Robust Cooperative Positioning for VANETs based on Multi Sensor System and Enhanced Estimation Algorithms

Azmir Hasnur Rabiain

Submitted in total fulfilment of the requirements of the degree of
Doctor of Philosophy

Department of Infrastructure Engineering
THE UNIVERSITY OF MELBOURNE

March 2018
Abstract

ROAD transportation injuries and wasted energy and time in traffic congestion cause considerable costs for society. The number of annual road deaths in Australia alone has averaged around 1,200 over the past 5 years, while the estimated economic cost is $27 million per annum. To mitigate this problem, there is great interest in developing intelligent transportation system (ITS). It is reported that this technology could address up to 82% of crashes involving unimpaired drivers. One of the fundamentals to realising a functioning ITS is, vehicular positioning. While standalone global navigation satellite system (GNSS) is able to provide positioning on a global scale, it fails to meet the criteria of ITS positioning accuracy particularly in urban environments where the required positioning accuracy for safety applications is at the lane level (1.1 m, 95% confidence level), with a communication latency of 0.1 second. This study aims to meet ITS positioning requirement by employing cooperative positioning (CP), which is an approach to solving vehicle locations through communication and ranging information from other vehicles within a network, known as a vehicular ad-hoc network (VANET). However, while CP does provide positioning improvement, in terms of accuracy, reliability and integrity in open sky environments, its implementation in dense urban areas remains a challenge due to restricted connectivity between vehicles, multipath, and unfavourable network geometry. To counter this problem, this study explores the possibility of exploiting existing, self-contained sensors such as MEMS inertial sensors and vehicular speed sensors, together termed as multi sensor system (MSS). However, due to the low cost nature of these self-contained sensors, their conventional implementations need to be enhanced through better error modelling and filtering techniques first, before being implemented as part of CP. The outcome of this work demonstrates the effectiveness of
CP with the inclusion of MSS in the pursuit of achieving ITS positioning requirements in dense urban environments. Using realistic simulation of navigation in dense urban environments and real world datasets, this study shows that MSS could significantly enhance CP, which in turn would provide better navigation solutions for the participating vehicles in VANET. Moreover, the outcome of this study highlights that lane level positioning accuracy is achievable in dense urban environments, using existing technologies and proposed techniques presented in this thesis.
Declaration

This is to certify that

1. the thesis comprises only my original work towards the PhD,

2. due acknowledgement has been made in the text to all other material used,

3. the thesis is less than 100,000 words in length, exclusive of tables, maps, bibliographies and appendices.

__________________________
Azmir Hasnur Rabiain, March 2018
Acknowledgements

To my parents, wife, daughter, brother and sister, I thank you all for the support you have given me.

I would like to thank my supervisor, Professor Allison Kealy for her constant guidance, encouragement and support and persistence in dealing with problems during my study. I could not ask for more.

My deepest appreciation to the staff and colleagues in the Department of Infrastructure Engineering, namely Simon Fuller, Tengku Ahmad Ridhwanuddin Tengku Baharuddin, Dr. Philip Collier and George fox.

Finally, I would like to acknowledge Prof. Guenther Retscher from Vienna University of Technology, Prof. Dorota Brzezinska and Dr. Charles Toth from Ohio State University, Prof. Terry Moore, Dr. Lukasz Bonenberg, Dr. Chris Hill from The University of Nottingham, Dr. Nima Alam, Dr. Joon Wayn and Dr. Feng from University of New South Wales and Dr. Vasillis Gikas from National Technical University of Athens for their useful help and advice.
# Contents

1 Introduction ................................................................. 1
   1.1 Motivation and background ........................................... 1
      1.1.1 ITS and Positioning Requirements ............................... 2
      1.1.2 Cooperative Positioning ........................................... 4
      1.1.3 Multi Sensor System .............................................. 5
   1.2 Research Aim, Hypothesis and Objectives .......................... 6
   1.3 Research Contributions .............................................. 7
   1.4 Thesis Outline ........................................................ 9
   1.5 Publications .......................................................... 11

