j} \) determine the CS parameters, while avoiding primary conflicts, and transmit data packets of duration \( d_{j,\tilde{j}} \) once in every \( t_{j,\tilde{j}} \) seconds starting from time \( s_{j,\tilde{j}} \).

In the following we identify three types of primary conflicts: 1) when a new CS interferes with active CSs, 2) when transmissions from active CSs interfere transmission of a new CS, and 3) interference within the CS.


6.2.1 Types of Primary Conflicts

Conflict type 1 - Interference to active CSs

Consider the two CSs $C_p, \tilde{p}$ and $C_{j, \tilde{j}}$, such that $p \in I_j$. In conflict type 1, packets from node $j$ arrive at node $p$ while the latter is receiving packets from node $\tilde{p}$. This scenario is illustrated in Figure 6.2a. To avoid such interference, $j$ should schedule its transmissions such that its packets arrive at $p$ while $p$ is not receiving from $\tilde{p}$.

Conflict type 2 - Interference from active CSs

Conflict type 2 involves packet transmission from node $j$ to node $\tilde{j}$ while $\tilde{j}$ experiences interferences from neighbor CSs, as illustrated in Figure 6.2b.

Conflict type 3 - Interference within the CS

In this type of conflict, a packet from node $j$ arrives at node $\tilde{j}$ while the latter is transmitting to $j$. Due to the half-duplex property of the acoustic transducers, $\tilde{j}$ would not be able to detect the packet of $j$ and it would be lost. This is illustrated in Figure 6.2c. Similarly, node $j$ might transmit its own packet while receiving a packet from $\tilde{j}$, as illustrated in Figure 6.2d. Avoiding such a conflict, $j$ should consider transmissions from $\tilde{j}$ while scheduling its own transmissions.

Conflict types 1 and 2 are caused by interference to and from neighbor CSs, while conflict type 3 is caused by synchronization problems with the destination node. We note that when scheduling transmissions, all types of primary conflicts should be taken into account. Next, we formalize constraints to avoid the above primary conflicts.

6.2.2 Formalizing Constraints

We start with formalizing general constraints for $t_{j, \tilde{j}}$ and $d_{j, \tilde{j}}$, which apply to all three scenarios discussed above. Avoiding primary conflicts of type 3, a new data packet cannot be transmitted before the destination node received the previous data packet. Moreover, avoiding primary conflicts of type 1 and 2, $t_{j, \tilde{j}}$ should be an integer multiple of cycle times of neighbor CSs, otherwise primary conflicts avoided in certain CS cycles might still exist in later cycles, resulting in packet collisions. Let $\mathcal{R}$ be the set of nodes already participating in CSs or seeking to reserve the channel. Then, for an integer $n$, the above constraints on $t_{j, \tilde{j}}$ can be formalized by

$$d_{j, \tilde{j}} + T_{j, \tilde{j}}^{pd} + t_{j, \tilde{j}} \leq n \times t_{p, \tilde{p}} \forall p \in \mathcal{R} \cap (I_j \cup I_{\tilde{j}}), \ p \neq j, \tilde{j}. \quad (6.6)$$

Assuming the nodes in $\mathcal{R}$ form a connected conflict graph (otherwise the network is divided into sub-networks, whose scheduling is performed separately), we observe
Figure 6.2: Illustration of different types of primary conflicts.
that constraint (6.6) is satisfied by setting
\[ t_{j, \tilde{j}} = \max_p \left( d_{p, \tilde{p}} + T_{p, \tilde{p}}^{pd} \right) + t_{tol} \forall p \in \mathcal{R}. \] (6.7)

That is, we adopt a common cycle time within the network. While the solution in (6.7) may not be the optimal one to minimize scheduling delay, it is the simplest and a more fair solution.

Next, as illustrated in Figure 6.1, in order for two nodes \( j \) and \( \tilde{j} \) to simultaneously transmit in a CS, the duration of their packets should be smaller than \( T_{j, \tilde{j}}^{pd} \). Thus, we apply the following constraint to the packet duration:
\[ d_{j, \tilde{j}} \leq T_{j, \tilde{j}}^{pd} - t_{tol}. \] (6.8)

We now continue to formalize specific constraints to avoid the three types of primary conflicts described in Section 6.2.1 when scheduling transmissions in the CS \( C_{j, \tilde{j}} \). For convenience, considering nodes \( p \) and \( q \) such that \( q \in \mathcal{I}_p \), we define the variable \( T_{b}^{p, q, n, p, \tilde{p}} \) as the time that the \( n_{p, \tilde{p}} \)th packet from node \( p \) arrives at node \( q \), i.e.,
\[ T_{b}^{p, q, n, p, \tilde{p}} = s_{p, \tilde{p}} + T_{p, q}^{pd} + n_{p, \tilde{p}} t_{p, \tilde{p}}, \] (6.9)
and the related time
\[ T_{e}^{p, q, j, n, p, \tilde{p}} = T_{b}^{p, q, n, p, \tilde{p}} + d_{j, \tilde{j}}. \] (6.10)

**Formalizing Conflict Type 1**

Avoiding conflict type 1 (see Figure 6.2a), the packets of node \( j \) should arrive at node \( p \) before or after those transmitted from node \( \tilde{p} \). Let \( n'_{p, j} \) be the index of the first packet transmitted from \( \tilde{p} \) which possibly experiences interference from the transmissions of \( j \). Assuming \( j \) is ready to transmit in time \( s_{j, \tilde{j}}^{init} \), for
\[ x_{n_{p, \tilde{p}}, p, j} = T_{b}^{p, p, n_{p, \tilde{p}} + 1} - \left( s_{j, \tilde{j}}^{init} + d_{j, \tilde{j}} + T_{j, \tilde{j}}^{pd} \right), \] (6.11)
\( n'_{p, j} \) can be found as
\[ n'_{p, j} = \arg\min_{n_{p, \tilde{p}}} \left( x_{n_{p, \tilde{p}}, p, j} \right), \] (6.12a)
\[ \text{s.t.} \quad x_{n_{p, \tilde{p}}, p, j} \geq 0. \] (6.12b)

Obtaining \( n'_{p, j} \) \( \forall p \in \mathcal{R} \cap \mathcal{K}_j, p \neq \tilde{j} \), from (6.12), constraint 1 is formalized as
\[ s_{j, \tilde{j}} \geq T_{e}^{p, p, n', p, j} - T_{j, \tilde{j}}^{pd} + 2t_{tol}, \] (6.13a)
\[ s_{j, \tilde{j}} + d_{j, \tilde{j}} \leq T_{b}^{p, p, n', p, j + 1} - T_{j, \tilde{j}}^{pd} - 2t_{tol}. \] (6.13b)
Chapter 6. Time and Spatial Reuse Handshake Protocol for UWANs

Formalizing Conflict Type 2

To formalize constraint 2, illustrated in Figure 6.2b, we schedule the transmissions of node \( j \) such that its packets arrive at node \( \tilde{j} \) when the latter is not experiencing interference from nodes in \( I_{\tilde{j}} \). To this end, we find nodes \( \hat{p} \in \mathcal{R} \) and \( \hat{q} \in \mathcal{R} \), whose transmissions are the first and last to possibly interfere the reception of \( j \)’s first packet at \( \tilde{j} \), respectively (note that it is possible that \( \hat{p} = \hat{q} \)), and corresponding packet indices \( n_{p,j}'' \leq N_{\hat{p}}^{\text{cycle}} \) and \( n_{q,j}'' \leq N_{\hat{q}}^{\text{cycle}} \), according to

\[
(p, \hat{q}, n_{\hat{p},j} '', n_{\hat{q},j} '') = \arg\min_{(p,q,n_{p,j}', n_{q,j}')}(T_{q,j,n_{q,j}'}) - T_{p,j,p,n_{p,j}'}, \tag{6.14a}
\]

subject to

\[
T_{p,j,p,n_{p,j}'+1}^{e} - 2t_{\text{tol}} > s_{j,j} + T_{j,j}^{\text{pd}}, \tag{6.14b}
\]

\[
T_{p,j,p,n_{p,j}'+1}^{e} - 2t_{\text{tol}} > s_{j,j} + T_{j,j}^{\text{pd}}, \tag{6.14c}
\]

\[
T_{q,j,n_{q,j}'-1}^{b} + 2t_{\text{tol}} < s_{j,j}^{\text{init}} + T_{j,j}^{\text{pd}}, \tag{6.14d}
\]

\[
T_{p,j,p,n_{p,j}'+1}^{e} - 2t_{\text{tol}} > s_{j,j} + T_{j,j}^{\text{pd}}, \tag{6.14e}
\]

\[
n_{p,j}'' \leq N_{\hat{p}}^{\text{cycle}}, n_{q,j}'' \leq N_{\hat{q}}^{\text{cycle}}. \tag{6.14f}
\]

Using the solutions from (6.14), we set the following constraints

\[
s_{j,j} \geq T_{p,j,p,n_{p,j}'}^{e} - T_{j,j}^{\text{pd}} + 2t_{\text{tol}}, \tag{6.15a}
\]

\[
s_{j,j} + d_{j,j} \leq T_{q,j,n_{q,j}'}^{b} - T_{j,j}^{\text{pd}} - 2t_{\text{tol}}. \tag{6.15b}
\]

Formalizing Conflict Type 3

Next, referring to Figure 6.2c, we formalize constraint 3. Assuming \( s_{j,j}^{\text{init}} = s_{j,j}^{\text{init}} \), i.e., both nodes \( j \) and \( \tilde{j} \) are ready to transmit at the same time and that \( T_{j,j}^{\text{pd}} = T_{j,j}^{\text{bd}} \), we set the following conditions

\[
s_{j,j} + T_{j,j}^{\text{pd}} \geq T_{j,j,j,0}^{e} - T_{j,j}^{\text{pd}} + t_{\text{tol}}, \tag{6.16a}
\]

\[
s_{j,j} + d_{j,j} \leq T_{j,j,j,1}^{b} - 2T_{j,j}^{\text{pd}} - t_{\text{tol}}. \tag{6.16b}
\]

Similarly, we set

\[
s_{j,j} + d_{j,j} \leq T_{j,j,j,0}^{b} - t_{\text{tol}}. \tag{6.17}
\]

Merging Constraints for the Different Types of Primary Conflicts

Fortunately, the four conditions (6.13), (6.15), (6.16) and (6.17) can be merged into two constraints. In order to avoid conflict of type 1 we should consider all neighbor CSs. To formalize this, referring to constraint (6.13) we construct upper and lower bound vector \( \mathbf{u}_1 \) and \( \mathbf{l}_1 \) with elements \( T_{p,p,p,n_{p,j}'}^{e} - T_{j,p}^{\text{pd}} + 2t_{\text{tol}} \) and \( T_{p,p,n_{p,j}'+1}^{b} - T_{j,p}^{\text{pd}} - 2t_{\text{tol}} \),
respectively, \( \forall p \in \mathcal{R} \cap \mathcal{J}, p \neq \tilde{j} \). Furthermore, considering the other types of primary conflicts and referring to (6.15), (6.16) and (6.17) we set the lower bounds
\[
\begin{align*}
l_2 &= T_{p;j} - L_{p,j} + 2t_{tol} \\
l_3 &= T_{j,j,0} - 2T_{j,j} + t_{tol},
\end{align*}
\]
and the upper bounds
\[
\begin{align*}
u_2 &= T_{b;j} - L_{b,j} - 2t_{tol} \\
v_3 &= T_{j,j,1} - 2T_{j,j} - t_{tol} \\
v_4 &= T_{j,j,0} - t_{tol}.
\end{align*}
\]
Then, the four constraints introduced above can be merged onto
\[
\begin{align*}
s_{j,j} &\geq \max (l_1, l_2, l_3) \triangleq l \quad (6.20a) \\
s_{j,j} + d_{j,j} &\leq \min (u_1, u_2, u_3, u_4) \triangleq u, \quad (6.20b)
\end{align*}
\]
where (6.20a) and (6.20b) are performed element-wise.

We next formalize an optimization problem taking all of the above constraints into consideration.

### 6.2.3 Channel Utilization Maximization Problem

We are interested in maximizing channel utilization by maximizing the packet duration, \( d_{j,j} \), and minimizing the transmission starting time, \( s_{j,j} \), for each CS while avoiding packet collisions. Considering constraints (6.8) and (6.20), for a utility function \( f(s_{j,j}, d_{j,j}) \), a minimum required packet duration, \( d_{\text{min}} \) (set by the packet header duration), and a fixed time interval \( T_{\text{offset}} \) used to bound the waiting time \( s_{j,j} - s_{j,j}^{\text{init}} \), we formalize the channel utilization maximization problem (CUMP)

\[
\begin{align*}
\text{maximize} & \quad \sum_{j \in \mathcal{R}} f(s_{j,j}, d_{j,j}) \\
\text{s.t.} & \quad s_{j,j} \geq l \quad (6.21b) \\
& \quad s_{j,j} + d_{j,j} \leq u \quad (6.21c) \\
& \quad s_{j,j} \leq s_{j,j}^{\text{init}} + T_{\text{offset}} \quad (6.21d) \\
& \quad d_{j,j} \leq T_{j,j} - t_{tol} \quad (6.21e) \\
& \quad d_{j,j} \geq d_{\text{min}}, \quad (6.21f)
\end{align*}
\]
to obtain \( s_{j,j} \) and \( d_{j,j} \) \( \forall j \in \mathcal{R} \), followed by a calculation of \( t_{j,j} \) using (6.7).

We observe that solving (6.21) requires a centralized approach to obtain the parameters of the CS \( C_{j,j} \) \( \forall j \in \mathcal{R} \), which significantly increases the communications overhead. Therefore, we next describe our TSR protocol which offers a sub-optimal distributed solution for the CUMP (6.21).
6.3 The TSR Protocol - A Sub-Optimal Approach

We first relax (6.21) by considering synchronized communication in the CS \( C_{\tilde{j},j} \) such that \( s_{\tilde{j},j} = s_{j,j}, d_{\tilde{j},j} = d_{j,j} \) and \( t_{\tilde{j},j} = t_{j,j} \) (however the numbers of packets transmitted by \( j \) and \( \tilde{j} \) need not be the same). As a result, as long as constraints (6.8) and (6.6) are satisfied, conflict type 3 does not need to be considered. Moreover, we replace the set \( R \) with a set \( R_j \subseteq (I_j \cup I_{\tilde{j}}) \), which is the set of nodes who participate in active CSs at time \( s_{\text{RTS,init}} \), which are neighbors to either \( j \) or \( \tilde{j} \), hereby scheduling one CS at a time (instead of scheduling all CSs together, as we did in (6.21)). To formalize the TSR protocol we replace \( u \) and \( l \) in (6.21) with different bounds \( \tilde{u} \) and \( \tilde{l} \), which unlike the former are not a function of the parameters of CS \( C_{\tilde{j},j} \) (a detailed description will be given in Section 6.3.2 below). Then, for a single CS \( C_{\tilde{j},j} \), the CUMP (6.21) becomes

\[
\begin{align*}
\text{minimize} & \quad f(s_{\tilde{j},j}, d_{\tilde{j},j}) \\
\text{s.t.} & \quad s_{\tilde{j},j} \geq \tilde{l} \quad (6.22a) \\
& \quad s_{\tilde{j},j} + d_{\tilde{j},j} \leq \tilde{u} \quad (6.22b) \\
& \quad s_{\tilde{j},j} \leq s_{\text{init}}^{\tilde{j},j} + T_{\text{offset}} \quad (6.22c) \\
& \quad d_{\tilde{j},j} \leq T_{p_{\tilde{j},j}}^{pd} - t_{\text{tol}} \quad (6.22d) \\
& \quad d_{\tilde{j},j} \geq d_{\text{min}} \quad (6.22e)
\end{align*}
\]

Since in (6.22a), \( s_{\tilde{j},j} \) is minimized and \( d_{\tilde{j},j} \) is maximized, regardless of the utility function, the solution for (6.22) is given by

\[
\begin{align*}
\quad s_{\tilde{j},j} &= \tilde{l} \quad (6.23a) \\
\quad d_{\tilde{j},j} &= \min \left( T_{p_{\tilde{j},j}}^{pd} - t_{\text{tol}}, \tilde{u} - \tilde{l} \right) \quad (6.23b)
\end{align*}
\]

and we satisfy constraints (6.22d) and (6.22f) by verifying that \( s_{\tilde{j},j} \leq s_{\text{init}}^{\tilde{j},j} + T_{\text{offset}} \) and that \( d_{\tilde{j},j} \geq d_{\text{min}} \). Otherwise, since constraints (6.22d) and (6.22f) are upper and lower limits on \( s_{\tilde{j},j} \) and \( d_{\tilde{j},j} \), respectively, there is no feasible schedule for the CS \( C_{\tilde{j},j} \) and transmissions are deferred. Since \( n_{p_{\tilde{j},j}} \) in (6.12), required for both \( \tilde{u} \) and \( \tilde{l} \) (as we show further below), is a function of \( d_{\tilde{j},j} \), it might be hard to calculate \( d_{\tilde{j},j} \) from (6.23b). Considering this problem, we suggest a heuristic approach that finds the largest feasible \( d_{\tilde{j},j} \) (whose effect on both throughput and scheduling delay is greater than \( s_{\tilde{j},j} \)) using a bisection procedure over the range \( T_{p_{\tilde{j},j}}^{pd} \) and \( d_{\text{min}} \).

Next, since the solution in (6.7) to obtain \( t_{j,j} \) requires a centralized approach in which \( d_{p,\tilde{p}} \) and \( T_{p_{\tilde{j},j}}^{pd} \) \( \forall p \in R \) are known, we offer a modified solution that satisfies
(6.6) by setting \( t_{\hat{j},\hat{j}} \) to be the least common multiple (LCM) of neighbor CS’s cycle times. To formalize this, let us denote the cycle time vector \( \mathbf{t}_j = [t_{p_1,\hat{p}_1}, \ldots, t_{p_t,\hat{p}_t}] \) with rational elements \( t_{p_i,\hat{p}_i} \), for nodes \( p_i \in \mathcal{R}_j \), \( p_i \neq \hat{j}, \hat{j} \), with \( I \) being the number of such elements (note that \( \mathbf{t}_j = \mathbf{t}_j \)). Also define \( \mathbf{\tau}_j = [\mathbf{t}_j, Q(d_{\hat{j},\hat{j}} + T^{pd}_{\hat{j},\hat{j}} + t_{tol})] \) (where \( Q(x) \) is the nearest rational number of \( x \) such that \( Q(x) \geq x \)), and let \( L(\mathbf{v}) \) be the LCM of the elements of vector \( \mathbf{v} \), such that \( \frac{L(\mathbf{\tau}_j)}{L(t_j)} \) is an integer. Then, for an optimized integer \( \hat{n}_j \) we set

\[
\hat{n}_j = \min(n) \quad \text{s.t.} \quad n \times L(\mathbf{t}_j) \geq d_{\hat{j},\hat{j}} + T^{pd}_{\hat{j},\hat{j}} + t_{tol} 
\]

\[
\hat{n}_j \in \{1, \ldots, \frac{L(\mathbf{\tau}_j)}{L(t_j)} \}. \tag{6.25c}
\]

Assuming that \( \forall p \in \mathcal{R}_j \), parameters \( s_{p,\hat{p}}, \tau_{p,\hat{p}} \) and \( d_{p,\hat{p}} \) are known to \( j \) at time \( s_{j,\hat{j}}^{\text{RTS,init}} \) (we qualify this assumption further below by introducing a mechanism in which nodes learn the CS parameters of their one-hop neighbors and synchronize this data with their destination node), node \( j \) can calculate \( \hat{n}, \hat{l} \) and \( \hat{n}_j \) to solve (6.23) and (6.24). Thus, unlike (6.21), (6.22) can be solved distributively for a single CS \( C_{\hat{j},\hat{j}} \).

The TSR protocol includes a sequential procedure to schedule multiple CSs. Consider the sequence \( \mathcal{J} = \{j_1, j_2, \ldots, j_M\} \), of \( M \) nodes forming a fully-connected conflict graph (i.e., \( j_m \in \mathcal{R}_{j_n}, \forall j_m, j_n \in \mathcal{J} \)) and wishing to reserve the channel, enumerated in ascending order according to \( s_{j_m,j_n}^{\text{RTS,init}} \), such that CSs \( C_{j_m,\hat{j}_m} \) and \( C_{j_n,\hat{j}_n}, \forall j_m, j_n \in \mathcal{J}, n \neq m \) are neighbor CSs (note that this setting does not mean that the network is fully connected). First, node \( j_1 \) determines the CS \( C_{j_1,\hat{j}_1} \) parameters using (6.23) and (6.24) for \( \mathcal{R}_{j_1} = \emptyset \). Next, detecting the CS \( C_{j_1,\hat{j}_1} \) at time \( s_{j_2,j_2}^{\text{RTS,init}} \), node \( j_2 \) determines the CS \( C_{j_2,\hat{j}_2} \) parameters for the set \( \mathcal{R}_{j_2} = \{j_1, \hat{j}_1\} \). Finally, detecting transmissions of the previous CSs \( C_{j_m,\hat{j}_m}, m = 1, \ldots, M - 1 \) at time \( s_{j_M,j_M}^{\text{RTS,init}} \), node \( j_M \) determines the parameters of the CS \( C_{j_M,\hat{j}_M} \) for the set \( \mathcal{R}_{j_M} = \{j_1, \hat{j}_1, \ldots, j_{M-1}, \hat{j}_{M-1}\} \). We note that in order to make the TSR a distributed protocol, a node scheduling the parameters of its CS will use all available network resources. Thus, the TSR protocol is a greedy protocol and fairness in resource allocation may be affected. We also observe the similarity of this approach to cognitive radio in which secondary users utilize available frequency bands without interfering primary users [135]. Here, primary users may be CSs in progress while secondary users are newly established CSs, and the sensing step is the mechanism used to identify the parameters of active CSs.