2 Intelligent Transportation System ........................................ 13
   2.1 Introduction .......................................................... 13
   2.2 Intelligent Transportation System ................................... 14
      2.2.1 Current Implementation and Future Developments ............. 16
      2.2.2 ITS Summary .................................................... 19
   2.3 Global Navigation Satellite System .................................. 20
      2.3.1 Positioning with GNSS ........................................... 20
      2.3.2 User Equivalent Ranging Error ................................. 23
      2.3.3 Navigation in Urban and Semi-Urban Environments - A Case Study  26
   2.4 Inertial Measurement Unit ............................................ 34
      2.4.1 Introduction to IMU .............................................. 34
      2.4.2 IMU Mechanisation ............................................... 37
      2.4.3 Initialisation and alignment techniques ........................ 40
      2.4.4 Error Sources and Models ...................................... 40
   2.5 Vehicular Speed Sensor ............................................... 43
      2.5.1 Measurement Characteristics ................................... 44
   2.6 Wireless Communication for Cooperative Positioning ............... 46
      2.6.1 Dedicated Short Range Communication .......................... 46
      2.6.2 Ultra-Wide Band ............................................... 48
      2.6.3 Radio Range/Range-Rate CP Measurements ..................... 50
      2.6.4 Non-Radio Range CP Measurements .............................. 52
      2.6.5 Deployment and Ranging in a Wireless Ad-Hoc Environment ... 53
   2.7 Summary ............................................................... 56
5.3.2 UWB Range Characterisation ........................................ 143
5.3.3 UWB Measurement Model ........................................... 149
5.3.4 System Architecture .................................................. 150
5.3.5 Analysis Methodology ................................................ 152
5.3.6 Results ................................................................. 154
5.4 Summary ................................................................. 162

6 Multi-Sensor System and Cooperative Positioning: Analysis Using Realistic Simulated Datasets 165
6.1 Introduction ............................................................... 165
6.2 Generation of a VANET Simulation ................................ 166
6.2.1 Simulation of Urban Mobility (SUMO) ......................... 167
6.2.2 GEMV2 ................................................................. 168
6.3 Simulation set-up ........................................................ 169
6.3.1 Scenario 1 - Open Sky Environment ......................... 169
6.3.2 Scenario 2 - Dense Urban Environment ..................... 172
6.4 Analysis Methodology .................................................. 177
6.5 Results ................................................................. 179
6.5.1 Scenario 1 .............................................................. 179
6.5.2 Scenario 2 .............................................................. 181
6.6 Summary ................................................................. 184

7 Conclusion and Further Research 187
7.1 Introduction ............................................................... 187
7.2 Summary of Research Findings ...................................... 187
7.3 Conclusion ............................................................... 190
7.4 Recommendations for Future Studies ......................... 191

A GLONASS Positioning 197
A.1 GLONASS coordinates ............................................... 197
A.1.1 Coordinates transformation to an inertial reference frame 197
A.1.2 Numerical integration ............................................. 198
A.1.3 Coordinate transformation ....................................... 199

B WGS84 Parameters ....................................................... 201

C FSPAWN Clique Determination ....................................... 203

D Logarithmic Likelihood .................................................. 207
## List of Figures

1.1 Illustration of V2X (adopted from eblog.mercedes-benz-passion.com) . . . 2

2.1 Local and network-level observations . . . . . . . . . . . . . . . . . . . . . 14
2.2 Overview schematic of ITS . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15
2.3 Deflected signals (adopted from [34]) . . . . . . . . . . . . . . . . . . . . . . 25
2.4 Route of experiment . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28
2.5 Snapshot along Collins Street . . . . . . . . . . . . . . . . . . . . . . . . . 29
2.6 Snapshot along Rathdown Street . . . . . . . . . . . . . . . . . . . . . . . . 29
2.7 Position plot for segment 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30
2.9 Position Error for segment 2 . . . . . . . . . . . . . . . . . . . . . . . . . . . 30
2.8 Position plot for segment 2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . 31
2.10 Number of satellites for segment 1 . . . . . . . . . . . . . . . . . . . . . . . 32
2.12 Body Frame . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 38
2.13 VSS vs GNSS velocity . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 44
2.14 Speed sensor error . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 45
2.16 DSRC frequency allocation [72] . . . . . . . . . . . . . . . . . . . . . . . . . 47
2.17 Message passing route . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 54
2.18 Inter-vehicle ranging . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 54