Consider the network topology in Figure 6.3a for the purpose of illustration of our protocol, where \( x \in \mathcal{I}_p, p \in \mathcal{I}_j \), and \( \hat{x} \in \mathcal{I}_j \). The propagation delay in all links
is assumed to be equal to $T_{pd}$. We start with node $p$ transmitting $N_{cycle}^p$ packets each of duration $T_{pd}$ to node $\tilde{p}$. In this case, for a time window of $T = N_{cycle}^p T_{pd}$, MAC throughput is simply 1. Next, we allow $\tilde{p}$ to transmit the same amount of data to $p$. This is possible if packet transmissions are spaced by at least $T_{pd}$ seconds, as suggested in [69] and illustrated in Figure 6.3b. Since both nodes can transmit $N_{cycle}^p/2$ packets during the time window $T$, neglecting channel reservation time, MAC throughput remains the same. We now consider the case where node $j$ tries to establish two-way CS $C_j$, transmitting the same amount of information as in $C_{p,\tilde{p}}$. By implementing our protocol we allow $j$ and $\tilde{j}$ to transmit together with $p$ and $\tilde{p}$. Since transmissions in the CS $C_{j,\tilde{j}}$ are only limited by the CS $C_{p,\tilde{p}}$ as illustrated in Figure 6.3b, MAC throughput for a time window of $T = N_{cycle}^p T_{pd}$ is expected to increase to 2. Finally, we consider the case of CS $C_{x,\tilde{x}}$ joining CSs $C_{p,\tilde{p}}$ and $C_{j,\tilde{j}}$. Avoiding interference, nodes $x$ and $\tilde{x}$ schedule their transmissions according to (6.23) and (6.24) such that their transmissions do not interfere $C_{p,\tilde{p}}$ and $C_{j,\tilde{j}}$. This process is illustrated in Figure 6.3b, where $d_{x,\tilde{x}}$ and $s_{x,\tilde{x}}$ are set such that packets from $x$ arrive at $p$ and $j$ while they are transmitting and at $\tilde{x}$ while it is not experiencing interference from $\tilde{j}$, and such that packets from $\tilde{x}$ arrive at $\tilde{j}$ while it is transmitting and at $x$ while it is not experiencing interference from $j$ and $p$. Numerical results (see Section 6.4 below) show that for the latter case MAC throughput for the time window $T$ increases to 2.5.
Data packet transmission in the CS $C_{j,\bar{j}}$ is preceded by channel reservation process based on exchanging RTS, CTS and notification (NT) control packets of fixed sizes, $d_{\text{RTS}}, d_{\text{CTS}}$ and $d_{\text{NT}}$, respectively, between $j$ and $\bar{j}$. The RTS packet is transmitted by node $j$ for the process of initializing channel reservation and includes a temporal set of parameters for the CS $C_{j,\bar{j}}$, determined by $j$. The CTS packet, including the final parameters of the CS $C_{j,\bar{j}}$, is transmitted by node $\bar{j}$ to synchronize the parameters of the CS $C_{j,\bar{j}}$ with node $j$, and to let node $p \in K_j$, $p \neq j$ update its $R_j$ database. Finally, the NT packet, which also includes the final CS parameters, is transmitted by node $j$ to notify node $p \in K_j$, $p \neq \bar{j}$ of the CS $C_{j,\bar{j}}$ parameters, immediately followed by $N^\text{cycle}_{j} \frac{T_{\text{msg}}}{d_{j,\bar{j}}} \text{ data packets of duration } d_{j,\bar{j}} \text{ in time cycles of } t_{j,\bar{j}} \text{ seconds,}$ while node $j$ starts transmitting its $N^\text{cycle}_{j}$ at the instant when $j$ transmits its NT packet (note that the duration of the first data packet from $j$ is $d_{j,\bar{j}} - d_{\text{NT}}$). We note that in the TSR protocol the RTS and CTS packets need to be scheduled to avoid primary conflicts with neighbor CSs. However, the NT packet is not treated as a separate packet and is transmitted just before the first data packet of node $j$. This process is illustrated in Figure 6.1.

6.3.1 Priority of Control Packets

A common problem in handshake-based protocols (e.g., [68],[69]) are collisions resulting from two neighbor CSs trying to reserve the channel at the same time. Considering this problem, in TSR we apply a priority mechanism for incoming packets. The priority mechanism gives advantage to a node which is in a more advanced stage of its channel reservation process. Consider nodes $j$ and $p$, which are in the process of channel reservation to establish CSs $C_{j,\bar{j}}$ and $C_{p,\bar{p}}$, respectively. The following rules apply:

1. If node $j$ sends an RTS packet at time $T_j$ and is waiting for a CTS response while detecting an RTS packet sent from node $p$ at time $T_p^{16}$, node $j$ will stop its channel reservation process if $T_j > T_p$. Channel reservation of $j$ would also stop if $j$ receives a CTS packet or a data packet from node $p$ while waiting for a CTS packet. However, if $j$ receives an RTS packet from node $p$ after transmitting its own CTS packet, node $j$ will ignore the received RTS packet.

2. If node $j$ sends a CTS packet at time $T_j$ and detects a CTS packet sent from node $p$ at time $T_p$, node $j$ will avoid transmitting its data packets if $T_j > T_p$. Similarly, if node $j$ sends a CTS packet to node $\bar{j}$ at time $T_j$ and detects a data packet from node $p$ with $s_{p,\bar{p}} = T_p$, node $j$ will delay its data packet transmission if $T_j > T_p$.

$^{16}$Note that since we assume that node $j$ is aware of $T_{j,p}^{\text{rd}}$, it is capable of estimating $T_p$. 

126
We note that if node \( j \) stops its channel reservation process due to detection of packets from node \( p \), it will restart the process once the parameters of the newly established CS \( C_{p,\hat{p}} \) are included in \( R_j \). We also note that if \( j \) cancels its data packet transmission, node \( \hat{j} \) may continue transmitting if its transmissions do not collide with transmissions from the newly established CS \( C_{p,\hat{p}} \). The above priority mechanism ensures interference-free schedule in neighbor CSs if these CSs are connected via at least one communication link, otherwise collisions may occur.

Next, we describe the mechanism by which nodes \( j \) and \( \hat{j} \) schedule their RTS and CTS packets and share information of neighbor CSs with minimum communication overhead to readily solve (6.23) and (6.24).

### 6.3.2 Scheduling Control Packets

We observe that when \( I_j \neq I_{\hat{j}} \), node \( j \) might not be able to resolve conflict type 2 without exchanging information with node \( \hat{j} \) to solve (6.23) and (6.24). Therefore, the objective of the channel reservation phase is twofold: 1) to allow nodes \( j \) and \( \hat{j} \) to reserve link \( (j, \hat{j}) \) and 2) to synchronize information of neighbor CSs between \( j \) and \( \hat{j} \) with minimum communications overhead.

Avoiding interference to neighbor CSs, nodes \( j \) and \( \hat{j} \) should schedule the transmission of their RTS and CTS packets\(^{17}\) while avoiding conflict types 1 and 2. Different from data packets, control packets are transmitted only once and have a fixed pre-determined duration. Thus, only the transmission times, \( s_{j,\hat{j}}^{\text{RTS}} \) and \( s_{j,\hat{j}}^{\text{CTS}} \) for the RTS and CTS control packets, respectively, need to be determined.

#### Scheduling RTS

We start with scheduling the transmission time of the RTS packet from node \( j \) to node \( \hat{j} \). As in the process of scheduling data packets, described in Section 6.2.2, \( \forall p \in R_j \), using (6.12) we find index \( n_{p,j}^{\text{RTS}} \) of a packet transmitted from node \( \hat{p} \) to node \( p \) that possibly collides with the RTS packet from \( j \) by replacing \( s_{j,j}^{\text{init}}, d_{j,j}^{\text{init}} \) in (6.11) with \( s_{\hat{j},j}^{\text{RTS,init}}, d_{\hat{j},j}^{\text{RTS}} \), respectively. Then, avoiding conflicts of type 1 we require

\[
\begin{align*}
 s_{j,\hat{j}}^{\text{RTS}} &\geq T_{\hat{p},p,\hat{p},p,j}^{\text{init}} - T_{j,p}^{\text{init}} + 2t_{\text{tol}}, \\
 s_{j,\hat{j}}^{\text{RTS}} &\leq T_{\hat{p},p,\hat{p},p,j}^{\text{init}} + d_{\hat{j},j}^{\text{RTS}} - T_{j,p}^{\text{init}} - 2t_{\text{tol}}.
\end{align*}
\]

Next, in order to avoid primary conflicts of type 2 (see Section 6.2.2) we obtain nodes \( \hat{p}^{\text{RTS}}, q^{\text{RTS}} \) and corresponding indices \( n_{p^{\text{RTS}},j}^{\text{RTS}}, n_{q^{\text{RTS}},j}^{\text{RTS}} \) of packets whose transmissions

\(^{17}\)Recall that the NT control packet is appended to the first data packet of node \( j \), and therefore need not be separately scheduled.
might interfere the reception of the RTS packet at node $\tilde{j}$, by replacing set $\mathcal{R}$ with set $\mathcal{R}_j$ and $s^{\text{RTS,init}}_{j,j}$ in (6.14). Then, as in (6.15), we require

\begin{align}
    s^{\text{RTS}}_{j,j} & \geq T^e_{p^{\text{RTS}},j,j^{\text{RTS}},p^{\text{RTS}}_j} - T^{pd}_{j,j} + 2t_{\text{tol}} \\
    s^{\text{RTS}}_{j,j} & \leq T^b_{q^{\text{RTS}},j,j^{\text{RTS}},q^{\text{RTS}}_j} - d^{\text{RTS}} - T^{pd}_{j,j} - 2t_{\text{tol}}.
\end{align}

(6.27a)  

(6.27b)

Define vectors $\mathbf{u}^{\text{RTS}}_1$ and $\mathbf{l}^{\text{RTS}}_1$ with elements $u^{\text{RTS}}_1(p)$ and $l^{\text{RTS}}_1(p)$ respectively, such that

\begin{align}
    l^{\text{RTS}}_1(p) &= T^e_{p,p,p,p^{\text{RTS}}_j} - T^{pd}_{j,j} + 2t_{\text{tol}}, \\
    u^{\text{RTS}}_1(p) &= T^b_{p,p,p,p^{\text{RTS}}_j + 1} - T^{pd}_{j,j} - d^{\text{RTS}} - 2t_{\text{tol}},
\end{align}

(6.28a)  

(6.28b)

and let

\begin{align}
    l^{\text{RTS}}_2 &= T^e_{p^{\text{RTS}},j,j^{\text{RTS}},p^{\text{RTS}}_j} - T^{pd}_{j,j} + 2t_{\text{tol}}, \\
    u^{\text{RTS}}_2 &= T^b_{q^{\text{RTS}},j,j^{\text{RTS}},q^{\text{RTS}}_j} - T^{pd}_{j,j} - d^{\text{RTS}} - 2t_{\text{tol}}.
\end{align}

(6.29a)  

(6.29b)

Then, constraints (6.26) and (6.27) can be merged onto

\[ l^{\text{RTS}} \triangleq \max(l^{\text{RTS}}_1, l^{\text{RTS}}_2) \leq s^{\text{RTS}}_{j,j} \leq \min(u^{\text{RTS}}_1, u^{\text{RTS}}_2) \triangleq u^{\text{RTS}}. \]

(6.30)

Hence, if $u^{\text{RTS}} \geq l^{\text{RTS}}$, node $j$ sets $s^{\text{RTS}}_{j,j} = l^{\text{RTS}}$, otherwise it waits a backoff time of $T_{\text{offset}}$ seconds (with possibly different bounds $l^{\text{RTS}}$ and $u^{\text{RTS}}$) before trying to reschedule the transmission of its RTS packet.

### Scheduling CTS

Since node $j$ might not be aware of all neighbor CSs of node $\tilde{j}$, it might wrongly identify nodes $p^{\text{RTS}}$ and $q^{\text{RTS}}$ used in (6.27). However, since conflicts of type 2 do not interfere neighbor CSs (see Figure 6.2b) the effect might be the loss of the RTS packet, which can then be retransmitted. However, since transmission of a CTS packet does not require feedback to start data packet transmission, unlike scheduling RTS packets, CTS scheduling must be collision-free. Avoiding information exchange of neighbor CSs between $j$ and $\tilde{j}$, which renders large communication overhead, in the following we describe a mechanism to obtain collision-free CTS transmission based on sharing bounds of scheduling the CS $C_{j,\tilde{j}}$.

Our approach is based on the observation that due to the min/max operations in (6.20), which set the limitations on $s_{j,j}$ and $d_{j,j}$, only the scheduling bounds need to be shared by nodes $j$ and $\tilde{j}$. After scheduling its RTS packet, node $j$ determines temporary parameters for the CS $C_{j,\tilde{j}}$, $\tilde{s}_{j,j}, \tilde{d}_{j,j}$ and $\tilde{l}_{j,j}$ by solving a version of (6.23).
and (6.24) for \( \bar{u} = \min(u_1, u_2) \) and \( \bar{l} = \max(l_1, l_2) \) for \( s_{ij}^{\text{init}} = s_{ij}^{\text{RTS}} + T_{j}^{\text{rd}} \), and replacing \( \mathcal{R} \) with a temporary set \( \mathcal{R}_{j} \subseteq \mathcal{I}_{j} \) including all neighbor CSs node \( j \) is aware of at time \( s_{ij}^{\text{RTS,init}} \). Next, \( j \) transmits the temporary CS parameters to its destination node \( \bar{j} \) along with the upper bound \( \bar{u} \), and waits for the CTS response from \( \bar{j} \) within the time window \([s_{ij}^{\text{RTS}}, \bar{u} + T_{j}^{\text{rd}} + T_{\text{offset}}]\). In turn, \( \bar{j} \) extracts the temporary CS parameters and \( \bar{u} \) from the received RTS packet and is thereby aware of the scheduling limitations set by the neighbor CSs of node \( j \). Hence, \( \bar{j} \) refers to these limitations as a time frame for scheduling its CTS packet transmission and to later set the parameters of the CS \( C_{j,\bar{j}} \). More specifically, \( \bar{j} \) determines \( s_{ij}^{\text{CTS}} \) as in (6.30) replacing \( d_{\text{RTS}} \) with \( d_{\text{CTS}} \) and \( s_{ij}^{\text{RTS,init}} \) with \( s_{ij}^{\text{CTS,init}} = \bar{s}_{ij} + k_1 \times \bar{t}_{j,\bar{j}} \), where \( k_1 \in \{0, \ldots, \lfloor \bar{u} + T_{\text{offset}} - \bar{s}_{j,\bar{j}} / \bar{t}_{j,\bar{j}} \rfloor \} \) is the minimum integer for which

\[
s_{ij}^{\text{CTS}} \leq \bar{u} + k_1 \times \bar{t}_{j,\bar{j}} - d_{\text{CTS}}, \tag{6.31}
\]

and can be found in an iterative procedure.

The final parameters of the CS \( C_{j,\bar{j}} \) (which may be different from the temporal parameters set by node \( j \)) are determined by node \( \bar{j} \). Since \( \bar{j} \) determines the CS parameters for both \( j \) and \( \bar{j} \), it must 1) ensure CA transmission and reception of packets at node \( j \) and 2) allow CA transmission and reception of packets at node \( \bar{j} \). The first condition is satisfied by scheduling transmission of data packets within the limitations set by node \( j \). Since the first data packet is transmitted after \( j \) receives the CTS packet of \( \bar{j} \), we set

\[
s_{ij}^{\text{init}} = \bar{s}_{ij} + k_2 \times \bar{t}_{j,\bar{j}}, \tag{6.32}
\]

where \( k_2 = \left\lfloor \bar{s}_{ij}^{\text{RTS}} + d_{\text{CTS}} + T_{j}^{\text{rd}} - \bar{s}_{j,\bar{j}} / \bar{t}_{j,\bar{j}} \right\rfloor \). In addition, ensuring data packet transmission within the backoff time of node \( j \), we require

\[
s_{j,\bar{j}} \geq \bar{s}_{j,\bar{j}} + k_3 \times \bar{t}_{j,\bar{j}} \quad s_{j,\bar{j}} + d_{j,\bar{j}} \leq \bar{u} + k_3 \times \bar{t}_{j,\bar{j}} \tag{6.33}
\]

for at least one integer \( k_3 \in \{0, \ldots, \lfloor \bar{u} + T_{\text{offset}} - \bar{s}_{j,\bar{j}} / \bar{t}_{j,\bar{j}} \rfloor \} \). Since \( s_{ij}^{\text{CTS}} \) and \( s_{ij}^{\text{CTS}} \) are not restricted to the time proposed by node \( j \), bounds (6.31) and (6.33) offer some flexibility in scheduling CTS and data packet transmission, respectively, while satisfying primary conflicts of node \( j \).

The second condition is satisfied by calculating the bounds \( \bar{u} \) and \( \bar{l} \) in (6.23) not only for data packets transmitted from \( j \) but also for data packets transmitted from \( \bar{j} \). That is, \( \bar{j} \) calculates upper bound vectors \( \bar{u}_{1,\bar{j}} \) and \( \bar{u}_{1,\bar{j}} \), and lower bound vectors

129
\( \bar{t}_{1,j} \) and \( \bar{t}_{1,j'} \) with elements

\[
\begin{align*}
\tilde{u}_{1,j}(p) &= T^e_{\hat{p},p,n'_{p,j}} - T^p_{j,p} + 2t^{tol} \\
\tilde{u}_{1,j'}(p) &= T^e_{\hat{p},p,n'_{p,j'}} - T^p_{j,p} + 2t^{tol} \\
\tilde{l}_{1,j}(p) &= T^b_{\hat{p},p,n'_{p,j}+1} - T^p_{j,p} - 2t^{tol} \\
\tilde{l}_{1,j'}(p) &= T^b_{\hat{p},p,n'_{p,j'}+1} - T^p_{j,p} - 2t^{tol}
\end{align*}
\] (6.34)

\( \forall p \in \mathcal{R}_j, p \neq j, j', \) and bounds

\[
\begin{align*}
\tilde{u}_{2,j} &= T^b_{\hat{q},j,n''_{q,j}} - T^p_{j,j} - 2t^{tol} \\
\tilde{u}_{2,j'} &= T^b_{\hat{q},j,n''_{q,j'}} - T^p_{j,j'} - 2t^{tol} \\
\tilde{l}_{2,j} &= T^e_{\hat{p},j,n''_{p,j}} - T^p_{j,j} + 2t^{tol} \\
\tilde{l}_{2,j'} &= T^e_{\hat{p},j,n''_{p,j'}} - T^p_{j,j'} + 2t^{tol},
\end{align*}
\] (6.35)

for set \( \bar{\mathcal{R}}_j \subseteq \mathcal{I}_j \) including all neighbor CSs \( j' \) is aware of at time \( s_{j,j'}^{init} \). Then, the parameters of the CS \( C_{j,j} \) are determined at node \( j' \) by replacing bounds \( \bar{u} \) and \( \bar{l} \) with

\[
\begin{align*}
\hat{u} &= \min \left( \bar{u} + k_3 \times \bar{t}_{j,j'}, \tilde{u}_{1,j}, \tilde{u}_{1,j'}, \tilde{u}_{2,j}, \tilde{u}_{2,j'} \right) \\
\hat{l} &= \max \left( \bar{l} + k_3 \times \bar{t}_{j,j'}, \tilde{l}_{1,j}, \tilde{l}_{1,j'}, \tilde{l}_{2,j}, \tilde{l}_{2,j'} \right),
\end{align*}
\] (6.36)

and solving (6.23) to obtain \( s_{j,j'} \) and \( d_{j,j'} \). Next, \( t_{j,j'} \) is obtained by adding \( \tilde{t}_{j,j} \) to \( t_j \) and solving (6.24).

The final parameters of the CS \( C_{j,j} \) are available to \( j \) and to nodes in \( \mathcal{K}_j \cup \mathcal{K}_{j'} \) by piggybacking them, along with the number of data packets \( j \) and \( j' \) wish to transmit, \( N_{cycle,j} = \frac{T_{\text{msg}}}{T_{\text{pd}}} \) and \( N_{cycle,j'} \), respectively, on the CTS packet of \( j' \) and on the NT packet of node \( j \), which, as noted above, is appended to the first data packet of \( j \). Having both \( N_{cycle,j} \) and \( N_{cycle,j'} \), allows nodes to further utilize the channel when communication is asymmetric, as reflected in (6.14f). In addition, we allow nodes overhearing data packets to estimate the parameters of neighbor CSs even when control packets are lost. This is performed by detecting the arrival time, packet duration, and transmission cycle, of decoded data packets. However, when (control or data) packets from nodes outside the communication range but still affecting SINR are not detected, collisions may occur. This problem is not unique to the TSR and appears in any handshake-based protocol which relies on RTS/CTS exchange to alert nearby nodes (e.g., [68],[69]). To limit the effect of such interference, direct-sequence-spread-spectrum (DSSS) signals with different pseudo-random sequences allocated to each node (often used in UWAC to mitigate inter-symbol-interferences [9]) can be
used to significantly decrease the interference range. The structures of the RTS, CTS, NT, and data packets are shown in Figure 6.4. A flow chart describing the process of scheduling RTS and CTS packets, as well as determining the CS \( C_{j,j} \) parameters, is offered in Figure 6.5. Furthermore, a software implementation of the TSR protocol can be downloaded from [96].

Finally, a word on complexity and communications overhead of the TSR protocol is in order. While the steps of our protocol may seem complex, we note that both \( j \) and \( \tilde{j} \) schedule the parameters of the CS \( C_{j,j} \) by solving a number of closed form equations, using up to three iterative processes. In addition, for each CS, the communications overhead is limited to transmission of the CS parameters piggyback on the RTS, CTS and NT control packets.

### 6.4 Results

For the purpose of comparing the performance of the TSR protocol to benchmark protocols we choose the TDMA, Slotted floor acquisition multiple access (FAMA) [60], adaptive propagation-delay-tolerant MAC protocol (APCAP)\(^{18} [68] \), and the bidirectional concurrent MAC (BiC-MAC) [69] protocols. The TDMA is an interference-free protocol best suited for static networks with heavy load, while Slotted-FAMA combines TDMA and handshake-based scheduling to allow some network dynamics. We choose the APCAP and BiC-MAC protocols for their usage of timing-advance techniques, utilizing the long propagation delay in the channel. The source code of our implementation of the three benchmark protocols as well as the TSR protocol is available from [96]. We compare the four protocols in terms of throughput (6.2),

\(^{18}\)We note that due to high collision rate in APCAP occurring when a node overhears another RTS packet while waiting for CTS response, we had to slightly modify APCAP and rather than transmitting the data message according to the time suggested by the destination node, we also considered scheduling constraints known to the source at the time CTS is received.
Chapter 6. Time and Spatial Reuse Handshake Protocol for UWANs

![Diagram]

Figure 6.5: Scheduling of RTS and CTS and determining of CS parameters in the TSR protocol.
Chapter 6. Time and Spatial Reuse Handshake Protocol for UWANs

fairness (6.3), and scheduling delay (6.4).

6.4.1 Simulation Setting

In our simulations, we generated a Monte-Carlo set of 10000 channel realizations, where for each one, $N = \{6, 10, 15\}$ nodes are uniformly placed in a square area of $5 \times 5 \text{ km}^2$ at a fixed depth of 40 m for a 50 m long water column. In each realization, the channel includes four horizontal obstacles and one vertical obstacle at uniformly distributed positions and with lengths uniformly distributed in $[100, 200]$ m. For each node pair, we calculate the Euclidian distance, and set transmission loss to infinite if an obstacle blocks the line-of-sight. A packet is decoded correctly if the SINR is such that the packet error probability (with no error-correction-code) is below $10^{-4}$ and a BPSK modulation is considered, otherwise collisions occur and all colliding packets are lost. The SINR is calculated taking into account all signals arriving to the receiver. We regard a packet as lost if its destination node experiences interference (of any portion) from another packet, i.e., we look at the worst-case SINR over the period of the packet. In addition, we consider transmission of a data message as successful only if all its packets are successfully received by the destination node. The received power of each packet is calculated for a common power source level of 155 dB/µPa@1m, and using the Bellhop ray-tracing model (cf. [2]) for a flat sand surface, carrier frequency of 15 kHz, and sound speed of 1540 m/sec. For the above numbers, and based on the Bellhop attenuation model, the obstacle- and interference-free maximum transmission range is about 2 km. Note that this range may decrease due to effect of interferences on SINR.

To show the effect of long interference range compared to the communication range, we present simulation results with and without the use of DSSS signaling. For the former, we use different pseudo-random sequences with $L_c = 15$ chips allocated to each node, and consider a cross correlation factor of $\frac{1}{L_c}$ between arriving signals, which limits the interference range. We measure the network performance for a fixed time interval of $W = 1000$ seconds, during which each node is assigned with original data messages of length 100 kbit to be transmitted to one of its one-hop neighbor nodes (in smaller packets of size $d_{j,j}$), and the arrival of messages is modeled as a Poisson process with a mean arrival rate of $0.01 \frac{1}{\text{sec}}$, which is regenerated for each channel realization and for each node. This results in the transmission of on average 60 data messages for each channel realization. Such messages can accommodate any small-scale image file (e.g., the Deep-Link application) or reasonable set of collected data (e.g., the Kintama application). If a node $j$ has more than one neighbor node, destination node, $\tilde{j}$, is chosen uniformly at random for each message.

Following the list of commercial UWAC modems in [9], we consider a transmission rate of 10 kbps, and use a data packet header of 500 bits (mostly for the preamble sequence, e.g., in the Evologics modem [9]), such that, for TSR, the average ratio between the number of header and information bits per data packet (see Figure 6.4)
Figure 6.6: Average of $\rho_{\text{through}}$ from (6.2) and $\rho_{\text{delay}}$ from (6.4) as a function of $R_{\text{msg}} = \frac{T_{\text{msg}}}{T_{\text{msg},j;j}}$ for TSR with and without DSSS.

was measured to be about 10%. Note that, while it is not always the case, $\tilde{j}$ might also have a message to transmit to node $j$. For TDMA, we choose time slot duration of $T_{\text{slot}} = 6.2$ sec, which for the maximal range of $\sqrt{2} \times 5$ km, allows data transmission of 1.6 sec per time slot. For the TSR protocol, we assume a perfect propagation delay estimation (i.e., $t_{\text{tol}} = 0$), and use $T_{\text{INR}} = 0$ dB for the INR threshold. The source code for the simulation environment is also available at [96].

### 6.4.2 Simulation Results

First, we consider two-way communication and explore the effect of asymmetric transmission and interference range on throughput and delay of TSR. In Figure 6.6, we show the average of $\rho_{\text{through}}$ from (6.2) and of $\rho_{\text{delay}}$ from (6.4) as a function of the symmetry rate, $R_{\text{msg}} = \frac{T_{\text{msg}}}{T_{\text{msg},j;j}}$, both with and without the use of DSSS signaling. Both throughput and delay performance increase when DSSS is used to limit interference range. This is because shorter interference range result in higher network sparsity, and thus potential higher gain from spatial reuse. Moreover, from Figure 6.6, we observe that naturally, since amount of transmitted data increases with $R_{\text{msg}}$, so is delay. However, since TSR aims to maximize channel utilization, throughput also increases with $R_{\text{msg}}$. In the following, we consider common message duration for all nodes for two-way or one-way communication, where the former occurs opportunistically when both $j$ and $\tilde{j}$ have a message to transmit to each other.

Figure 6.7 shows the empirical C-CDF of $\rho_{\text{fair,through}}$ from (6.3) for the five simu-
Chapter 6. Time and Spatial Reuse Handshake Protocol for UWANs

Figure 6.7: Empirical C-CDF of $\rho_{\text{fair,through}}$ in (6.3) for the TSR, TDMA, Slotted-FAMA, BiC-MAC, and APCAP protocols. $|\mathcal{N}| = 6$

...lated protocols. As expected, TDMA, which allocates equal number of time slots for each node, is the fairest protocol. From the handshake-based protocols, we observe that the fairness indices of the TSR and the BiC-MAC protocols are higher than those of the Slotted-FAMA and APCAP. This is due to the time reuse in both protocols that allows both nodes $j$ and $\tilde{j}$ to transmit during the CS $C_{j,\tilde{j}}$, while Slotted-FAMA and APCAP give priority to the node which first reserves the channel. We observe that the fairness index of the TSR is lower than that of BiC-MAC. This is because the low node density in our simulation model causes large variations in propagation delays, thus variation of the elements of $t_j$ is large. Since the CS cycle is proportional to the LCM of $t_j$ [see (6.24)], in TSR, fewer CSs could occur simultaneously, affecting fairness.

Figure 6.8 shows the empirical CDF of $\rho_{\text{delay}}$ from (6.4) for the five protocols. We note that the delay shown in the x-axis is normalized by $T_{\text{msg}}^j$. Although TDMA is interference-free, its delay is significantly higher. This is because, in TDMA, each message is divided into 6 time slots, transmitted $|\mathcal{N}|T_{\text{slot}}$ apart. Due to the relatively long time slot, delay is also high in Slotted-FAMA, which channel reservation process depends on TDMA. We observe that the scheduling delay of the TSR is considerably lower than those of the BiC-MAC and APCAP protocols. This is due to the time and spatial reuse in the TSR protocol that allow simultaneous transmission of neighbor CSs.

Next, we compare $\rho_{\text{through}}$ from (6.2) for the five protocols. Figure 6.9a shows empirical C-CDF results without using DSSS signaling. We observe that performance of TDMA, which includes many empty time slots if network traffic is low, is the lowest.
Figure 6.8: Empirical CDF of $\rho_{\text{delay}}$ in (6.4) for the TSR, Slotted-FAMA and BiC-MAC protocols. $|\mathcal{N}| = 15$.

Furthermore, APCAP, which allows simultaneous channel reservation for neighbor CSs, achieves better throughput performance than Slotted-FAMA, but lower performance than BiC-MAC for high throughput values. The latter is because at higher network traffic rates (and potentially higher throughput), there is a higher probability for the destination node to have a message to transmit to the source node, which in BiC-MAC, can be done simultaneously. We observe that throughput results of TSR outperform those of the benchmark protocols, except for very low throughput values. This is due to the channel utilization maximization performed by the TSR protocol. Such channel utilization is possible in TSR by dividing messages into data packets. While such division increases the overhead due to the packet header (see Figure 6.4) on throughput, it has the positive effect of possible simultaneous transmissions from nearby nodes. To quantify this, in Figure 6.9b we show the empirical C-CDF results for $\rho_{\text{through}}$ when DSSS signaling is used, and thus interference range is much shorter (usually shorter than the communication range). From Figures 6.9a and 6.9b, we observe a large variance for the results of the TSR protocol. This is mainly because, being a greedy suboptimal time- and spatial-reuse protocol, the performance of TSR depends on the network topology and the order of incoming packets. Comparing results of Figures 6.9a and 6.9b, we observe that results are the same for the Slotted-FAMA and the TDMA protocols. We also observe, that performance improves for the APCAP and BiC-MAC protocols, mainly since using these protocols, packet collisions occur when interference range is longer than the communication range. However, the most significant improvement is observed in the TSR protocol, which due to the shorter interference range, can simultaneously schedule more CSs.

Finally, we evaluate effect of network size. Let $\rho(W, |\mathcal{N}|)$ be a performance mea-
Figure 6.9: Empirical C-CDF of $\rho_{\text{through}}$ from (6.2) for the TSR, TDMA, Slotted-FAMA, APCAP, and BiC-MAC protocols. $|\mathcal{N}| = 6$. (a) without DSSS signaling, (b) with DSSS signaling.

which serve as an indicator of the scalability of the different protocols, and optimally, $\rho_{\text{change}} \approx 1$. From Figure 6.10, we observe that similar to the results shown in Figure 6.7, the effect of $|\mathcal{N}|$ on the fairness index is lowest using TDMA and Slotted-FAMA. However, comparing results for throughput and delay, we observe that for TSR $\rho_{\text{change}}(|\mathcal{N}| = 6, 10, 15)$ is close to 1, and the effect of the network size is much lower than for the other four protocols. Thus, in terms of scalability, TSR is close to optimum [62].

From the results of our simulations we observe that at the cost of fairness in resource allocation, throughput and scheduling delay are considerably improved using the TSR compared to the TDMA, Slotted-FAMA, APCAP, and the BiC-MAC protocols. Furthermore, in TSR we observe a much smaller effect of the number of network nodes on both throughput and delay. Comparing the results shown in Figure 6.6 to those in Figures 6.8 and 6.9a, this performance gain of TSR is achieved even when communication is asymmetric. Our protocol is thus an effective solution that trades off fairness with throughput and delay. The improvement in throughput and scheduling delay are achieved by the TSR protocol at only a small cost of communications overhead, as only the CS parameters are exchanged between the transmitting nodes, and the protocol is fully distributed.
6.5 Conclusions

In this chapter, we considered the problem of designing a handshake-type protocol for UWAN supporting CA unicast communications. We formalized the problem of resource assignments to nodes to maximize the per-link channel utilization while avoiding mutual access interference within the communication range, and suggested a sub-optimal distributed protocol to solve it. Our protocol combines spatial reuse and timing advance techniques to utilize the long propagation delay in the channel and the expected sparsity of the network graph. We described the process of channel reservation and distributed scheduling while keeping communications overhead at a minimum. By means of simulation results, we demonstrated that at the cost of fairness in resource allocation for large transmission ranges, our protocol outperforms existing handshake protocols in terms of per-node throughput and scheduling delay.
Chapter 7

Robust Spatial Reuse Scheduling in UWANs

To complete our solution to transmission scheduling in UWANs, we now consider the problem of resource assignment through broadcast scheduling in UWANs which support frequent transmission of broadcast packets, i.e., packets that need to be received by all nodes in the network. This is required for sharing of navigation information, simultaneous control of several systems, sending distress signals, etc. A practical example is the "Deep-Link" system [133], which supports a network of up to 15 nodes for command and control and divers safety purposes and includes periodic transmission (once every minute) of broadcast packets including localization coordinates. Another specific example is transmission of data from moving AUVs to a surface station. In this scenario, each node broadcasts its packets to all its neighbors which on-the-fly decide whether to relay these packets.

Considering the problem of low channel utilization of contention-based scheduling algorithms for high-rate UWANs, we propose a spatial-reuse TDMA scheduling algorithm. Spatial-reuse scheduling algorithms assume accurate topology information, such that possible network sparsity is used to increase throughput by allowing simultaneous transmissions (e.g., [136]). However, this assumption may be violated in UWANs, where node movements render time-varying topology or topological information is temporary unavailable due to high packet loss rate. As a result, several nodes might temporary hold conflicting topology information leading to packet collisions. Previous works on wireless mesh networks attended to this problem by developing algorithms to ensure fast propagation of topology variations across the network (e.g., [137]). Unfortunately, in UWAC such an approach might be too slow due to the long propagation delay. Uncertainty of topology information in topology-dependent scheduling can be regarded as a problem of robustness since we require a certain minimum performance to be achieved even under topology mismatch. The authors of [138] addressed this problem making the assumption that a probabilistic model for the uncertain topology parameters is available. A different approach is the use of topology-transparent scheduling which does not depend on the instantaneous network topology. Topology-transparent scheduling was pioneered in [139], which suggested a schedule based on the maximal degree of the network graph and an upper bound on the number of conflicts between any two nodes regardless of network topology. Cai et al. [140] generalized this algorithm and reduced the number of conflicts.

In this chapter, our goal is to reconcile the seemingly conflicting requirements
Chapter 7. Robust Spatial Reuse Scheduling in UWANs

of high channel-resource utilization and robustness to network-topology information uncertainty. To this end, we combine the concepts of topology-transparent and topology-dependent scheduling. The former element ensures that topology information mismatch does not cause an uncontrolled amount of packet collisions. The latter component allows us to make use of additional spatial reuse in case of reliable topology information. To guarantee delivery of broadcast packets in (possibly) sparse networks, our algorithm includes packet flow control. Flow control also integrates a certain amount of fairness into our system by ensuring that improved aggregate network throughput reflects in increased throughput for each node. We present simulation results for typical UWAC environments as well as for network topologies recorded in sea trial experiments, and compare the performance of our algorithm with that of two spatial-reuse topology-dependent algorithms and two topology-transparent algorithms for both fixed and time-varying topologies. The results demonstrate that our algorithm provides a favorable tradeoff between network throughput and robustness to outdated topology information due to topology changes, while also achieving fairness in terms of per-node throughput. The motivation for our algorithm is topology uncertainties, which are common in UWAC due to permanent motion of nodes and phenomena like shadowing and transient ambient noise [5].

The remainder of this chapter is organized as follows. System model and design objectives are introduced in Section 7.1. In Section 7.2, we first formalize the BSP and present a topology-dependent scheduling algorithm, based on which we develop the proposed mixed topology-transparent/dependent approach. In Section 7.3, we describe our approach to obtain the topology and conflict matrixes. Simulation results are presented and discussed in Section 7.4, and conclusions are drawn in Section 7.5.

7.1 Preliminaries

In this section, we introduce the system model and the objectives for resource allocation considered in this work.

7.1.1 System Model

We consider UWANs with a fixed small to moderate number of nodes \( N \), say \( N < 50 \), distributed over an area of a few square kilometers. Each node is given a unique identification number and can be a source, relay, or destination node for a given message. We require only a coarse periodic time-synchronization between the nodes to establish a network-wide TDMA frame structure; that is, node clock offset and skew should be negligible compared to the propagation delay, which is on the order of 1 to 3 seconds for distances of 1 to 4 km. The applications supported by the UWAN generate high network traffic in the form of periodic broadcast packets (i.e., messages from a single node to all other nodes) relayed by a common routing mechanism across
Chapter 7. Robust Spatial Reuse Scheduling in UWANs

the network (e.g. minimal hop-distance, greedy routing, etc.). We further assume heavy network traffic, and link this model to the Deep-Link system, where nodes periodically share their location information with the rest of the network nodes.

7.1.2 Objectives of Resource Allocation

We measure the performance of our scheduling algorithm in terms of throughput, scheduling delay, and fairness in resource allocation, as described in the following.

Throughput

Define $y_{i,j}(T)$ as the number of broadcast packets originated by node $i$ and received by node $j$ in $T$ time slots, referred to as original packets. Considering broadcast packets, the throughput of node $i$ is given by

$$\rho_{\text{through, node}}(i) = \frac{1}{T(N-1)} \sum_{j=1, j\neq i}^{N} y_{i,j}(T),$$  (7.1)

where we assume that the observation window $T$ is much larger than the TDMA frame length $L$. The per-node throughput is defined as

$$\rho_{\text{through}} = \frac{1}{N} \sum_{i=1}^{N} \rho_{\text{through, node}}(i).$$  (7.2)

Scheduling Delay

We define scheduling delay as the delay between the time an original packet (i.e., not routed) is delivered to the MAC layer at its source and the time it is received at its destination (which may be several hops away). For the latter, since we consider broadcast packets, we take the average reception time. As such, scheduling delay includes the end-to-end transmission and queuing delay, and the waiting time for a transmission time slot. To formalize this, let $T_{\text{schedule}}(n, j, i)$ be the scheduling delay (measured in number of time slots) for message $i$, $i = 1, \ldots, M_n$, transmitted from source $n$ to destination $j$, where $M_n$ is the total number of messages transmitted by node $n$. The average scheduling delay for the network is then expressed by

$$\rho_{\text{delay}} = \frac{1}{N(N-1)} \sum_{n=1}^{N} \sum_{j=1, j\neq n}^{N} \frac{1}{M_n} \sum_{i=1}^{M_n} T_{\text{schedule}}(n, j, i)$$  (7.3)

time slots.
Chapter 7. Robust Spatial Reuse Scheduling in UWANs

**Fairness**

Since we assume nodes can always transmit a packet, fairness is measured by comparing the differences in node-wise throughput. More specifically, we apply the widely used Jain’s fairness index [134] to throughput (7.1) and measure fairness by

\[
\rho_{\text{fair,through}} = \frac{1}{N} \left( \frac{\sum_{i=1}^{N} \rho_{\text{through, node}}(i)}{\sum_{i=1}^{N} [\rho_{\text{through, node}}(i)]^2} \right)^{\frac{2}{3}}. \tag{7.4}
\]

Towards the goal of achieving fairness, we formalize resource-allocation constraints that guarantee node \( i \) a minimal number of \( d_i \) time slots per TDMA frame for transmission [81]. That is, defining \( x_i \) as the number of time slots assigned to node \( i \) within a TDMA frame, we have

\[ x_i \geq d_i , \quad i = 1, \ldots, N . \tag{7.5} \]

Since our algorithm should guarantee delivery of broadcast packets in (possibly) sparse networks, we include constraints to control flow of packets in the network. To this end, given multihop routes established by a routing algorithm, constraint (7.5) can be used to respect traffic-flow control and thus avoid bottleneck nodes. Specifically, by defining routing variables \( \theta_{i,j} \), \( i, j = 1, \ldots, N \), with \( \theta_{i,i} = 1 \) if node \( i \) is a relay for node \( j \)’s broadcast packets and \( \theta_{i,j} = 0 \) otherwise (note that \( \theta_{i,i} = 1 \)), and

\[ d_i = \sum_{j=1}^{N} g_j \theta_{i,j} , \tag{7.6} \]

node \( i \) is guaranteed to transmit \( g_i \) original packets per TDMA frame through constraint (7.5)\(^\text{19}\). Note that both \( g_i \) and \( d_i \) are defined per frame and thus do not represent delay requirements but rather fairness constraints. Consequently, only after transmitting \( g_i \) original packets, and relaying awaiting packets in its routing queue, node \( i \) may utilize its additional time slots in the TDMA frame and broadcast additional original packets.

Next we formalize the BSP to optimize our objectives under the above fairness/flow constraint.