3.1 Example of a FG representation . . . . . . . . . . . . . . . . . . . . . . . . . 68
3.2 Simulation setup . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 71
3.3 RMSE for simulation 2.1 (small prior distribution) . . . . . . . . . . . . . . 73
3.4 RMSE for simulation 2.2 (large prior distribution) . . . . . . . . . . . . . . 74
3.5 Simulation with concentrated prior . . . . . . . . . . . . . . . . . . . . . . . 74
3.6 Simulation with diffused prior . . . . . . . . . . . . . . . . . . . . . . . . . . 75
3.7 Probability density at iteration 1 and 2 . . . . . . . . . . . . . . . . . . . . . 77
3.8 Probability density at iteration 4 and 10 . . . . . . . . . . . . . . . . . . . . 77
3.9 Cliques and connection types in FSPAWN . . . . . . . . . . . . . . . . . . . . 84
3.10 RMSE over time for each node . . . . . . . . . . . . . . . . . . . . . . . . . 86
3.11 RMSE for simulation 2.2 (large prior distribution) . . . . . . . . . . . . . . 86

4.1 IMU/GNSS LC schematic . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 90
4.2 Tightly coupled schematic . . . . . . . . . . . . . . . . . . . . . . . . . . . . 91
4.3 Allan deviation plot of static accelerations . . . . . . . . . . . . . . . . . . . . 102
4.4 Allan deviation plot of static turning rates ........................................... 103
4.5 DAV plot of kinematic acceleration X .................................................... 104
4.6 ∆ Accelerations: CIMU/GNSS vs. MEMS XsensMTi ................................. 105
4.7 ∆ Turning Rates: CIMU/GNSS vs. MEMS XsensMTi ............................... 106
4.8 DAV plot of kinematic acceleration X .................................................... 106
4.9 DAV plot of kinematic turning rate Z ..................................................... 107
4.10 IMU/GNSS positioning error ................................................................. 108
4.11 AV-EKF vs. DAV-EKF 2D performance .................................................. 109
4.12 AV-EKF vs. DAV-EKF 3D performance .................................................. 110
4.13 Overall process of the extended FDE ..................................................... 115
4.14 Number of satellites (GPS only) ............................................................ 118
4.15 Satellite ranging error resulting from simulated multipath ....................... 119
4.16 2D RMSE for full GNSS availability ...................................................... 120
4.17Rejected satellites for s-2 and s-3 in scenario 1 ....................................... 122
4.18 Retained satellites for s-2 and s-3 ......................................................... 123
4.19Rejected satellites for s-2 and s-3 in scenario 2 ....................................... 124
4.20 n-sat and RMSE over time during 180 seconds of induced multipath error 125

5.1 Schematic of the proposed TC CP approach ........................................... 129
5.2 Double differencing technique ............................................................... 130
5.3 Tightly coupled CP Architecture .......................................................... 132
5.4 Trajectory of test site ............................................................................. 134
5.5 Pseudorange double differencing range ................................................. 135
5.6 Position error comparison with full GNSS availability ............................. 137
5.7 Position error - 3 satellites, 300 seconds outage ..................................... 138
5.8 Position error - 2 satellites, 300 seconds outage ..................................... 139
5.9 Experiment platforms consisting of two moving vehicles and two RSUs 141
5.10 Sensor platform .................................................................................... 142
5.11 UWB transceivers connections between platforms ................................ 144
5.12 Static UWB test ................................................................................... 145
5.13 Time synchronisation compensation for vehicle 1 to RSU 1 .................. 147
5.14 Sensors schematic and lever arm measurements ..................................... 147
5.15 Corrected range after lever arm compensation ...................................... 149
5.16 Matched ranges from UWB and GNSS ............................................... 149
5.17 Centralised integration architecture ....................................................... 151
5.18 Decentralised integration architecture .................................................... 152
5.19 Reference trajectory and area with simulated outage and multipath ....... 154
5.20 V2X connections for scenarios 1 and 2.1 ............................................. 155
5.22 GNSS CP vs. MSS CP 2D error ........................................................... 160
5.23 V2X connections in scenario 2.2 .......................................................... 161

6.1 Flow chart of VANET simulation ......................................................... 166
6.2 Open Street Map Extraction and Generated map in SUMO ..................... 167
6.3 Trajectory and RSU placement for scenario 1 ......................................... 170
6.4 Number of vehicles and V2X connections for scenario 1 ....................... 171
6.5 Number of vehicles with velocity threshold ........................................... 171
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.6</td>
<td>V2X connection visualisation for scenario 1</td>
<td>172</td>
</tr>
<tr>
<td>6.7</td>
<td>Trajectory and RSU placement for scenario 2</td>
<td>173</td>
</tr>
<tr>
<td>6.8</td>
<td>Number of vehicles and velocity profile</td>
<td>174</td>
</tr>
<tr>
<td>6.9</td>
<td>Number of vehicles and velocity profile</td>
<td>174</td>
</tr>
<tr>
<td>6.10</td>
<td>V2X connection visualisation for scenario 2</td>
<td>175</td>
</tr>
<tr>
<td>6.11</td>
<td>Number of V2X connections and averaged distance</td>
<td>175</td>
</tr>
<tr>
<td>6.12</td>
<td>Averaged satellite visibility throughout scenario 2</td>
<td>176</td>
</tr>
<tr>
<td>6.13</td>
<td>Satellite visibility and multipath effects for Veh ID 185</td>
<td>177</td>
</tr>
<tr>
<td>6.14</td>
<td>RMSE over time for scenarios 1.1 and 1.2</td>
<td>180</td>
</tr>
<tr>
<td>6.15</td>
<td>RMSE over time for scenario 2.2</td>
<td>183</td>
</tr>
</tbody>
</table>
# List of Tables