\(^{19}\)Note that if \( \theta_{i,j} \) is not given (e.g., AUVs reporting to a surface station), flow control is preserved by pre-defining parameters \( d_i \).
Chapter 7. Robust Spatial Reuse Scheduling in UWANs

7.2 Formalizing the BSP

7.2.1 Basic Approach

Before getting into the details, in the following we describe the basic approach of our algorithm. We are interested in a resource allocation algorithm, for which flow control (7.5) is preserved, and a favorable tradeoff between throughput (7.2) and scheduling delay (7.3) is achieved making use of topology information. This is accomplished through tracking topology changes and applying a topology-dependent schedule which adapts to these changes. Considering the problem of slow propagation of topology information in an UWAN of moving nodes, and thus the occurrence of outdated topology information, our algorithm should accommodate such topology uncertainties while nodes update their topology. This is required to avoid cases where, due to topology uncertainties, the network collapses and is unable to recover from a topology change. We refer to this objective as short-term robustness. The result is an algorithm that adjusts to topology changes to improve performance while reducing scheduling conflicts in the case of temporarily outdated topology information.

The output of our algorithm is an \( N \times L \) spatial-reuse TDMA (STDMA) binary scheduling matrix \( M \) such that node \( i, 1 \leq i \leq N \), is allowed to transmit a single packet in time slot \( j, 1 \leq j \leq L \), if and only if \( M_{ij} = 1 \), and \( L \) is the number of time slots in each TDMA frame. Flow control is enforced through pre-defined \( g_i \) in (7.6) or \( d_i \) in (7.5) to achieve fairness in resource allocation (7.4). Note that since both \( L \) and \( \theta_{i,j} \) from (7.6) are time-varying, so is matrix \( M \). Thus, \( M \) is re-calculated each time a topology change is detected.

Our algorithm is general in the sense that it can operate for a variety of multiple packet reception (MPR) scenarios. This includes, for example, systems with ideal MPR that can resolve all collisions of received packets (due to DSSS), MPR up to a maximal number of simultaneously received packets, MPR depending on received signal strength for packets from different users (to account for near-far effects), and systems without MPR. Each of these scenarios gives rise to a conflict graph used in our scheduling algorithm. The conflict graph is represented as an \( N \times N \) binary matrix \( Q \), with elements \( q_{i,j} \), such that \( q_{i,j} = 1 \) if and only if nodes \( i \) and \( j \) must not transmit simultaneously. For example, usually acoustic transducers are half-duplex, and therefore, since we consider broadcast packets, we do not allow nodes which are connected through one-hop links to transmit simultaneously, i.e., we respect primary conflicts [77] and set \( q_{i,j} = 1 \) when nodes \( i \) and \( j \) are one hop apart\(^{20}\).

Since both \( \rho_{\text{through}} \) and \( \rho_{\text{delay}} \) depend on the network topology, it is hard to optimize them directly for broadcast packets. However, since in our algorithm flow control is treated as a problem constraint, sufficient time slots for relaying incoming packets are guaranteed.

\(^{20}\)We note that this constraint can be relaxed for non-broadcast packets or if the long propagation delay in the channel is utilized for simultaneous transmission and reception. However, the latter would require tracking the time-varying location of each node.
multihop packets are available, and channel utilization and thus availability of time slots to nodes is proportional to throughput. In addition, for a given \( L \), channel utilization is inversely proportional to scheduling delay. Therefore, following, e.g., [82] and [81], instead of direct optimization of \( \rho_{\text{through}} \) and \( \rho_{\text{delay}} \), we maximize channel utilization defined as

\[
\rho_{\text{avail}} = \frac{1}{NL} \sum_{i=1}^{N} x_i .
\] (7.7)

We account for topology variations due to both link instabilities, i.e., flickering and node movements. The former is dealt with by conservative approach to determine network topology, and for the latter we combine topology-transparent with topology-dependent schedules. The input to our algorithm is the time-varying network topology, which for a given MPR model, is used to construct the conflict matrix, \( Q \). The more limited the MPR, the less sparse is \( Q \), and channel utilization decreases. Since our system requires broadcast scheduling to cope with high network packet transmission rate, and flow control to realize successful delivery of broadcast packets, possibly with delay constraints\(^{21}\), we modify the BSP described in [82, 81] and [80] to include flow control. In the following, we first consider the maximization of (7.7) under constraint (7.5) assuming network topology information is available, which we refer to as the topology-dependent BSP (T-BSP). This not only provides a benchmark case, but the formulation we propose also sets the stage for the robust BSP (R-BSP) presented thereafter.

### 7.2.2 Formalizing the T-BSP

Following [81] and towards trading throughput (7.2) with scheduling delay (7.3), we first minimize the frame length \( L \) under fairness/flow constraints and then maximize channel utilization (7.7) for a given \( L \). Different from [81], which considered \( d_i = 1 \forall i \) in (7.5), we avoid bottleneck nodes by allowing possibly different minimal numbers of packet transmission per-TDMA frame as determined by (7.6). Furthermore, while [81] obtained a collision-free schedule through problem constraints, we avoid scheduling conflicts by first constructing all feasible time slot allocations and then choosing the sequence of allocations that leads to maximum channel utilization. This structure of successively solving two sub-problems to obtain the T-BSP schedule allows us to readily combine topology-dependent and topology-transparent scheduling in the R-BSP formulation considered next.

First, based on the conflict matrix \( Q \), we find the set of possible node assignments to a time slot by the columns of the binary matrix \( I \in \{0,1\}^{N \times K} \) such that for column \( j \), \( I_{n,j} = I_{m,j} = 1 \) only if \( q_{n,m} = 0 \), i.e., nodes \( n \) and \( m \) can transmit in the same time slot, where \( K \) is the number of possible such node assignments. The T-BSP

\(^{21}\)For example, the Deep-Link system [133], requires that navigation information is shared across the network at least once in every minute.
is then solved by associating each column $j$ of $I$ with a non-negative integer $a_j$, representing how many times each combination of node assignments is used in the resulting schedule $M$. Mathematically, defining vector $a = [a_1, \ldots, a_K]^T$, the vector $x = [x_1, \ldots, x_N]^T$ of the number of time slots within a frame assigned to nodes is obtained by

$$x = Ia.$$  \hspace{1cm} (7.8)

Thus, (7.7) can be rewritten as ($1$ denotes the all-one column vector of appropriate length)

$$\rho_{\text{avail}} = \frac{1}{NL} 1^T I a.$$  \hspace{1cm} (7.9)

Now, to minimize the frame length $L$ given constraint (7.5) we formulate the minimum-frame-length problem (MFLP)

$$\begin{align*}
\min_a & \quad 1^T a \\
\text{s.t.} & \quad Ia \geq d \\
& \quad a \in \mathbb{N}^K,
\end{align*}$$  \hspace{1cm} (7.10)

whose solution $a^{MFLP}$ gives the frame length

$$L = 1^T a^{MFLP}.$$  \hspace{1cm} (7.11)

Finally, defining the vector of traffic demands $d = [d_1, \ldots, d_N]^T$ with $d_i$ from (7.5) and using $L$ from (7.11), the channel-utilization-maximization problem (CUMP) can be written as

$$\begin{align*}
\max_a & \quad 1^T Ia \\
\text{s.t.} & \quad Ia \geq d \\
& \quad 1^T a \leq L, \\
& \quad a \in \mathbb{N}^K.
\end{align*}$$  \hspace{1cm} (7.12)

The sequence of (7.10) and (7.12) solves the T-BSP. More specifically, if $a^{\text{CUMP}}$ is the solution of (7.12), the scheduling matrix $M$ is constructed from columns of $I$ with $a_j^{\text{CUMP}}$ being the number of times the $j$th column $I_j$ is used:

$$M = \left[ \begin{array}{c|c|c}
I_1 & \ldots & I_1 \\
\hline
\ldots & \ldots & \ldots \\
I_K & \ldots & I_K \\
\end{array} \right].$$  \hspace{1cm} (7.13)

We note that both (7.10) and (7.12) are instances of the cutting-stock problem (CSP), for whose solution optimized numerical methods exist, e.g., [142].

Inspecting (7.12) and (7.10), we observe that we can restrict the $K$ possible patterns as follows.
Lemma 1 Considering the graph representation of the UWAN with nodes being vertices and communication links being edges of the graph, the only possible node assignments according to the MFLP and CUMP are maximal independent sets\(^{22}\) (MISs) of this graph.

Proof 1 Let us consider two possible node assignment patterns \(p_1\) and \(p_2\), with \(p_2 - p_1 = e_i\), for some \(i \in \{1, 2, \ldots, N\}\), and \(e_i\) is the unit column vector whose elements are all zero except the \(i\)th element which is equal to 1. Using \(p_2\) instead of \(p_1\) as a column vector of \(I\) does not decrease the left-hand side of constraints (7.10b) and (7.12b) and the objective (7.12a). Hence, using \(p_2\) can only improve the solution compared to using \(p_1\). Since because of scheduling constraints in \(Q\), all possible patterns are independent sets of the graph specified by the conflict matrix \(Q\), it is sufficient to consider independent sets with maximal cardinality, i.e., MISs, as columns of \(I\).

Hence, to solve the T-BSP we first construct the MIS matrix \(I\), using, for example, the algorithm described in [143]. Next, we enforce flow control using constraints (7.6), where \(\theta\) is given by the routing mechanism. Finally, we form the CSPs (7.10) and (7.12) and solve them using, e.g., a branch-and-bound solver [142].

Based on the above formalization of the T-BSP, we now proceed by combining topology-dependent with topology-transparent scheduling to enhance short-term robustness to topology uncertainties.

7.2.3 Formalizing the R-BSP

While using T-BSP scheduling in UWANs is promising for increasing channel utilization and decreasing scheduling delay due to spatial reuse, it is sensitive to erroneous topology information (as we will demonstrate by simulation results in Section 7.4). For example, consider two connected nodes that do not share the same frame length \(L\) due to flawed topology information. The result would be an almost random schedule and thus catastrophic packet collisions. Hence, a topology-independent frame length is necessary to achieve short-term robustness to uncertain topology information. To combine the benefits of both topology-transparent and topology-dependent schedules, we suggest a new approach that combines an underlying skeleton schedule with a fixed frame length \(L_{skel}\) obtained from a topology-transparent schedule with the use of topology information.

Combining Topology-Transparent with Topology-Dependent Scheduling

We start from a topology-transparent schedule with an \(N \times L_{skel}\) skeleton matrix \(S\), a-priori known to all nodes, whose element \(S_{i,j} = 1\) if and only if node \(i\) transmits

\(^{22}\)An independent set is a collection of vertices/nodes that can simultaneously transmit with no scheduling conflicts. A maximal independent set is an independent set with maximal size.
in time slot $j$ (specific examples for $S$ will be given in Section 7.2.3). First, for each column $j$ of matrix $S$ we identify a unique *slot node* $r(j)$. Slot nodes serve as reference nodes for conflict removal and will always transmit in their time slots. For this purpose, we rearrange $S$ to form a matrix $S_{\text{mod}}$ such that $S_{r(j),j} = 1$. We note that the assignment of slot nodes should be fair, and ideally nodes will be selected equally often as slot nodes. Algorithm 4 shows the pseudo-code of an $L_{\text{ske}}$ step process in which slot nodes are selected in a round robin fashion to form $S_{\text{mod}}$. In the $l$th step of Algorithm 4 slot node $r(l)$ of the $l$th column of $S_{\text{mod}}$ is determined. To achieve fairness we start with $i = l \mod N$ (line 5) and continue with $i = (i + 1) \mod N$ (line 15) until we find a column $j$ in $S$ which was not already assigned a slot node and for which $S_{i,j} = 1$ (line 8). This column becomes the $l$th column of $S_{\text{mod}}$ (line 9) and we set $r(l) = i$ (line 10). The fairness property of this algorithm will be demonstrated by numerical results in Section 7.4. Note that this rearrangement of $S$ into $S_{\text{mod}}$ does not affect channel utilization (7.9) of the topology-transparent schedule and is performed a-priori as it does not require topology information.

\begin{algorithm}
\caption{Rearranging topology-transparent schedule $S$ into $S_{\text{mod}}$}
\begin{algorithmic}
  \State $\mathcal{U} := \emptyset$
  \For {($l := 1$ to $L_{\text{ske}}$)}
    \State \{Determine the slot node\}
    \State $\text{FLAG} := 0$
    \State $i := l \mod N$
    \While {$\text{FLAG} = 0$}
      \For {($j := 1$ to $L_{\text{ske}}$)}
        \State if ($S_{i,j} = 1$ and $j \notin \mathcal{U}$) then
          \State $S_{l}^{\text{mod}} := S_{j}$ \{ $S_{l}^{\text{mod}}, S_{j}$ are the $l$th and $j$th columns of $S_{\text{mod}}, S$, respectively \}
          \State $r(l) := i$
          \State $\mathcal{U} := \{\mathcal{U}, j\}$
          \State $\text{FLAG} := 1$
        \EndIf
      \EndFor
      \State $i := (i + 1) \mod N$
    \EndWhile
  \EndFor
\end{algorithmic}
\end{algorithm}

Based on the rearranged skeleton matrix $S_{\text{mod}}$ and the scheduling constraints of the conflict graph matrix $Q$, a matrix $I_{\text{ske}}$ is constructed which replaces matrix $I$ in the CUMP (7.12). This procedure, whose pseudo-code is shown in Algorithm 5, makes use of topology information (which, if erroneous, leads to an increased collision rate, but it does not cause a collapse of the schedule due to the underlying topology-transparent skeleton schedule and the pre-defined choice of slot nodes). Considering
Chapter 7. Robust Spatial Reuse Scheduling in UWANs

the $j$th column of $S^{\text{mod}}$, a list $T_j$ consisting of all (not necessarily maximal) independent sets of the network graph, satisfying all scheduling constraints in $Q$, which include $r(j)$ is formed (line 2). To increase channel utilization while preserving as much as possible the structure of the topology-transparent schedule, the independent sets in $T_j$ which include the largest number of pre-assigned nodes, $T_j^{\text{mod}}$, are appended to $I_{\text{ske}}$ (lines 3, 4 and 6). Since there may be multiple independent sets chosen from $T_j$ for each column $j$ of $S^{\text{mod}}$, (i.e., $|T_j^{\text{mod}}| \geq 1$), the number of columns of $I_{\text{ske}}$, $K_{\text{ske}}$, is possibly larger than $L_{\text{ske}}$. Since the R-BSP draws its short-term robustness from the a-priori chosen frame size $L_{\text{ske}}$, to maintain the deterministic frame size, only one set of each $T_j^{\text{mod}}$ can be chosen to form the scheduling matrix $M$. This can be formulated by the constraint

$$Aa = 1,$$

(7.14)

where $A$ is an $L_{\text{ske}} \times K_{\text{ske}}$ 0/1-matrix such that $A_{n,m} = 1$ only if the $m$th column of $I_{\text{ske}}$ was derived from the $n$th column of $S^{\text{mod}}$.

**Algorithm 5** Determine $I_{\text{ske}}$ from the skeleton schedule $S^{\text{mod}}$ and conflict matrix $Q$

1: for $(j := 1 \text{ to } L_{\text{ske}})$ do
2: $T_j :$ all independent sets satisfying scheduling constraints in $Q$, which include $r(j)$
3: $P_j :$ all nodes $i$ for which $S_{i,j}^{\text{mod}} = 1$
4: $T_j^{\text{mod}} :$ sets from $T_j$ that include the largest number of nodes from $P_j$
5: end for
6: $I_{\text{ske}} := \{T_1^{\text{mod}}, \ldots, T_{L_{\text{ske}}}^{\text{mod}}\}$

The example shown in Figure 7.1 illustrates the described process. We consider a network of $N = 6$ nodes represented by the undirected graph in Figure 7.1(a). The skeleton matrix $S$ is selected according to the orthogonal topology-transparent (OTT) schedule, discussed in the introduction and described in [140] (see Section 7.2.3 for further details), where the shaded entries represent slot nodes. It is first rearranged into $S^{\text{mod}}$ using Algorithm 4, such that the columns 1 to 9 in $S^{\text{mod}}$ are columns 1,2,3,6,4,5,9,7, and 8 in $S$, as shown in Figure 7.1(b). Then, $I_{\text{ske}}$ is formed by expanding the columns of $S^{\text{mod}}$ utilizing the topology information according to Algorithm 5. Note that $I_{\text{ske}}$ is recalculated at each node every time a topology change is detected and would change accordingly. For the topology in Figure 7.1(a), there are two possible expansions for columns 5 and 8 of $S^{\text{mod}}$. Finally, Figure 7.1(c) shows the masking matrix $A$ for this example.

**A Suboptimal R-BSP**

Before proceeding with the R-BSP, we deviate to present a spatial reuse suboptimal schedule, referred to as the hybrid spatial-reuse time-division multiple access (HSR-
Chapter 7. Robust Spatial Reuse Scheduling in UWANs

Figure 7.1: Example: (a) sample topology for a UWAN. (b) constructing matrix $I^{\text{skel}}$ using Algorithms 4 and 5 for the sample network. (c) masking matrix used in (7.14).

TDMA), which implement the concept of a skeleton schedule. This would help to develop the intuition required to formalize the R-BSP. This algorithm has two distinct advantages: (i) it has polynomial worst case complexity (rather than exponential complexity as the optimal solution), and (ii) it is amenable to analytical performance evaluation.

We start from a skeleton TDMA schedule with node $i$ being the slot node in time slot $t = i$. Then, additional nodes are added to each time slot. For this purpose, each edge is assigned a unit weight and the shortest-path matrix, $H$, with elements $h_{i,j}$ being the minimal number of hops required for transmitting a packet from node $i$ to node $j$, is established running a polynomial-time shortest-path technique such as the Dijkstra algorithm [144] on $T$. Since nodes at hop distance one from the slot node cannot transmit, nodes at hop distance two from the slot node can safely use the time slot, as long as nodes at hop distance three do not transmit and so on. Thus, nodes with even hop distance to the slot node are candidate joining nodes, while nodes with odd hop distance to the slot node are set to be receiving nodes. In cases where nodes with even hop distance to the slot node are neighbors, only one of these nodes can become a joining node. To resolve this conflict, each candidate joining node $j$ is assigned a weight $w_{j,t}$ for slot $t$, and the candidate with the largest weight among the competing nodes becomes the joining node. In order to achieve fairness among candidate joining nodes, $w_{j,t}$ are assigned afresh in each TDMA frame. This is accomplished by random generation of weights in each TDMA frame at all nodes, which use a common random-generator seed and reference time for updating weights.

Defining

$$K_{j,t} = \{k \in \mathcal{N}_{\text{nodes}} | c_{j,k} = 1, h_{t,j} = h_{t,k}\}$$

(7.15)

for $j, t \in \mathcal{N}_{\text{nodes}}, j \neq t$, the proposed schedule for HSR-TDMA can be formalized as
Chapter 7. Robust Spatial Reuse Scheduling in UWANs

follows:

\[ \text{Initialize } I_t = \{t\} \]  
\[ \text{(skeleton TDMA schedule) } \]  
\[ (7.16a) \]

\[ \text{If } ((h_t,j \mod 2 = 0) \land (w_{j,t} > w_{k,t} \forall k \in K_{j,t})) \text{ then } j \in I_t \quad (j \neq t \text{ is a joining node}) \]
\[ (7.16b) \]

Adjusting Flow Constraints in Robust Scheduling

Recall that \( d_i \) in (7.6) represents the minimum number of time slots for transmission which should be assigned to node \( i \) per time frame. Due to the fixed skeleton time-frame, the fairness constraints (7.5) may not always be satisfiable, i.e., the problem may be infeasible. For this reason, we relax the original problem by expanding the time frame from \( L_{\text{skeleton}} \) to \( c \cdot L_{\text{skeleton}} \) slots. Consequently, (7.14) becomes

\[ Aa = c \cdot 1. \]  
\[ (7.17) \]

Note that parameter \( c \) does not affect delay in the network but rather ensures flow control. Consider, for example, a demand in (7.6) of \( g_i = 2 \) transmission slots per-TDMA frame in a fully connected network with \( L_{\text{skeleton}} = 4 \) time slots. Clearly, this demand is infeasible, but can still result a feasible schedule through (7.17) by setting \( c = 2 \).

In the following we present a worst-case approach to determine \( c \). Let \( v_i \) be the pre-defined number of times that node \( i \) is selected as a slot node in one time frame of \( L_{\text{skeleton}} \) time slots. Since a slot node always transmits in its designated time slot and \( x_i \geq v_i \), constraint (7.5) for node \( i \) is surely satisfied within a maximum of

\[ \frac{d_i L_{\text{skeleton}}}{v_i} \leq \frac{\max_i(d_i)L_{\text{skeleton}}}{\min_i(v_i)} \]  
\[ (7.18) \]

time slots. Using the upper bound from the right-hand side of (7.18), we set \( c = \lceil \frac{\max_i(d_i)}{\min_i(v_i)} \rceil \). To ensure the flow constraints from (7.6), we can use \( \sum_{i=1}^{N} g_i \) for \( \max_i(d_i) \) in (7.18).

Formalizing the Robust Scheduling Optimization Problem

Using (7.17) and (7.18), the CUMP R-BSP can be formalized as

\[ \max_a 1^T I_{\text{skeleton}} a \]  
\[ (7.19a) \]

\[ Aa = c \cdot 1, \]  
\[ (7.19b) \]

\[ a \in \mathbb{N}^{K_{\text{skeleton}}}. \]  
\[ (7.19c) \]
Since \( I_{\text{skel}} = [I_{\text{mod}}, \ldots, I_{L_{\text{skel}}}] \), the solution of (7.19) is \( a_{j(i)} = c \) for \( j(i) \) being the index of the column of \( I_{\text{mod}} \) with the largest number of non-zero elements, \( i = 1, \ldots, L_{\text{skel}} \), and \( a_j = 0 \) for the remaining \( j \). The scheduling matrix \( M \) is composed of the selected columns of \( I_{\text{skel}} \) as in (7.13). For the example given in Figure 7.1 we choose columns 6 and 10 of \( I_{\text{skel}} \), since their one-norms are higher than for columns 5 and 9, respectively. Thus, while both T-BSP and R-BSP are NP-hard (since both involve finding independent sets in the network graph), different from the T-BSP, the R-BSP schedule in (7.19) can be obtained without solving an optimization problem.

While Algorithm 4 is performed only once, matrix \( I_{\text{skel}} \) is recalculated through Algorithm 5 every time a topology change is detected, and the R-BSP schedule (7.19) is updated. Hence, the only fixed element of R-BSP is the skeleton matrix \( S_{\text{mod}} \), and even the schedule length may change (but not \( L_{\text{skel}} \)) if parameter \( c \) is updated due to changes in the routing matrix in (7.6). The flow chart in Figure 7.2 summarizes this process, and a software implementation of the algorithm can be downloaded from [96]. To provide an estimate for numerical complexity, we report that, using an Intel Core Duo CPU with a 2.66 GHz processor, the R-BSP schedule for 50 nodes is typically obtained in less than one second.

### A Fair R-BSP Schedule

We note that the R-BSP in (7.19) can be modified to facilitate different objective functions (7.19a). One example would be the case that fairness (7.4) is equally important to channel utilization. Here, we consider equal-resource allocation fairness to approximately equalize the number of resources assigned to users, which is regarded as the fairest resource assignment schedule [145]. Towards this objective we minimize the sample variance of the slot assignments given by

\[
\text{var}(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} (x_i)^2 - \left( \frac{1}{N} \sum_{i=1}^{N} x_i \right)^2 = \frac{1}{N} \mathbf{x}^T \mathbf{x} - \left( \frac{1}{N} \mathbf{1}^T \mathbf{x} \right)^2 .
\]

Since \( \mathbf{x} = I_{\text{skel}} \mathbf{a} \) [see (7.8)] and defining \( \mathbf{V} = (I_{\text{skel}})^T I_{\text{skel}} - \frac{1}{N} (I_{\text{skel}})^T \mathbf{1} \cdot \mathbf{1}^T I_{\text{skel}} \), we have

\[
\text{var}(\mathbf{x}) = \frac{1}{N} \mathbf{a}^T \mathbf{V} \mathbf{a} .
\]

We use the expression from (7.21) to regularize the utilization objective (7.19a) with the regularization weight \( \mu \) and arrive at the fair R-BSP (FR-BSP):

\[
\max_{\mathbf{a}} \left( \frac{1}{c L_{\text{skel}}} \mathbf{1}^T I_{\text{skel}} \mathbf{a} - \mu \mathbf{a}^T \mathbf{V} \mathbf{a} \right) \quad \text{(7.22a)}
\]

s.t. \( \mathbf{A} \mathbf{a} = c \cdot \mathbf{1} \) , \( \mathbf{a} \in \mathbb{N}^{K_{\text{skel}}} \) .

\[
\quad \text{(7.22b)} \quad \text{(7.22c)}
\]

151
Chapter 7. Robust Spatial Reuse Scheduling in UWANs

Algorithm 1

Form list of slot nodes $r(j)$
Rearrange skeleton matrix $S^{\text{mod}}$

Algorithm 2

Construct matrix $I^{\text{ske}}$

Constraints
Construct matrix $A$
Determine $c$

Solve R-BSP
Choose set $j(i)$ of $T^{\text{mod}}_j$ with maximal cardinality
Apply $a_{j(i)} := c$

Form scheduling matrix $M$

Figure 7.2: Flow chart to obtain the R-BSP scheduling matrix $M$.

FR-BSP (7.22) is an integer geometric programming (GP) optimization problem and can be sub-optimally solved in polynomial-time (using e.g., the simplex-like (integer) quadratic programming algorithm [146] or the branch-and-bound algorithm [142]).

We note that the R-BSP in (7.19) is generic with regards to the topology transparent schedule $S$ that is used. However, channel utilization, scheduling delay and short-term robustness performances are affected by the specific choice of the skeleton schedule. Thus, we continue with considerations for selecting the skeleton schedule and the description of two possible skeleton schedules used for numerical results in Section 7.4.

Choosing a Skeleton Schedule

A skeleton schedule is a topology-transparent schedule with fixed frame length $L_{\text{ske}}$. A reasonable choice of such a schedule would be one that achieves high channel utilization while guaranteeing minimal packet collision rate. However, there are other properties that deserve consideration. Since we combine the skeleton schedule with
Chapter 7. Robust Spatial Reuse Scheduling in UWANs

topology-dependent schedule, a high level of flexibility in allocating additional time slots to nodes is also of interest. Moreover, to increase fairness (7.4), the skeleton schedule should allow a fair assignment of slot nodes using Algorithm 4. Another consideration in choosing a skeleton schedule is its frame length $L_{\text{ske}}$, where a short $L_{\text{ske}}$ decreases scheduling delay (7.3).

Following the above considerations, a possible skeleton schedule is the conventional TDMA schedule with frame length $L_{\text{ske}} = N$ in which node $i$ is the slot node for slot $i$. Here, although channel utilization is low, slot node allocation is fair. Another, more sophisticated topology-transparent schedule is the orthogonal topology-transparent (OTT) schedule suggested in [140]. In the OTT schedule the time frame is divided into $v$ subframes, each of which consists of $u$ time slots, and a node is assigned to transmit at least once in each subframe. Here, channel utilization is higher compared to TDMA, but slot node allocation is less fair. The values $u$ and $v$ are assigned such that, for any conflict matrix, $Q$, it is guaranteed that no more than $x$ conflicts occur between any pair of nodes during a time frame of $u \times v$ time slots. We consider the special case of $x = 1$ for which $u = \lceil \frac{1}{2} + \sqrt{\frac{1}{4} + \frac{N}{N}} \rceil$ and $v = \min(u, \Delta + 1)$, where $\Delta$ is the maximum degree of the network graph [140]. In the case that $\Delta$ is unknown, we suggest using a mismatched version of the OTT schedule in which a pre-defined $\Delta_p$ replaces the true degree $\Delta$, which we refer to as mismatched OTT (M-OTT). Since $v = \min(u, \Delta + 1)$ in OTT with $x = 1$ it follows that $v = u$ if $\Delta \geq u - 1$. Hence, in this case M-OTT and OTT become identical schedules if $\Delta_p \geq u - 1$.

To comment on the frequency of the event $\Delta \geq u - 1$, let us consider the popular example of a Bernoulli random graph with non-directed edge probability $p$, for which the node degrees are binomially distributed with parameters $N - 1$ and $p$ (cf. [147]). We can generate such a graph by adding outgoing (directed) edges from each vertex to every other vertex with probability $p' = 1 - \sqrt{1 - p}$. Denoting the outgoing-edge degree of node $i$ by $\delta'_i$, then the C-CDF of $\Delta' = \max_i(\delta'_i)$ is

$$P_r(\Delta' \geq k) = 1 - (P_r(\delta'_i < k))^N = 1 - \left( \sum_{\ell=0}^{k-1} \binom{N-1}{\ell} (p')^\ell (1 - p')^{N-1-\ell} \right)^N. \quad (7.23)$$

Since the edge degree is lower bounded by the outgoing-edge degree, evaluating (7.23) for $k = u - 1$ gives a lower bound on the probability that $\Delta \geq u - 1$. Figure 7.3 shows the lower bound (7.23) for $k = u - 1$ as well as the empirical probability $P_r(\Delta \geq k)$ as a function of $p$ and several values of $N$. We observe that already with $p = 0.5$ the $P_r(\Delta' \geq u - 1)$ and thus $P_r(\Delta \geq u - 1)$ is close to one, and hence a choice of $\Delta_p = u - 1$ for M-OTT would render it identical to OTT with high probability. We report that the same conclusion was drawn for a more realistic (and limited) set of topologies obtained from both simulations and sea trial recordings, as described in the following.

153
Figure 7.3: Probability (7.23) for $k = u - 1$ and empirical probability of $\Delta \geq u - 1$ when generating a Bernoulli random graph with $N$ nodes and edge probability $p$.

7.3 Obtaining the Topology Matrix

The optimal T-BSP and R-BSP schedules as well as the suboptimal HSR-TDMA one require knowledge of the MIS matrix $I$, obtained from conflict matrix $Q$. In this section we describe a mechanism to construct $Q$ for different MPR scenarios, and share it across the network.

7.3.1 Obtaining the Topology Matrix

We start with tracking and sharing the time-varying network topology. Network topology is represented by an $N \times N$ matrix $T$ with binary elements $t_{i,j}$, representing existence of a direct communication link between nodes $i$ and $j$. Matrix $T$ is shared across the network by nodes piggy-backing their connectivity lists (CLs) of one-hop neighbor nodes on broadcast packets (for example periodic navigation packets in the Deep-Link system [133]). Initially, we assume a fully connected network, i.e., $T$ is the all-one matrix. This way, initial transmissions are interference-free and nodes can start updating their topology matrix. Note that this initial schedule is updated through executing Algorithm 5 and solving (7.19) every time a topology change is detected. The mechanism to update the topology information is described in the following.

Flickering

Considering the time-varying characteristics of the channel, we wish to avoid the \textit{flickering} problem in which topology changes too fast to distribute across the network.
Thus, links need to be stabilized. We account for the flickering problem by introducing counters $b_1$ and $b_2$ used for adding and removing communications links, respectively. In R-BSP, since node $j$ is a slot node at least $v_j$ times in a time frame, and $v_j$ is pre-defined (see Section 7.2.3), node $i$ would remove $j$ from its connectively list if it did not receive a packet from $j$ within the last $T_{\text{remove},j} = \lceil \frac{b_2}{v_j} \rceil$ time frames. Similarly, in T-BSP, node $i$ is guaranteed to transmit $d_i$ packets per time frame, and thus, in T-BSP, $T_{\text{remove},j} = \lceil \frac{b_2}{d_j} \rceil$. On the other hand, $i$ would add $j$ to its connectivity list only if it receives $b_1$ successive packets from $j$.

Applying the above sensitivity mechanism introduces a delay in the update of CLs. The values of $b_1$ and $b_2$ can be regarded as the convergence parameters of the scheduling algorithm, since the process of recovery from topology changes does take on the order of $b_1$ and $b_2$ time frames for adding and removing links, respectively. However, when topology changes faster than the convergence time of the algorithm or when topology updates bearing packets are lost, CLs may be outdated, and scheduling conflicts may occur. Furthermore, multihop routes would break resulting in packet losses. To account for these fast topology variations, we extend the sensitivity mechanism beyond neighbor nodes such that if a node $i$ did not receive a packet (via a single or multiple hops) from node $j$ in $b_3$ time frames, it refrains from transmitting in slot $j$. The value $b_3$ trades off robustness to fast topology variations (increases with smaller $b_3$) and availability. A pragmatic choice is $b_3 = \Upsilon \cdot b_2$, where $h$ is the maximal number of hops in the network.

Denoting by $P_{e,\text{pac}}(i,j)$ the packet error rate (PER) for the link between $i$ and $j$, the probability for node $i$ to receive $k$ successive packets transmitted from node $j$ is $p_{\text{suc}}(i,j) = (1 - P_{e,\text{pac}}(i,j))^k$. Hence we can lower bound the miss-detection probability, $p_{\text{mis}}$, by choosing $k$ such that $p_{\text{suc}}(i,j) < p_{\text{mis}}$. Since $P_{e,\text{pac}}(i,j)$ is difficult to estimate before a link between nodes $i$ and $j$ has been established, a maximal expected PER, $p_{e,\text{max}}$, based on transmission range of the system can be used. Thus, as a pragmatic solution we suggest the selection of $b_1$ such that

$$ (1 - p_{e,\text{max}})^{b_1} > 1 - p_{\text{mis}}, \quad (7.24) $$

where $p_{\text{mis}}$ is set according to the tolerated miss-detection rate in worst-case links. On the other hand, node $j$ is removed from the CL of node $i$ if node $i$ did not receive a packet transmitted from $j$ in the last $b_2$ time frames. Since the probability of receiving at least one packet in $k$ frames is given by $1 - (P_{e,\text{pac}}(i,j))^k$, we adjust $b_2$ according to

$$ (p_{e,\text{max}})^{b_2} < p_{\text{drop}}, \quad (7.25) $$

where $p_{\text{drop}}$ represents an acceptable dropping probability for the expected worst-case link. Figure 7.4 shows $b_1$ and $b_2$ according to (7.24) and (7.25), respectively. We note that in our system only high values of $p_{\text{mis}}$ and $p_{\text{drop}}$ are considered since (i) these apply for worst-case links and (ii) the expected value of $P_{e,\text{pac}}(i,j)$ is high for transmission ranges above 1 Km, which are of our interest.
Symmetry

While we assume $T$ is a symmetric matrix, in the underwater channel links between nodes located at different depths might not be reciprocal due to different noise levels. Therefore, we apply a conservative strategy and locally replace all asymmetric components $t_{i,j} \neq t_{j,i}$ of $T$ by $t_{i,j} = t_{j,i} = 1$ to force symmetry. Note that the symmetrized connectivity matrix is generated locally, whereas the actual CLs giving rise to a (possibly) asymmetric $T$ are shared throughout the network.

7.3.2 Constructing the Conflict Graph

If the system has either ideal or no MPR capabilities, the conflict matrix $Q$ is directly obtained from matrix $T$. Specifically, considering primary conflicts, for the former we set $Q = T$, while for the latter $Q = T \cdot T$. However, as mentioned in Section 7.1.1, we can also represent conflict graphs for systems with limited MPR capabilities. In particular, in a system where MPR is limited to a pre-defined number of packets, $N_{\text{limit}}$, we require $\sum_m q_{m,n} \leq N_{\text{limit}}$. A conflict graph for such system is constructed by rejecting independent sets in $I$ which include more than $N_{\text{limit}}$ nodes connected to a common node. Alternatively, a conflict graph for a more realistic system, where MPR ability is determined by the relative distances of transmitters to a common receiver, is constructed by setting $q_{i,j} = 1$ if nodes $i$ and $j$ are the interfered and jammer node in a near-far scenario [148] as described in the following.

In a near-far scenario, transmissions from nodes close to a receiver interfere transmissions of farther nodes. One possible approach to reduce such interference is to include a desired SINR level as constraints in the scheduling optimization problem.
Chapter 7. Robust Spatial Reuse Scheduling in UWANs

(e.g. [149] and [74]). However, this requires distance evaluation to other nodes and an accurate propagation model for the estimation of the SINR. In case of a near-far, we define the receiver node as the center node, the closer nodes (which cause the problem) as the interfering nodes, and the more distant nodes whose transmissions are blocked as the jammed nodes. We identify the interfered and jammer nodes based on the fact that, as a result of near-far, the interfered node appears to be connected to the center node, but not vice-versa. That is, T becomes asymmetric. Unlike temporal instabilities of T due to flickering or short-term interference, asymmetry due to the near-far effect is stable. Hence, the identities of the center, interfering, and jammed nodes can be detected from observing long-term asymmetric components of T. In case a node cannot determine which of a set of nodes is the jammer node, it would assume all nodes in this set are jammers, and would avoid joining their transmissions. Furthermore, in case of multiple near-far scenarios, since symmetry is checked per center node, separate a-symmetric components cannot be identified.

7.4 Simulation Results

In this section we report simulation results to illustrate the performance of the proposed R-BSP scheduling with regards to 1) per-node throughput, 2) scheduling delay, and 3) fairness. In particular, we present results for the CUMP R-BSP schedule (7.19) and the FR-BSP schedule with \( \mu = 1 \) (7.22) using two skeleton schedules: 1) the M-OTT and 2) the TDMA, and compare them with results for the T-BSP, HSR-TDMA (7.16), OTT from [140] and the conventional TDMA schedule. Furthermore, to show the effect of the skeleton schedule, we also consider the schedule resulting from the CUMP in (7.12) when applying the frame length \( L = c \cdot L_{\text{skele}} \), i.e., the R-BSP frame length, which we refer to as the CUMP schedule in the following.

In our simulations, we implemented the mechanism described in Section 7.3, which allows nodes to track the network time-varying topology and conflict matrices. By allowing nodes to move and measuring performance over time \( T \), we show how each algorithm reacts to topology changes. Furthermore, while the above algorithms may have different TDMA frame-lengths, we always consider the same traffic generation. In particular, we require a minimal number of original packets per frame of \( g_i = 1 \) \( \forall i \) [see (7.6)] for fairness, and let nodes transmit up to \( N \) original packets per-TDMA frame. We apply the minimal hop-distance routing mechanism for all experiments. Given the topology matrix, this is obtained using the Dijkstra algorithm (cf. [144]). Since we consider broadcast packets, i.e., each packet is directed to all other network nodes, and since we consider half-duplex communication, primary conflicts are not allowed in the network conflict matrix, \( Q \). In our simulations we consider four different MPR scenarios: i) ideal MPR, ii) MPR of up to 2 overlapping received packets (Limit MPR \( (N_{\text{limit}} = 2) \)), iii) MPR of up to 3 overlapping received packets (Limit MPR \( (N_{\text{limit}} = 3) \)), and iv) no MPR capabilities (No MPR). For the above
four scenarios, we neglect interference from nodes located more than two hops away. This is considered since unlike for terrestrial radio-frequency communication, where concurrent transmission of nodes three or more hops away may notably decrease the SINR at the receiving node [149], such degradation is usually negligible in UWAC due to the significant effect of the channel absorption loss. We used two approaches to obtain network topologies in our simulations. First, a model-based topology was generated based on an attenuation model. The model-based simulations are used to simulate movements of AUVs and divers in a near-shore or harbor environment, as considered in the Deep-Link application. The main focus of these simulations is to compare short-term robustness performance. Since model-based topologies might be too simplified, we also use a time-varying topology recorded in a sea trial, where we tested our algorithm also for the realistic case of MPR limited by the relative distance between sources and destination (i.e., the near-far problem). In the following we present results of both simulations.

### 7.4.1 Model-Based Topology

For model-based topologies, we generated a Monte-Carlo set of 1000 topologies with \( N = 8 \) nodes, and measured the network performance for each topology in a fixed interval of \( T = 1000 \) time slots with duration of 5 seconds, considering transmitted packets of duration 1.49 seconds, maximal transmission range of 5 km, sound speed of 1500 m/sec, and time-synchronization uncertainties up to 10 msec [14]. We placed nodes according to a uniform distribution in a square area of size 5 km \( \times \) 5 km. This area included four horizontal obstacles and one vertical obstacle at uniformly distributed positions within the square area, with lengths being uniformly distributed in \([100, 200]\) m. In our setting, a link exists if no obstacle obstructs the line-of-sight path and the expected error rate for packets of 100 bits and BPSK communication is below \(10^{-4}\). The effect of topology variations on system performance was investigated for each topology by allowing nodes to move during the network operation (the effect of link flickering is considered in the sea-trial-based topologies, described in the next section). At the start of the simulation, each node was assigned with a motion vector with uniformly distributed speed and direction of \([-5, 5]\) knots and \([0, 360]\) degrees, respectively, and sustained the same course. If a node reaches an obstacle, a new randomized direction is determined, away from the obstacle. We note that the above parameters reflect the expected packet length and node mobility in the Deep-Link application, where longitude and latitude information are shared in the network and AUVs or divers are assumed moving in a fixed course and speed to explore a near-shore or harbor environment, which often leads to sparse topologies (for further details see [133]).

While there exist several tools to simulate an underwater channel, e.g., the Bellhop simulator [2], their run-time is rather slow. Since we require a large number of channel realizations, we used the popular transmission loss model (which implies a carrier
Chapter 7. Robust Spatial Reuse Scheduling in UWANs

Figure 7.5: C-CDF of $\rho_{\text{through}}$ from (7.2) for matched and mismatched OTT and static model-based topology.

frequency of about 30 kHz) [5, Ch. 5]

$$TL(r) = 20 \log_{10}\left(\frac{r}{1 \text{ m}}\right) + \frac{3r}{1 \text{ km}} \text{ dB},$$  \hspace{1cm} (7.26)

where $r$ is the transmission distance. We consider a source level of 155 dB/µPa@1m, a noise level of 50 dB/µPa/Hz, DSSS signals with $L_c = 31$ chips, a packet size of 100 bits, a transmission rate of 100 bps, and a required packet error rate of $10^{-4}$. Based on these parameters, the maximal hop distance, $d_{\text{hop}}$, is approximately 2000 m. Furthermore, assuming a normalized cross correlation of $\frac{1}{L_c}$ for different pseudo-random sequences, we get negligible interference (i.e., more than 10 dB below the noise level) from transmissions of nodes located more than 3200 m from the receiver, i.e., $1.6 \cdot d_{\text{hop}}$. This justifies our assumption above for neglecting interference from nodes located more than two hops away.

Following the discussion in Section 7.2.3 we choose the value $\Delta_p = u - 1$ for the pre-defined graph degree of the M-OTT. The suitability of this choice is verified in Figure 7.5 showing the empirical C-CDF of the measured per-node throughput (7.2) for the OTT and M-OTT schedules for static model-based topologies (i.e., nodes did not move after setting up the network topology). The results show that M-OTT with the pre-determined value $\Delta_p = 3$ achieves practically the same performance as OTT. Since $u = 4$ for networks with $N = 8$ and $x = 1$, $\Delta_p = 3$ for M-OTT in the following.

We start with comparing the measured per-node throughput (7.2). For the case of static nodes, Figure 7.6 shows the empirical C-CDF of throughput of the R-BSP schedule for the different MPR scenarios. Additionally, we show throughput results for the TDMA schedule. We observe that the performance of R-BSP with ideal
MPR is almost identical to that of a system with limited MPR and $N_{\text{limit}} = 3$, and that only a small performance degradation occurs when $N_{\text{limit}} = 2$. Furthermore, we observe that while performance for a system without MPR capabilities decreases notably compared to that of a system with MPR, significant spatial reuse gain over conventional TDMA is still achieved. However, the effect of simultaneous transmissions on SINR is expected to grow with $N$. Thus, we conclude that, for small $N$, the proposed spatial-reuse schedule is beneficial regardless of the MPR capabilities. For clarity and simplicity, in the following we focus on the case of ideal MPR. In Figure 7.7a we compare per-node throughput for the different considered scheduling algorithms when nodes are static. Considering, for example, a per-node throughput of 0.1 as a desired quality of service (QoS), we observe that using OTT and TDMA, the desired performance is hardly reached for all topology configurations. However, by applying spatial reuse, the desired QoS is achieved in more than 50% of all topology configurations using the R-BSP approach. Furthermore, comparing performance of R-BSP to that of T-BSP we observe clear throughput advantages for R-BSP. This result is expected since, while R-BSP uses a fixed frame length, T-BSP minimizes frame length, which tradeoffs channel utilization with scheduling delay. The CUMP schedule, which applies the same frame length as R-BSP, achieves a notably higher throughput than R-BSP. This, however, comes at a considerable cost in scheduling delay and fairness as will be discussed below. We observe that the R-BSP schedule with TDMA skeleton achieves higher throughput than HSR-TDMA, which also uses

---

23 We note that since we regard throughput as a measure of successfully received packets, which may involve routing, it is not identical to channel utilization and thus throughput of TDMA is not limited to $\frac{1}{N}$. 

---
Figure 7.7: C-CDF of $\rho_{\text{through}}$ from (7.2) for model-based topologies. (a) static (no outdated topology). (b) dynamic (slowly propagating topology)

Chapter 7. Robust Spatial Reuse Scheduling in UWANs

...
for static model-based topologies. Since scheduling delay is strongly connected to the number of time slots a node is assigned to, a significantly lower delay is obtained for the T-BSP and the R-BSP schedules using M-OTT as a skeleton schedule compared to the conventional TDMA and the OTT schedules. Comparing delay performance of R-BSP with M-OTT to that of CUMP we observe that although both schedules have the same frame length, due to the skeleton schedule used in R-BSP its delay is significantly lower. However, since the frame length is directly minimized in the T-BSP schedule (see (7.10)), it achieves a slightly lower scheduling delay in almost all topologies than R-BSP with M-OTT skeleton schedule. We observe that delay performance of FR-BSP and CUMP R-BSP with M-OTT skeleton are almost the same. While delay performance of R-BSP with TDMA skeleton and HSR-TDMA are almost the same, since in M-OTT nodes are assigned to transmit in more time slots than in TDMA, better delay performance is obtained comparing R-BSP with M-OTT to R-BSP with TDMA. We do not show results for scheduling delay for a dynamic topology, since we only consider successfully received packets for scheduling delay (7.3) and no retransmission mechanism in case of packet collisions. Thus, outdated topology information leading to packet collision has little influence on scheduling delay.

### 7.4.2 Sea-Trial-Based Topology

While the model-based simulation, introduced above, include node mobility and thus topology uncertainties, the considered movements may be too artificial. For this reason, in the following we also consider topologies obtained from communication links recorded in a sea trial conducted in May 2009 in the Haifa harbor. The sea
Chapter 7. Robust Spatial Reuse Scheduling in UWANs

Figure 7.9: Structure of Sea Trial: (a) satellite picture of the sea trial location (picture taken from Google maps on September 29, 2009). (b) recorded network topologies.

The sea trial included four vessels, which moved between the harbor docks in a slow speed of about 1 m/sec. The different locations of the nodes during the sea trial are marked in Figure 7.9a, where, for example, node 1 moved between points 1A, 1B, 1C, and 1D, at which it stayed for some time. The topology structures of the network as recorded during the sea trial are shown in Figure 7.9b, where first the nodes communicated in a fully connected network (i.e., Topology 1), then formed Topology 2, and so on until the nodes formed the chain Topology 6 where node 1 is not connected. Note that the difference between Topologies 4 and 5 is the distance between nodes 1 and 2.

As in the Deep-Link application, during the sea trial, each node broadcasted its time-varying location coordinates to the rest of the network nodes every single minute, which we adopt as the duration of the time slot. This motion of nodes and communication type represent the practical scenario of divers moving in a near-shore environment with obstacles and broadcasting their location coordinates to all other nodes for safety reasons and command and control. The four nodes transmitted at a power such that given the small dimensions of the harbor (1500 m at its longest axis) connectivity is mainly determined by the structure of the harbor and existing obstacles. Based on received packets and using a flickering-mitigation mechanism to stabilize network topology, each node identified its list of one-hop neighbor nodes and piggy-backed it on its broadcast packets to create a shared network topology. Given the measured network topology, conflict matrix $Q$ was obtained considering a system with $N_{\text{limit}} = 3$, while satisfying primary conflicts and accounting for possible near-far problems. For the latter, referring to Figure 7.9a, near-far problem occurred when nodes 1, 2, and 4 were located at 1A, 2A, and 4A, respectively. Using the mechanism described in Section 7.3, our algorithm resolved this problem and limited scheduling such that nodes 1 and 4 cannot transmit simultaneously.
Chapter 7. Robust Spatial Reuse Scheduling in UWANs

Figure 7.10: Results of $\rho_{\text{through}}$ from (7.2) for topologies recorded in the sea trial and $T = 10$ time-slots. Each vertical line represents the time a topology change starts affecting results.

In Figure 7.10, we show the throughput (7.2) during the sea trial as function of time, where each throughput measurement is averaged over a sliding window of $T = 10$ time slots. For clarity, we present result only of TDMA, T-BSP, and R-BSP with OTT skeleton. The results reflect exactly what has been recorded during the sea-trial, including the sharing of topology information, and thus include the effect of topology mismatch. The straight lines in Figure 7.10 mark the time a topology change starts affecting the throughput results, and the labels for topologies match the topologies shown in Figure 7.9b. The effect of a topology change is evident by the network recovery time and the absolute throughput decrease. The former is measured by the difference from the time topology changes until it is updated and throughput converges, and the latter by the difference between the throughput once the change occurs and after topology updates. From Figure 7.10, we observe that compared to performance of T-BSP, effect of topology changes on throughput of R-BSP is much lower. Consider, for example, Topology 4 recorded from time slot number 150 until 210. Here, for T-BSP, throughput drops to 0.14 until it converges to 0.26 (i.e., a difference of 0.12) after roughly 40 time slots. However, for R-BSP, the throughput difference is only 0.03 and network recovers after less than 20 time slots.

In Figure 7.11, we show scheduling delay performance for the sea-trial-based topology. As for the simulation-based topologies, we again distinguish between static and dynamic topologies. The latter is similar to the time-varying topologies considered for Figure 7.10, and the former is emulated by assuming that all nodes share the same (and correct) topology information. The value of this static-topology case is to show results for measured and thus realistic topologies which complement those
for simulation-based topologies in the previous section. We observe that when topology is dynamic, scheduling delay of T-BSP, CUMP and HSR-TDMA considerably increases compared to the case of static topology. However, delay performances of the topology-transparent algorithms and the three variants of the R-BSP algorithm hardly change, which demonstrates their short-term robustness to topology variations.

In summary, considering the results for the different performance criteria for both model-based and sea-trial-based topologies, we conclude the proposed R-BSP provides a favorable tradeoff of throughput and scheduling delay, together with flow control and a high short-term robustness to outdated topology information.

Finally, a word on communication overhead is in order. While topology-transparent and random-access scheduling algorithms (e.g., handshake and Aloha) are fully distributed, both R-BSP and T-BSP are centralized solutions as they require topology information of the entire network. However, the actual calculation of the schedule is performed locally. Furthermore, as these algorithms support transmission of broadcast packets, and topology information is piggy-backed on broadcast packets, the overhead for each packet is limited to $N \times (N - 1)$ bits. This overhead can be further reduced by transmitting only updates of topology information.

### 7.5 Conclusions

In this chapter, we considered the problem of scheduling in high-traffic UWANs supporting transmission of broadcast packets. We formalized the problem of resource assignments to nodes to maximize channel utilization under certain fairness/flow con-
Chapter 7. Robust Spatial Reuse Scheduling in UWANs

trol. Considering the challenges of resource assignment for time-varying topologies, we suggested combining topology-transparent with topology-dependent scheduling to improve short-term robustness to inaccurate or outdated topology information. We presented two robust scheduling algorithms to maximize channel utilization and to improve fairness. By means of simulation results, we demonstrated that our approach outperforms existing topology-transparent algorithms in terms of throughput and scheduling delay by reusing topology information. Furthermore, we showed that compared with an optimized topology-dependent schedule, short-term robustness to topology variations is significantly improved, while scheduling delay is only slightly increased. Further work will include a utilization of the long propagation delay in the channel by scheduling transmissions within time slots to allow simultaneous transmissions from neighbor nodes, while avoiding scheduling constraints.
Chapter 8

Adaptive Error-Correction Coding Scheme for UWANs

The spatial-reuse TDMA scheduling protocol offered in the previous chapter uses topology information in the form of connectivity and conflict matrices. While the latter requires location information to evaluate whether a node is within the interference range, only a rough location estimation is required and the full localization capability developed in Part I is not used. To complement this, in this chapter we utilize propagation delay information to improve reliability in TDMA UWANs through adaptive channel coding. While the existing literature of adaptive transmissions in UWANs presented in Section 1.2.2 demonstrated significant improvement in network throughput and latency, we argue that there is room for further improvement by obtaining channel state information (CSI) in the form of distance to nearby nodes. Since in slotted scheduling, the guard interval is usually dimensioned according to the modem’s maximal detection range (which for UWANs corresponds to a propagation delay on the order of a few seconds for a typical interference range of a few kilometers [5]), each (re)transmission includes a sizeable overhead. Thus, considering that actual propagation delay for specific communication links is often notably shorter than the maximal expected delay, we suggest improving link reliability by utilizing the often over-sized guard interval in a time slot. In particular, assuming that the propagation delays to nodes within the interference range of a transmitting node are known (e.g., from frequent packet exchange, such as from navigation packets in Deep-Link), our scheme opportunistically includes extra parity symbols in the data packet if the guard interval is longer than needed for interference-free transmission. Since extending the code length trades off reliability and energy consumption for transmission, we optimize the number of parity symbols used, considering both single packet transmission and packet retransmission with IR-HARQ. We note that compared to conventional adaptive coding, where CSI usually is given in the form of a link quality parameter like SNR, one novel aspect of our approach lies in the direct use of ranging information as part of the CSI. Such adaptation of coding rate based on delay has not been considered before.

We present two possible implementations for our adaptive coding approach, one based on a bank of codes and another one based on rateless codes. For both schemes, we include the case of ranging information being available only at the transmitter side, thus obtaining range-based adaptive coding with no communication overhead. We present simulation results for typical underwater acoustic environments as well.
as experimental results from a sea trial. The results demonstrate that, when ranging information is available (e.g., the Deep-Link application), our protocol provides significant gains compared to the performance of fixed coding schemes in both reliability and transmission energy consumption.

The remainder of this chapter is organized as follows. The details of the system model are introduced in Section 8.1. In Section 8.2, we describe our adaptive coding approach for both single packet transmission and IR-HARQ. Two possible implementations schemes of the proposed adaptive coding method are introduced in Section 8.3. Simulation and experimental results are presented and discussed in Section 8.4, and conclusions are drawn in Section 8.5.

8.1 System Model

We consider slotted UWANs where node $i$ is assigned with messages of $K$ information symbols to transmit to its designated receiver $j$. The considered scenario accommodates unicast communication, and besides slotted transmissions, we set no limitation on the scheduling protocol. We define the link interference range as the distance up to which transmissions significantly affect the signal-to-interference-plus-noise ratio (SINR) at an unintended receiver. Due to the large attenuation in the underwater acoustic channel (cf. [5]) we assume that the per-link interference range can be bounded by the detection range of synchronization signals, whose energy is usually much larger than that of information bearing signals. We assume periodic packet exchange between nodes, such that by estimating the time-of-flight [150] of received synchronization signals, each node is aware of the distance to nodes located within its interference range.

Let $d_{\text{max}}$ be the maximal interference range of the system. Furthermore, for time slot $t$, let $d_{t,i}$ be the maximum distance of node $i$ to nodes within its interference range which are scheduled to receive in time-slots $t$ and $t+1$. Clearly,

$$d_{t,i} \leq d_{\text{max}},$$

and thus often the guard interval could be reduced while still ensuring that packets are received interference-free. Using knowledge of $d_{t,i}$, our goal is to optimize time slot utilization of node $i$. The optimization aims to strike a balance between reliability of and energy consumption for transmission.

8.2 The Adaptive Coding Scheme

Let us first consider the transmission of a single packet, i.e., no ARQ is applied. If a fixed-rate error-correction code is used, $K$ information symbols are encoded into $N_{\text{min}}$ coded symbols. $N_{\text{min}}$ is determined assuming $d_{t,i} = d_{\text{max}}$ and thus the full guard
interval is needed for interference-free scheduling. However, assuming that node \( i \) has an accurate estimate of the range difference

\[
\Delta t,i = d_{\text{max}} - d_{t,i},
\]

(8.2)

we can shorten the guard interval and transmit up to

\[
N_{t,i}^{\text{max}} = \lfloor N_{\text{min}} + \frac{\Delta t,i}{T_s \cdot c} \rfloor
\]

(8.3)

symbols without causing interference to other unintended receivers. In (8.3), \( T_s \) and \( c \) are the symbol period and propagation speed, respectively. Hence, denoting by \( N_{t,i}^{\text{actual}} \) the number of actually transmitted symbols transmitted at time \( t \) from source \( i \), then \( N_{\text{min}} \leq N_{t,i}^{\text{actual}} \leq N_{t,i}^{\text{max}} \).

Next, we demonstrate the gain that can be achieved by adaptive coding in terms of the packet-error-rate (PER) and the energy consumption for transmission. This not only highlights the possible improvements by better utilizing the time slot, but also sets the stage for the following optimization of \( N_{t,i}^{\text{actual}} \).

### 8.2.1 Gain of Adaptive Coding

We use the example of an \((K,N)\) Reed-Solomon (RS) code [151] to analyze the performance improvements due to adaptive coding.

**PER Gain**

Let \( N' \) be the number of transmitted symbols after puncturing \( N - N' \) symbols of the original RS code. Denoting by \( p^{\text{RS}} \) the RS symbol error probability before decoding,
the packet error probability (PER) after decoding is given by

\[ p_{\text{packet}}^{\text{RS}}(N') = \sum_{k=\lceil (N' - K)/2 \rceil + 1}^{N'} \left( \begin{array}{c} N' \\ k \end{array} \right) (p_{\text{RS}})^k (1 - p_{\text{RS}})^{N' - k}. \] (8.4)

Let us consider the PER when the receiver obtains \( N' = N_{\text{actual}}^{t,i} \) demodulated symbols, and compare it to the case of coding with \( N_{\text{min}}^{t,i} \). By (8.4), the PER of these schemes is \( p_{\text{packet}}^{\text{RS}}(N_{\text{actual}}^{t,i}) \) and \( p_{\text{packet}}^{\text{RS}}(N_{\text{min}}^{t,i}) \), respectively. Thus, in terms of success rate, the gain of the adaptive punctured RS coding scheme over the fixed code is

\[ g_{\text{coding}} = \frac{1 - p_{\text{packet}}^{\text{RS}}(N_{\text{actual}}^{t,i})}{1 - p_{\text{packet}}^{\text{RS}}(N_{\text{min}}^{t,i})}. \] (8.5)

Note that metric \( g_{\text{coding}} \) from (8.5) is meaningful only in the case of poor performance of the non-adaptive coding scheme, which requires an ARQ protocol and is the case considered.

In Figure 8.1, for \( T_s = 0.01 \) sec, \( c = 1500 \) m/sec, we show \( g_{\text{coding}} \) from (8.5) for \( N_{\text{actual}}^{t,i} = N_{\text{max}}^{t,i}, K = 54, \) and \( N_{\text{min}}^{t,i} = 63 \), as a function of \( \Delta_{t,i} \) from (8.2), and for several values of \( p_{\text{RS}}^{\text{RS}} \). We observe that \( g_{\text{coding}} \) is significant and fast increasing with \( \Delta_{t,i} \) and \( p_{\text{RS}} \). We note that for large \( \Delta_{t,i} \) the PER of the adaptive coding scheme becomes extremely small, and \( g_{\text{coding}} \) converges to \( 1/(1 - p_{\text{packet}}^{\text{RS}}(N_{\text{min}}^{t,i})) \) as observed in Figure 8.1.

**Energy Consumption for Transmission**

Since instead of \( N_{\text{min}}^{t,i} \) symbols, we opportunistically transmit \( N_{\text{actual}}^{t,i} \) symbols per packet, the potential gain of the adaptive coding scheme over the fixed code in terms of PER is achieved at the cost of higher energy consumption for per-packet transmission. This may seem to be a challenge for UWAC networks, where often energy resources at (battery-operated) network nodes are limited. However, considering the low reliability in UWAN links, and accordingly the energy spent for successfully received data packets, the improved error correction capability of the adaptive coding scheme may ultimately reduce energy consumption for transmission, as we discuss next.

Let \( p_{\text{packet}}(N) \) be the error rate of a packet transmitted using an \( (K, N) \) error-correction code. Then, a successful transmission requires on average

\[ M(N) = \frac{1}{1 - p_{\text{packet}}(N)}. \] (8.6)

packet transmissions. Furthermore, denoting by \( S \) and \( T_h \) the transmit power and duration of the packet header and pre-amble sequence, respectively, the average energy
Figure 8.2: Gain $g_{\text{energy}}$ from (8.7) as a function of $T_h$ and $\Delta_{t,i}$ from (8.2). Non-erasure channel.

consumption for one successful packet transmission is $M(N) \cdot S(NT_s + T_h)$. To quantify the advantage of our adaptive coding approach in terms of transmission-energy consumption we consider the ratio

$$g_{\text{energy}} = \frac{M(N_{\max}^t) \cdot (N_{\min}^t T_s + T_h)}{M(N_{\text{actual}}^t) \cdot (N_{\text{actual}}^t T_s + T_h)}.$$  

(8.7)

We note that $g_{\text{energy}}$ in (8.7) increases with $T_h$, and is influenced by $N_{\text{actual}}^t$ through the energy consumption term $PN_{\text{actual}}^t T_s$ and the average number of packet transmissions $M(N_{\text{actual}}^t)$, where the latter depends through (8.6) on the packet error rate (PER) $p_{\text{packet}}(N_{t,i}^\text{actual})$. Hence, $M(N_{\text{actual}}^t)$ decreases with $N_{t,i}^\text{actual}$, while the energy consumption term increases for larger $N_{t,i}^\text{actual}$.

To shed some light on the value of $g_{\text{energy}}$ from (8.7), we consider the example of the RS code used already in Section 8.2.1. In Figure 8.2, we show $g_{\text{energy}}$ from (8.7) as a function of $\Delta_{t,i}$ (8.2) and $T_h$, and the same set of parameters considered in Section 8.2.1. In particular, $N_{\text{actual}}^t = N_{\text{max}}^t$. We note that while $g_{\text{energy}}$ increases with $T_h$, due to the low transmission rate this increase is not significant. As expected, we observe that the curves have a distinct maximum, reflecting the trade-off between larger transmission energy and lower PER for increasing $N_{t,i}^\text{max}$. We note that $g_{\text{energy}}$ increases with larger symbol-error rate $p_{\text{RS}}^t$, as it emphasizes the error-rate gain of adaptive over fixed-rate coding. Furthermore, we observe that the ratio $g_{\text{energy}}$ is larger than one in a wide range of distances $\Delta_{t,i}$ for $p_{\text{RS}}^t \geq 0.08$. Hence, adaptive coding consistently provides a gain in energy consumption in such unreliable communication channels, which are typical for UWANs.

Considering the convergence of $g_{\text{coding}}$ from (8.5) to a constant gain value as shown
in Figure 8.1, and the non-monotonic behavior of $g_{\text{energy}}$ from (8.7) as can be seen from Figure 8.2, we note that the choice of $N_{t,i}^{\text{actual}} = N_{t,i}^{\text{max}}$ can be improved upon, especially for large values of $\Delta_{t,i}$. In particular, to limit energy consumption, we optimize $N_{t,i}^{\text{actual}}$, as described in the following.

### 8.2.2 Optimization of $N_{t,i}^{\text{actual}}$

The channel between transmitter $i$ and receiver $j$ at time $t$ is characterized by its capacity $C_{t,i,j}$, whose unit is bit per coded symbol. Assuming that channel conditions do not change much within a packet transmission, the receiver can likely successfully decode after $N_{t,i}^{\text{actual}}$ transmitted symbols if

$$C_{t,i,j}N_{t,i}^{\text{actual}} \geq (1 + \delta)KL,$$

where $L$ is the byte size of coded symbols and $\delta$ accounts for the gap to capacity using practical codes. Hence, by estimating $C_{t,i,j}$ transmitter $i$ can determine the number of symbols required for successful decoding. Combining (8.8) with adaptation based on interference range, for a single packet transmission we have

$$N_{t,i}^{\text{actual}} = \min \left( \frac{(1 + \delta)KL}{C_{t,i,j}}, N_{t,i}^{\text{max}} \right).$$

While for the transmission of the first packet, $N_{t,i}^{\text{actual}} \geq N_{i}^{\text{min}}$ (see Section 8.2), for later packets $N_{t,i}^{\text{actual}}$ according to (8.9) could be below $N_{i}^{\text{min}}$, which would lead to an overall energy savings compared to the fixed-rate coding scheme.

As an example of obtaining $N_{t,i}^{\text{actual}}$, we consider the $M$-ary symmetric channel (MSC) and the $M$-ary erasure channel (MEC). The former model is a good fit for the RS code discussed in Section 8.2.1, and the latter is used for rateless Fountain codes, discussed further below. For the case of RS coding and the MSC, we have $M = 2^L$ symbols and the capacity is

$$C_{t,i,j} = L - H(p_{t,i,j}^{\text{RS}}) - p_{t,i,j}^{\text{RS}} \log_2(M - 1),$$

where $H(\cdot)$ is the binary entropy function. For the MEC model with symbol erasure probability $p_{t,i,j}^{e}$, we have

$$C_{t,i,j} = L \left( 1 - p_{t,i,j}^{e} \right).$$

For (8.10) and (8.11), estimates $\hat{p}_{t,i,j}^{\text{RS}}$ and $\hat{p}_{t,i,j}^{e}$ of the symbol error and erasure rates, respectively, are required. The process of obtaining these estimates is described next for both single and multiple packet transmissions.

### Single Packet Transmission

If a feedback channel exists (as assumed in [89]), the receiver can measure the channel conditions (e.g., from a received request-to-send packet) and report $C_{t,i,j}$ to the
transmitter, which then directly calculates $N_{t,i}^{\text{actual}}$ from (8.9). Otherwise, since both $p_{t,i,j}^{\text{RS}}$ from (8.10) and $p_{t,i,j}^{\text{e}}$ from (8.11) can be represented as a function of the SNR, $\text{snr}_{t,i,j}$, we use the distance information available at the transmitter and an attenuation model to obtain an estimate for $\text{snr}_{t,i,j}$ and calculate $C_{t,i,j}$ under the assumption of small deviation of the SNR from its nominal value.

**Multiple Packet Transmission**

In a communication session between nodes $i$ and $j$, multiple packets may be transmitted and acknowledgments are received for successful packets. By knowing the number of symbols needed for a previous successful transmission, and assuming channel conditions do not change much between consecutive packets, the transmitter can reverse (8.9) to estimate $C_{t,i,j}$. In the case of unsuccessful (and thus unacknowledged) previous packets, we gradually update the assumed channel conditions. For an unsuccessful packet $m-1$ transmitted at time $t_{m-1}$, if $N_{t_{m-1},i}^{\text{actual}} < N_{t_{m-1},i}^{\text{max}}$, we have

$$
p_{t_{m-1},i,j}^{\text{RS}} > \hat{p}_{t_{m-1},i,j}^{\text{RS}},$$
$$p_{t_{m-1},i,j}^{\text{e}} > \hat{p}_{t_{m-1},i,j}^{\text{e}}.
$$

Thus, we monotonically increase $\hat{p}_{t,i,j}^{\text{RS}}$ or $\hat{p}_{t,i,j}^{\text{e}}$ till a threshold is reached or the number of retransmitted packets exceeds a maximum $P$, after which failure is declared and the packet is dropped. Given distance information between $i$ and $j$, $d_{t_{m},i}$, the above threshold can be calculated from an upper bound on the signal attenuation (e.g., Chapter 4). Note that the above process does not involve direct estimate of link quality and does not require additional communication overhead other than acknowledgments, which are already part of any ARQ protocol.

We observe the similarities in the problem of increasing $\hat{p}_{t,i,j}^{\text{RS}}$ or $\hat{p}_{t,i,j}^{\text{e}}$, and the congestion avoidance mechanism of the TCP-IP protocol [152]. In TCP-IP, congestion avoidance is required to manage failures in packet transmission by changing the congestion window, such that for each unacknowledged packet the maximal window size is halved and the window size is reduced to its initial value. Adopting the same strategies, we set

$$
\hat{p}_{t_{m},i,j}^{\text{RS}} = \hat{p}_{t_{m-1},i,j}^{\text{RS}} \cdot x_{\text{MSC}},$$
$$\hat{p}_{t_{m},i,j}^{\text{e}} = \hat{p}_{t_{m-1},i,j}^{\text{e}} \cdot x_{\text{MEC}},
$$

where $x_{\text{MSC}} > 1$ and $x_{\text{MEC}} > 1$ control the trade-off between energy consumption for transmission and the number of packets needed till successful decoding, i.e., network latency. An illustration of the above procedure is presented in Figure 8.3.

### 8.2.3 Extension to IR-HARQ

The process of optimizing $N_{t,i}^{\text{actual}}$ is extended next to use in the IR-HARQ protocol (cf. [85]). In the IR-HARQ protocol, instead of packet-wise decoding, the designated
receiver accumulates all demodulated symbols from previous packets. In turn, the source keeps transmitting packets until an acknowledgment of successful decoding is received. In this section, we describe how our adaptive coding scheme can be embedded in such protocol.

Consider unsuccessful transmission of previous packets, \( n = m', \ldots, m - 1 \). Assuming channel conditions do not change much from packet \( m' \) to \( m \), we modify (8.9) into

\[
N_{t_m,i}^{\text{actual}} = \min \left( \frac{(1 + \delta) KL - C_{t_m,i,j} \sum_{n=m'}^{m-1} N_{t_n,i}^{\text{actual}}}{C_{t_m,i,j}}, N_{t_m,i}^{\max} \right), \tag{8.14}
\]

and \( C_{t_m,i,j} \) is updated using the same process illustrated in Figure 8.3 for multiple packet transmission. Similar to the case of multiple packet transmission discussed in Section 8.2.2, the initial estimate \( C_{t_{m'},i,j} \) required to calculate \( N_{t_{m'},i}^{\text{actual}} \), is set by the number of symbols, \( \sum_{n=m''}^{m'-1} N_{t_n,i}^{\text{actual}} \), needed for successful decoding of a previous message accomplished after \( m' - m'' \) packet transmissions.

### 8.3 Implementation

We now describe practical implementation schemes for our adaptive coding approach. Let

\[
N_{t_m,i}^{\max} = N_{t_m,i}^{\min} + \frac{d_{t_m,i}^{\max}}{T_b \cdot c}. \tag{8.15}
\]
Chapter 8. Adaptive Error-Correction Coding Scheme for UWANs

Figure 8.4: Illustration of adaptive coding implementation for single packet transmission.

Since $N_{\text{min}}$ and $N_{\text{max}}$ are set by the maximal and minimal propagation delay of the system, respectively, by (8.3), $N_{\text{min}} \leq N_{t,i} \leq N_{\text{max}}$. While node $i$ has knowledge of $\Delta t_i$ and thus can calculate $N_{t,i}^{\text{max}}$ to set $N_{t,i}^{\text{actual}}$ according to (8.9) or (8.14), this may not be the case for its receiver $j$, who is aware of only $N_{\text{min}}$, $N_{\text{max}}$, and $K$. This is because node $j$ may not be aware of the distance of node $i$ to all nodes within its interference range. Considering this problem, one possibility for node $i$ is to transmit the value $N_{t,i}^{\text{actual}}$ as a separate header packet. If it is undesirable to transmit this side information, which of course takes away from the available guard time for sending extra parity symbols, the iterative decoding scheme illustrated in Figure 8.4 can be applied for the case of a single packet transmission. In this case, the receiver makes multiple decoding attempts with the first attempt starting after receiving enough information bits to satisfy (8.8). If decoding fails (e.g., the cyclic-redundancy check (CRC) did not pass), receiver $j$ makes another attempt after having obtained at least one more demodulated symbol. Decoding attempts are stopped when a post-amble signal to mark the position of the last symbol in each packet is detected or when at most $M_{\text{try}} = N_{\text{max}} - (1 + \delta)K + 1$ decoding attempts are made.

Next, we suggest two implementation examples for our adaptive coding approach.

**Bank-of-Codes**

The first implementation uses a pre-defined bank of up to $N_{\text{max}} - (1 + \delta)K$ codes, which can be a set of either optimized codes for different rates or punctured codes. For the latter, we apply an $(K, N)$ mother code, where $N$ is pre-defined at both transmitter and receiver. Since $N_{t,i}^{\text{actual}} \leq N_{t,i}^{\text{max}}$, for a single packet transmission,
Chapter 8. Adaptive Error-Correction Coding Scheme for UWANs

\( N \geq N^{\text{max}} \), while for IR-HARQ, \( N \geq P \cdot N^{\text{max}} \) (recall \( P \) is the maximum allowed number of packet retransmissions). At its designated time slot \( t \), node \( i \) transmits \( N^{\text{actual}}_{t,i} \) symbols by puncturing \( N - N^{\text{actual}}_{t,i} \) symbols of the codeword, following a predefined puncturing pattern, which, in the case of an IR-HARQ, is different for every packet transmission.

Rateless Codes

The second implementation example is based on Fountain codes [153]. At time \( t \), the transmitter generates a codeword whose length corresponds to \( N^{\text{actual}}_{t,i} \) symbols. Due to the rateless nature of Fountain codes, generation of any number of parity symbols is easily facilitated, and the process is similar in both single packet transmission and IR-HARQ. No side information is needed at the receiver assuming that a common seed for the random generation of the columns of the generator matrix is used. The same adaptive coding and successive decoding procedure can be applied to variants of Fountain codes, most notably Raptor codes [154]. For Raptor codes, we use an \((K, N^{\text{outer}})\) outer error-correction code and an inner \((N^{\text{outer}}, N^{\text{actual}}_{t,i})\) Fountain code, where \( N^{\text{outer}} = \beta \cdot K \) and \( \beta \) is a design parameter controlling the maximal probability of the failure of the inner Fountain decoding.

With regards to Fountain codes, using LT Fountain codes [155] has a benefit if the transmitter-receiver link can be modeled as an erasure channel. In this case, message passing decoding is alike successive cancelation, and additionally demodulated symbols available at the \( m \)-th decoding attempt can be used directly to improve the result from the \((m - 1)\)-st decoding stage (see [153]). Here, decoding with and without knowledge of \( N^{\text{actual}}_{t,i} \) are in fact identical, and thus there is no complexity overhead due to the iterative decoding procedure from Figure 8.4. However, since in underwater acoustic communication transmission rate is on the order of a few kbps and packets are small [9], we expect \( K \) to be on the order of a few tens to thousand symbols. Therefore, since popular LT and Raptor codes perform well only for large code word lengths, for our numerical results we apply the Fountain coding scheme described in [156], where good performance results are obtained for information word lengths as low as \( K = 100 \), at the cost of somewhat increased decoding complexity. When the channel cannot be modeled as an erasure channel, the integration of newly arrived samples into message-passing decoding is somewhat more complicated. For this case, favorable decoding schedules are described in [157] and [158].

Discussion

Comparing the above Bank-of-Codes and Rateless coding schemes, we note that for the former, since often there are restrictions for the code design parameters\(^{24}\), \( N \) may be much greater than \( N^{\text{max}} \) (for a single packet transmission) or \( P \cdot N^{\text{max}} \) (for

\(^{24}\)For example, for Reed-Solomon codes \( N \) must be equal to \( 2^n - 1 \) for some integer \( n \) [151].

176
IR-HARQ), and encoding and decoding complexity may be greater than that of the rateless scheme. However, in the Bank-of-Codes scheme, the punctured or pre-defined code can specifically be optimized for a rate and thus achieve better performance than the rateless code.

8.4 Performance Results and Discussion

We now evaluate the performance of our adaptive coding scheme in terms of the PER, energy consumption for transmission, and throughput using numerical simulations. Furthermore, to show the effect of a realistic sea environment, we also present results from a sea trial, where we implemented the punctured RS scheme in an actual underwater acoustic modem (namely, the Deep-Link modem).

8.4.1 Simulations

Setting

Our simulation setting includes a Monte-Carlo set of 10000 channel realizations. For each channel realization, four nodes are uniformly randomly placed in a square area of $2000 \times 2000$ m$^2$ at a fixed depth of 40 m for a 50 m long water column. The nodes operate for 100 sec in a TDMA network. Since in this paper, we are more interested in showing the possible gain of using our adaptive coding scheme rather than the absolute decoding capability, for simplicity in our simulations we adopt a binary erasure channel (BEC) model (i.e., MEC with $M = 2$) with binary phase shift keying (BPSK) modulation. The symbol-erasure probability is determined based on the transmitter-receiver link distance, $d$. More specifically, for each $d$, we calculate the receiver-side SNR using the Bellhop ray-tracing simulator (cf. [2]) for a flat sand surface, a common power source level of 130 dB//µPa@1m, and a noise level of 50 dB//µPa/Hz. Considering the large set of channel realizations, instead of running Bellhop for the entire bandwidth, we used the Bellhop results for the carrier frequency, which was set to 15 kHz. The channel erasure rate is then determined from the calculated SNR considering the BPSK modulation. The output of the Bellhop simulator in terms of the transmission loss as a function of range is given in Figure 8.5. From the figure we observe that for different ranges we may get the same transmission loss and thus channel erasure rate, which is due to the shadowing characteristics of the underwater acoustic channel [5]. An approximate model of this transmission loss is

$$TL = \gamma \log_{10}(d) + \alpha d/1000 ,$$

(8.16)

where $d$ is the transmission distance, $\gamma$ is the propagation loss with typical values between 10 and 30 [5], and $\alpha$ is the absorption loss which for a carrier frequency of 15 kHz is roughly 2 dB/km [5]. To allow channel changes (which affects $C_{t,i,j}$ from (8.8)), during the network operation we let the nodes drift between the transmission
Chapter 8. Adaptive Error-Correction Coding Scheme for UWANs

Figure 8.5: Transmission loss vs. range. Output from the Bellhop simulator [2].

of packets, and calculate the snr and the $p_e$, accordingly. Drift motion is simulated using the Shallow Water Hydrodynamic Finite Element Model (SHYFEM) ocean current model [109]. The model is set for location 49°16′33″N, 126°16′6.4″W (i.e., near Vancouver, BC), and a channel structure with two underwater hills at depth 30 m located at both corners of the considered square area.

The transmitter sends short information messages of $K = 456$ bits with a transmission rate of 500 bps, $T_h = 0.1$ sec, and an original code rate of $K/N_{\text{min}} \approx 0.9$. The duration of the time slot is determined according to a maximal detection range of 1500 m and a propagation speed of 1500 m/sec. Hence, using the adaptive coding scheme, at most, i.e., for $d_{t,i} = 0$ m, 1000 coded symbols can be transmitted to form a coding rate of $K/N_{\text{max}} \approx 0.45$. Packets are retransmitted until successful reception is obtained at the intended receiver. This scenario mimics the exchange of navigation packets in the Deep-Link system [133]. During the network operation, out of the four nodes, we choose a transmitter and receiver, and set $N_{t,i}^{\text{actual}}$ according to the maximal distance of the transmitter from its intended receiver and the nodes scheduled to transmit in time slots $t$ and $t + 1$. While due to strong multipath and clock offsets ranging errors are expected, in Chapters 2 and 4 we have shown that these errors can be limited to a few meters. Hence, given the low transmission rate, the effect of an overestimated interference range would be a minor interference of 1-3 symbols colliding with a later packet. For this reason, we assume that the interference range is accurately estimated at the transmitter. Nevertheless, due to the simulated drift motion and the fact that $d_{t,i}$ is estimated before the actual transmission, small inaccuracies in the estimation of $d_{t,i}$ do exist.

We evaluate performance of IR-HARQ using the adaptive coding scheme with fully utilized time-slot ($FIR$-HARQ), and IR-HARQ using optimized number of symbols.
(OIR-HARQ) To show the effect of using the IR-HARQ we also show results for a single packet transmission with a fixed coding gain (Single). For the considered schemes, the maximal number of packet retransmissions allowed until a message is declared failed is $P = 5$. For the OIR-HARQ, the threshold for the updating of $\hat{p}_{t,i,j}$ (see Section 8.2.2) is calculated from model (8.16) for $\gamma = 30$ and $\alpha = 3$ as an upper bound for power attenuation in the channel. Similarly, we use $\gamma = 10$ and $\alpha = 1.5$ as a lower bound on power attenuation to calculate the initial condition $\hat{p}_{1,i,j}$ at the beginning of the network operation, and $\delta = 0.1$ for the ratio of required information bits in (8.8). As a worst case scenario for the comparison with the fixed-rate code, we assume acknowledgments are always received.

For the case of a single packet transmission (denoted above as Single), we demonstrate the gain of the adaptive coding scheme by showing performance of different coding implementations: RS (fixed (F-RS) and punctured (P-RS)) with coded word of 8 bits, Fountain [156], and Raptor codes using an RS outer code (we note that similar values were obtained for Raptor codes using a low-density parity check (LDPC) outer code). As performance of the RS scheme varies greatly with the erasure patterns, we show performance results for both i.i.d. and burst-wise erasures. The former is motivated by the ambient noise in the channel, and the latter by temporarily correlated waves and ships-induced noises [5]. Finally, for the Raptor codes we use $\beta = 1.2$ (see Section 8.2). A software implementation of these protocols can be downloaded from [96].
Chapter 8. Adaptive Error-Correction Coding Scheme for UWANs

Figure 8.7: Channel utilization as a function of channel erasure probability (for $\Delta_{t,i} \approx 650$ m).

Results

We start off by comparing performance of the adaptive coding scheme to the fixed coding scheme for a single packet transmission, where for the former we transmit $N_{t,i}^{\text{max}}$ symbols and for the latter we transmit $N_{t,i}^{\text{min}}$ symbols. In Figure 8.6, we show the PER as a function of the erasure probability of the channel. For each value of erasure probability, we average the PER for the cases where the actual erasure probability is within a range of 0.02 from the one indicated through a marker in Figure 8.6. First, we observe that the performance of the RS scheme for burst-wise erasures is significantly better than that for i.i.d. erasures. This is due to the binary (BPSK) transmission that leads to more erroneous RS symbols (recall we used a coded word of 8 bits) and emphasizes on the suitability of RS coding in burst-noise channels. However, from Figure 8.6 we also see that even for i.i.d. erasures, the different adaptive coding schemes significantly outperform that with the fixed-rate RS code. The implementation with Fountain and Raptor codes\textsuperscript{25} shows its advantage for memoryless channels (i.i.d. erasures\textsuperscript{26}), and again greatly outperforms the fixed-rate coding scheme.

Next, we investigate channel utilization, defined as the rate of successful packets, for the different adaptive schemes. Since performance changes with both channel erasure probability and $\Delta_{t,i}$ from (8.2), in Figures 8.7 and 8.8 we show channel utilization as a function of the channel erasure probability and $\Delta_{t,i}$, respectively. The former figure is obtained by calculating channel utilization for cases where $\Delta_{t,i}$

\textsuperscript{25}The results of the Raptor codes are superimposed to the one for fountain codes.

\textsuperscript{26}We note that for rateless codes similar results were obtained for the case of burst-type erasures.
Figure 8.8: Channel utilization as a function of $\Delta_{t,i}$ (for erasure probability of roughly 0.21).

is within the range of $640 \pm 64$ m, and the latter for cases where the channel erasure probability is within the range of $0.21 \pm 0.02$. Note that since we consider four nodes and TDMA, channel utilization is bounded from above by $\frac{1}{4}$. However, it can considerably degrade if retransmissions are needed. For example, as also shown by the results in Figure 8.6, for the considered erasure rate in Figure 8.8, only a few packets were properly decoded using the fixed RS code and channel utilization is almost zero. From both figures, we observe a significant gain of the adaptive coding schemes over the fixed-rate RS scheme. From Figure 8.8, it can be seen that, as expected, performance improves as $\Delta_{t,i}$ increases. As in Figure 8.6, the best performance is achieved by the punctured RS scheme for burst-wise erasures with only a very small performance loss for the Fountain and Raptor coding schemes. In fact, from Figures 8.7 and 8.8 we observe that for these schemes starting from $\Delta_{t,i} \approx 150$ m and channel erasure rate of up to 0.3, channel utilization is maximal, i.e., no retransmissions are required.

We now compare performance of the considered three protocols, namely: Single, FIR-HARQ, and OIR-HARQ. Due to the small difference between performance of the adaptive coding schemes, in the following we present results of only the binary Fountain code implementation. In Figure 8.9, we show the error $\rho_p(m) = |p^e - \hat{p}^e|$ as a function of packet number $m$ for different values of $x_{BEC}$. Since motion in our simulations is restricted to drifting, $p^e$ does not change rapidly, and $\rho_p(m)$ reflects convergence of estimate $\hat{p}^e$. Since we bound the increase of $\hat{p}^e$, $\rho_p(m)$ decreases fast after three packet transmissions. We also observe that the initial error (i.e., $\rho_p(1)$) increases with $x_{BEC}$. This is because the number of transmitted symbols till decoding increases with $x_{BEC}$, and the former is used to determine the initial $\hat{p}^e$. 

181
Clearly, $\rho_p$ decreases for smaller $x_{BEC}$ values. This means less transmitted symbols per packet, but at the cost of transmitting more packets. However, as observed in Figure 8.9, the difference is not significant. To determine the effect of $x_{BEC}$ on network latency, in Figure 8.10 we show a histogram of the number of packets needed till decoding is possible. We observe that failure rate of the Single scheme is far greater than that of the IR-HARQ schemes. In fact, no failures are detected for the FIR-HARQ scheme, which utilizes the entire time slot. As expected, latency decreases for higher values of $x_{BEC}$. However, the latter characteristic comes at the cost of energy consumption for transmission, which improves as the number of transmitted symbols needed for successful decoding, $N_{\text{final}} = \sum_{m=1}^{P} N_m$, decreases. This is observed in Figure 8.11, where we show the complementary density function (CDF) of $N_{\text{final}}$. We note the large variance of $N_{\text{final}}$ for the FIR-HARQ method, for which the number of redundancy symbols greatly increases as more packets are required for decoding. Due to the adaptive transmission in the OIR-HARQ schemes, their $N_{\text{final}}$, and thus energy consumption for transmission, is significantly lower than the FIR-HARQ scheme. Here, we observe that $N_{\text{final}}$ does not change much for $x_{BEC} = 1.5$ and 1.2, which implies that our adaptive coding scheme is not very sensitive to the choice of this parameter.

Finally, in Figure 8.12 we show the CDF of the network goodput, defined as the number of delivered information bits during the network operation of 100 sec. As expected, goodput of the IR-HARQ schemes is considerably higher than that of the Single scheme. We also note that goodput is maximal for the FIR-HARQ scheme (at the cost of energy-consumption for transmission). However, not much difference is ob-
Chapter 8. Adaptive Error-Correction Coding Scheme for UWANs

Figure 8.10: Histogram of network latency in terms of number of packets needed till decoding.

Figure 8.11: CDF of the number of symbols transmitted till successful decoding, \(N_{\text{final}}\).
served comparing performance of FIR-HARQ and OIR-HARQ with $x_{\text{BEC}} = 1.5$. By these results, we argue that utilizing the time slot to adaptively adjust the code rate decreases network latency and increases network goodput. Furthermore, optimizing the coding rate in an IR-HARQ scheme significantly reduces energy consumption for transmission at a small cost in network goodput.

### 8.4.2 Sea Trial Results

To measure the performance of our adaptive scheme in a realistic sea environment, in November 2009 we conducted an experiment in the Haifa harbor, Israel. The experiment included three Deep-Link modems, statically deployed from the harbor docks as shown in Figure 8.13. The distance was 340 m between nodes 1 and 2 (Link 1), 780 m between nodes 1 and 3 (Link 2), and 910 m between nodes 2 and 3 (Link 3). The modems used RS codes with flexible parameters. Erasure decoding was performed, where erasures were declared based on thresholding the received signal samples. Using this capability, we tested both the fixed and punctured RS coding schemes. During the experiment, the three nodes periodically broadcasted short packets of 200 bits at a rate of 300 bps in a TDMA scheduling scheme, where in each time frame first node 1 transmits, then node 2, and then node 3. Each packet included a header of 0.67 sec for a synchronization signal and preamble sequence, and each time slot included a guard interval accounting for possible clock drifts and a maximal propagation delay of 1 sec. The latter was determined according to the size of the harbor, which was roughly 1500 m. We compared the performance of a single packet transmission using the punctured RS scheme with a fixed RS code of rate 4/5, such that the duration of the time slot was roughly 2.5 sec, and at most,
The experiment included two parts each lasting for one hour. In the first part we measured performance of the fixed RS coding scheme, and in the second part the punctured RS coding scheme was tested. Signals were transmitted at a low source level of 130 dB/µPa@1m, and as in our simulations we used model (8.16) to estimate the symbol error rate and the capacity required for evaluation of $N_{t,i}^{\text{actual}}$.

In Figure 8.14, we show the packet success rate, $P_{\text{success}}$, for the three links. Similar to the simulated results in Figure 8.6, we observe a significant gain of the adaptive over the fixed coding scheme, which is 14% on average. The success rate increases also for Link 1, where transmission distance is short and thus the PER has been low even for fixed-rate coding. We can thus conclude that the proposed adaptive coding scheme is indeed a viable approach to increase network goodput and decrease energy consumption in a realistic UWAN.

### 8.5 Conclusions

In this paper, we suggested an adaptive coding approach to improve the packet error rate in a time-slotted UWAC network. Different from usual adaptive coding approaches, our approach uses range information as the basis for adaptation of the code rate. This allows us to exploit the difference between the worst-case and the actually required guard time. By optimizing the number of parity check symbols transmitted in both single packet transmission and multiple packet transmission using IR-HARQ, we managed to control the trade-off between goodput and energy consumption for...
transmission. We described two implementations for our approach, using punctured and rateless codes. By means of analysis and simulation results, we demonstrated the advantages of our adaptive coding scheme over fixed-rate error correction in terms of packet error rate, transmission energy consumption, and throughput. The simulation results were verified in a sea trial.
Part III

Summary of Thesis and Further Research
In this thesis, we explored the problems of UWL and spatial-reuse MAC design for UWANs. We offered localization and tracking solutions that combat the continuous and irregular motion of nodes in the channel, lack of time-synchronization between the network nodes, uncertainties in propagation speed information, and the effect of measurement errors. We then used the developed localization capability to design spatial-reuse scheduling algorithms for both unicast and broadcast transmissions and a location-dependent adaptive coding technique. The proposed methods were tested in a developed simulator combining numerical models for both ocean current and power attenuation in the channel. The results were verified in four sea experiments of different channel bathymetry structures, using both industry and self-developed underwater acoustic modems. We now summarize the contributions of the thesis and propose possible future research directions.

- In Chapter 2, we have described a new algorithm for joint time-synchronization and localization for UWANs. Our algorithm is based on packet exchanges between anchor and unlocalized nodes, makes use of INS measurements to obtain accurate short-term motion estimates, and exploits the permanent motion of nodes. Our solution also allows self-evaluation of the localization accuracy. Using simulations, we have compared our algorithm to two benchmark localization methods as well as to the Cramér-Rao bound. The results demonstrate that our algorithm achieves accurate localization using only two anchor nodes and outperforms the benchmark schemes when node synchronization and knowledge of propagation speed are not available. Moreover, we have reported results of a sea trial where we validated our algorithm in open sea.

- In Chapter 3, we have presented a new UT scheme which considers sound speed uncertainties, incorporates Doppler shift measurements, and utilizes (possible) spatial correlation of ocean current to estimate the drift velocity of the TN as a combination of the drift velocities of anchor nodes. The latter is a distinctly new feature of our UT approach, which increases resilience to water current irregularities. We have also offered two types of unbiased confidence indexes aimed to control the use of drift velocity estimation. To evaluate the performance of our UT scheme, we have employed a hybrid simulator that combines numerical models for the ocean current and the signal-power attenuation in the ocean. We have also reported results from two sea trials conducted in the Mediterranean Sea and in the Indian Ocean. By tracking the sound speed, and utilizing Doppler-shift measurements and drift velocity information of anchor nodes, accuracy is significantly improved.

- In Chapter 4, we have utilized variations of ToF measurements due to mobility of nodes and have presented an algorithm to classify the former into LOS and NLOS links. First, by comparing signal strength-based and ToF-based range measurements, we have identified object-related NLOS links, where signals are
reflected from objects with high reflection loss, e.g., ships hull, docks, rocks, etc. In the second step, excluding ToF measurements related to ONLOS links, we have used a constrained expectation-maximization algorithm to classify ToF measurements into two classes: LOS and sea-related NLOS, and to estimate the statistical parameters of each class. Results from simulations and three sea trials demonstrate a high detection rate of ONLOS links, and accurate classification of ToF measurements into LOS and SNLOS.

• In Chapter 5, we have pointed out the problem of DR navigation for vessels located close to or on the sea surface, where, due to ocean waves, the vessel pitch angle is fast time-varying and its estimation via direct measurements of orientation is prone to drifts and noises. We have suggested a method to compensate on the vessel pitch angle using only a single acceleration sensor. First, our method classifies acceleration measurements into states of similar pitch angles. Then, for each class, we project acceleration measurements into the reference coordinate system along the vessel heading direction, and obtain distance estimations by integrating the projected measurements. Results in both simulated and actual sea environment demonstrate good DR performance using only acceleration measurements.

• In Chapter 6, we have utilized the long propagation delay in the UAC and the (possible) sparsity of the network topology, and have formalized conditions for which a node can transmit unicast packets even when it is located within the communication range of a node participating in an active communication session. We have considered these conditions as design constraints and have presented a distributed CA handshake-based MAC protocol, which, by jointly applying spatial and time reuse techniques, greatly improves channel utilization at the price of some reduction in fairness in resource allocation.

• In Chapter 7, we have addressed the problem of spatial-reuse scheduling in UWANs that support frequent transmission of broadcast packets and require robustness to inaccurate topology information. We have derived a broadcast scheduling algorithm that combines topology-transparent and topology-dependent spatial-reuse scheduling methodologies to achieve high throughput in static and dynamic topology scenarios. Results show that our scheduling algorithm achieves a favorable tradeoff between network throughput and robustness to outdated topology information due to topology changes, and that it also achieves fairness in terms of per node throughput.

• In Chapter 8, we have proposed a new adaptive coding method to maximize goodput in TDMA UWANs through time and spatial reuse by exploiting the surplus guard time that occurs for individual links for improving transmission reliability. In particular, using link distances as side information, transmitters
utilize the available portion of the time slot to adapt their code rate and increase reliability. Since increased reliability trades off with energy consumption for transmission, we have modified the code rate, considering both single and multiple packet transmission using the IR-HARQ algorithm. For practical implementation of this adaptive coding scheme we have considered punctured and rateless codes. Simulation and sea trial results demonstrate the gains achieved by our coding scheme over fixed-rate error-correction codes in terms of both throughput and consumption of transmitted energy per successfully delivered packet.

The following are suggestions for further research.

1. In Chapter 3, we have exploited spatial dependencies between the motion of nodes to offer an unbiased velocity estimate which increases the reliability of the motion model. The performance gap observed in Figure 3.4b relative to the Cramér-Rao Lower Bound was explained by the mismatch in the assumed covariance matrix for both model and measurement noise, as well as due to a non-linear spatial correlation of the ocean current. We believe that by applying Bayesian learning techniques both challenges can be overcome.

2. In Chapter 6, we have presented a handshake-based MAC protocol that carefully schedules transmissions to maximize the use of network resources by utilizing the long propagation delay in the channel. While the latter was also utilized in Chapter 8 by means of adaptive coding, the focus here was on reliability rather than throughput maximization. Hence, a possible extension of this work is a propagation delay-dependent approach for contention-free scheduling and for broadcast communication, i.e., a modification of the algorithms suggested in Chapter 7 to also utilize the long propagation delay.

3. In Chapter 8, we have suggested a heuristic algorithm to trade off reliability and energy consumption for time-slotted transmission of short packets by adaptively changing the channel coding rate. Potentially, by optimizing the number of parity symbols used, this tradeoff can be greatly improved. This would result in a new ARQ scheme that may be of interest to any energy constraint system. In addition, the rateless coding approach adopted in Chapter 8 relied on a model of binary erasure channel for the received symbols. By combining belief-propagation methods and interference identification techniques, still for short packets, more realistic non-erasure channel models can be considered.

4. Finally, while underwater acoustic channel modeling has greatly improved in the past decade, such models are still site specific, and results should be verified in sea experiments. However, such experiments are time and cost demanding, and are often hard to analyze. Moreover, sea experiments are hard to reproduce and thus comparison with previous UWAC schemes cannot be made. This
problem is referred to as the problem of reliability in UWAC. Therefore, along with the progress of channel modeling, there is a need to develop an open-source worldwide database of underwater acoustic channel measurements that would contain time samples of measured channel impulse response from various locations and time-of-year, as well as time-varying Doppler shift and noise measurements. Such database should also include a transformation system that for a given signal input and a chosen channel would produce an emulated output.
Bibliography


<table>
<thead>
<tr>
<th>Bibliography</th>
</tr>
</thead>
</table>


Bibliography


Appendix A

Alternating Optimization Approach for Solving (4.21)

In alternating optimization, a multivariate maximization problem is iteratively solved through alternating restricted maximization over individual subsets of variables \[120\]. Using the alternating optimization approach, we iteratively alternate parameters \(\omega_1\) and \(\omega_2\) to solve (21). The process is initialized by setting \(\upsilon_{m+1,0} = \upsilon_m\), \(\sigma_{m+1,0} = \sigma_m\), and \(\beta_{m+1,0} = \beta_m\), and ends after \(N_{\text{repeat}}\) iterations by setting \(\upsilon_{m+1} = \upsilon_{m+1,N_{\text{repeat}}}, \sigma_{m+1} = \sigma_{m+1,N_{\text{repeat}}}, \text{and } \beta_{m+1} = \beta_{m+1,N_{\text{repeat}}}, m = 1, 2.\)

Our objective is to find real-valued solutions

\[
\begin{align*}
\upsilon_{m+1,n+1} &= \arg\max_{\upsilon_m} f(\upsilon_m, \sigma_{m+1,n+1}, \beta_{m+1,n+1}) \\
\sigma_{m+1,n+1} &= \arg\max_{\sigma_m} f(\upsilon_{m+1,n+1}, \sigma_m, \beta_{m+1,n+1}) \\
\beta_{m+1,n+1} &= \arg\max_{\beta_m} f(\upsilon_{m+1,n+1}, \sigma_{m+1,n+1}, \beta_m),
\end{align*}
\]

satisfying constraints

\[
\begin{align*}
\upsilon_{1}^{p+1,n+1} - T_{\text{LIR}} &\leq \upsilon_1^{p+1,n} \\
\upsilon_1^{p+1,n+1} &\leq \upsilon_2^{p+1,n+1} + T_{\text{LIR}} \\
0 &\leq \sigma_1 \leq \min \left( \sigma_2^{p+1,n}, T_{\text{LIR}} \sqrt{\frac{\Gamma \left( \frac{1}{\beta_1^{p+1,n}} \right)}{\Gamma \left( \frac{3}{\beta_1^{p+1,n}} \right)}} \right) \\
\sigma_1^{p+1,n+1} &\leq \sigma_2 \leq T_{\text{LIR}} \sqrt{\frac{\Gamma \left( \frac{1}{\beta_2^{p+1,n}} \right)}{\Gamma \left( \frac{3}{\beta_2^{p+1,n}} \right)}}.
\end{align*}
\]

The above equations can be evaluated numerically by first finding the set of
Appendix A. Alternating Optimization Approach for Solving (4.21)

possible solutions for \( \nu_{m+1,n+1}, \sigma_{m+1,n+1}, \) and \( \beta_{m+1,n+1} \), solving

\[
\frac{\partial}{\partial \nu_m} f(\nu_m, \sigma_{m+1,n}, \beta_{m+1,n}) = \sum_{l=1}^{L} P_r(\lambda_l = m|\Lambda_l, \theta^p) \sum_{x_i \in \Lambda_l} \frac{|x_i - \nu_m|^{\beta_{m+1,n}-1} \beta_{m+1,n}}{(\sigma_{m+1,n}^{p+1})^{\beta_{m+1,n}}} \cdot S_{\text{gn}}(x_i - \nu_m) = 0
\]

\[
\frac{\partial}{\partial \sigma_m} f(\nu_{m+1,n+1}, \sigma_m, \beta_{m+1,n}) = \sum_{l=1}^{L} P_r(\lambda_l = m|\Lambda_l, \theta^p) \sum_{x_i \in \Lambda_l} \frac{1}{\sigma_m} + \frac{\beta_{m+1,n}}{(\sigma_m^{p+1+n+1})} \cdot S_{\text{gn}}(x_i - \nu_{m+1,n+1}) = 0
\]

\[
\frac{\partial}{\partial \beta_m} f(\nu_{m+1,n+1}, \sigma_{m+1,n+1}, \beta_m) = \sum_{l=1}^{L} P_r(\lambda_l = m|\Lambda_l, \theta^p) \sum_{x_i \in \Lambda_l} \frac{1}{\beta_m} + \frac{1}{\beta_m^2} \psi \left( \frac{1}{\beta_m^n} \right) - \left( \frac{|x_i - \nu_{m+1,n+1}|}{\sigma_{m+1,n+1}^{p+1}} \right)^{\beta_m} \log \left( \frac{|x_i - \nu_{m+1,n+1}|}{\sigma_{m+1,n+1}^{p+1}} \right) = 0
\]

where \( S_{\text{gn}}(x) \) is the algebraic sign of \( x \), and \( \psi(\cdot) \) is the digamma function, i.e., the first derivative of \( \log \Gamma(\cdot) \), that satisfy the above constraints. However, for integer \( \beta_{m+1,n} \leq 5 \) and approximating \( S_{\text{gn}}(x_i - \nu_m) \) with \( S_{\text{gn}}(x_i - \nu_m^{p+1,n}) \), both \( \frac{\partial}{\partial \nu_m} f(\nu_m, \sigma_{m+1,n}, \beta_{m+1,n}) \) and \( \frac{\partial}{\partial \sigma_m} f(\nu_{m+1,n+1}, \sigma_m, \beta_{m+1,n}) \) can be solved analytically.

Finally, in [120], it was proven that alternating maximization converges if for each alternation, problem constraints are handled internally (see also results in Figure 4).
Appendix B

Expressions for the HCRB

Denoting \( b_{i,m} = x_i - v_m \) and \( R = 4 \cdot (m - 1) \), for (4.24) we have

\[
F(\theta_r, \theta_1)_{j,q} = \begin{cases} 
\frac{\gamma_m}{\sigma_m^2} b_{i,m}^2 b_{i,m}^{\gamma_m - 2} \left[ \gamma_m - 2 \right] b_{i,m} \delta(b_{i,m}) \dagger, & j = R + 1, q = R + 1; \\
-\frac{2}{\sigma_m} \left[ \gamma_m + 1 \right] b_{i,m} \delta(b_{i,m}) \dagger, & j = R + 1, q = R + 2; \\
S_{gn}(b_{i,m}) \left[ \gamma_m - 2 \right] b_{i,m} \delta(b_{i,m}) \dagger, & j = R + 1, q = R + 3; \\
\left[ \gamma_m - 2 \right] b_{i,m} \delta(b_{i,m}) \dagger, & j = R + 2, q = R + 1; \\
0, & j = R + 2, q = R + 2; \\
\end{cases}
\]

where \( \delta(\cdot) \) is the Dirac delta function, and \( \psi(\cdot) \) and \( \psi'(\cdot) \) are the digamma and trigamma functions, i.e., the first and second derivative of \( \log \Gamma(\cdot) \), respectively.

Furthermore, defining \( a_{1,m} = \Gamma \left( \frac{1}{\gamma_m} \right) \), \( a_{2,m} = \Gamma \left( \frac{3}{\gamma_m} \right) \), and \( a_{3,m} = \frac{a_{1,2,m}}{a_{2,2}} \), we obtain

\[
\frac{\partial^2}{\partial \theta_j \partial \theta_q} \log p(\theta_r | \theta_1) = \begin{cases} 
\frac{1}{\left( T_{\text{LIN}} \sqrt{\frac{1}{\gamma_m} - 1} \right)^2} \left( 3 \gamma_{3,2} \gamma \left( \frac{3}{\gamma_m} \right) + \gamma' \left( \frac{3}{\gamma_m} \right) \right), & j = 2, q = 2; \\
-0.5 \left( T_{\text{LIN}} \sqrt{\frac{1}{\gamma_m} - 1} \right)^2 \left( \frac{3 \gamma_{3,2} \gamma \left( \frac{3}{\gamma_m} \right) + \gamma' \left( \frac{3}{\gamma_m} \right) - 0.5 \gamma' \left( \frac{3}{\gamma_m} \right) - 3 \gamma \left( \frac{3}{\gamma_m} \right) \right), & j = 3, q = 3; \\
-0.5 \left( T_{\text{LIN}} \sqrt{\frac{1}{\gamma_m} - 1} \right)^2 \left( \frac{3 \gamma_{3,2} \gamma \left( \frac{3}{\gamma_m} \right) + \gamma' \left( \frac{3}{\gamma_m} \right) - 0.5 \gamma' \left( \frac{3}{\gamma_m} \right) - 3 \gamma \left( \frac{3}{\gamma_m} \right) \right), & j = 7, q = 2; \\
\frac{\partial}{\partial \beta_2} \left[ \frac{0.5 T_{\text{LIN}} \sqrt{\frac{1}{\gamma_m} - 1} \gamma \left( \frac{3}{\gamma_m} \right) + \gamma' \left( \frac{3}{\gamma_m} \right) - 0.5 \gamma' \left( \frac{3}{\gamma_m} \right) - 3 \gamma \left( \frac{3}{\gamma_m} \right) \right], & j = 7, q = 7; \\
0, & \text{otherwise,}
\end{cases}
\]

where \( \Gamma'(\cdot) \) is the first derivative of \( \Gamma(\cdot) \).