1.1 ITS positioning requirements [9] ................................................. 3

2.1 GNSS UERE Summary [26] .................................................. 26

2.2 Repeatability percentage of number of visible satellites ................. 32

2.3 Comparison of different IMU grades .................................. 36

2.4 UWB performance using coherent integration technique ............... 50

3.1 Averaged RMSE for varying algorithms ................................ 73

3.2 Averaged RMSE for varying algorithms ................................ 84

4.1 IMU stochastic Noise Terms ............................................... 100

4.2 Specifications of CIMU and Xsens MTi-G IMU ....................... 101

4.3 IMU A/VRW coefficients ................................................ 103

4.4 IMU bias instability coefficients ...................................... 103

4.5 A/VRW coefficients from DAVAR static ................................ 105

4.6 IMU A/VRW coefficients from the DAVAR kinematic test ......... 107

4.8 2D maximum error during GNSS outage ................................ 110

4.9 3D maximum error during GNSS outage ................................ 110

4.10 Given codename for each navigation technique ...................... 119

4.11 Mean and maximum 2D RMSE at varying multipath error magnitude 121

4.12 Mean and maximum 2D RMSEs at varying time durations ........... 122

4.13 Mean 2D RMSEs at varying error magnitudes .......................... 123

4.14 Mean and maximum 2D RMSE at varying time duration .............. 125

5.1 Full GNSS coverage mean RMSE and maximum error ................. 137

5.2 Partial GNSS 3 satellites 2 RMSE and maximum error .............. 139

5.3 Partial GNSS 3 satellites mean RMSE and maximum error .......... 140

5.4 Sensors deployed on each platform ................................... 142

5.5 Sensor specifications .................................................... 143

5.6 Total station vs. UWB distance measurements .......................... 145

5.7 Summary of the different types of scenarios ............................ 154

5.8 RMSE derived from scenario 1.1 ....................................... 156

5.9 Mean RMSE and maximum error derived from scenario 1.2 ........... 156

5.10 Mean RMSE and maximum error derived from scenario 1.3 ........... 157

5.11 Mean RMSE and maximum error for scenario 2.1 ..................... 159
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AHRS</td>
<td>Attitude and heading reference system</td>
</tr>
<tr>
<td>ARMA</td>
<td>Auto-regressive moving average process</td>
</tr>
<tr>
<td>AV</td>
<td>Allan variance</td>
</tr>
<tr>
<td>BPSK</td>
<td>Bi-phase shift key</td>
</tr>
<tr>
<td>C/A</td>
<td>Course/Acquisition</td>
</tr>
<tr>
<td>CAPS</td>
<td>Chinese area positioning system</td>
</tr>
<tr>
<td>CDMA</td>
<td>Code-division multiple access</td>
</tr>
<tr>
<td>CFO</td>
<td>Carrier frequency offset</td>
</tr>
<tr>
<td>C-ITS</td>
<td>Cooperative intelligent transportation system</td>
</tr>
<tr>
<td>CP</td>
<td>Cooperative positioning</td>
</tr>
<tr>
<td>CRB</td>
<td>Cramer Rao Bound</td>
</tr>
<tr>
<td>CV</td>
<td>Connected vehicles</td>
</tr>
<tr>
<td>DAV</td>
<td>Dynamic Allan variance</td>
</tr>
<tr>
<td>DOP</td>
<td>Dilution of precision</td>
</tr>
<tr>
<td>DSRC</td>
<td>Dedicated short-range communications</td>
</tr>
<tr>
<td>DTG</td>
<td>Dynamically tuned gyroscope</td>
</tr>
<tr>
<td>ECEF</td>
<td>Earth-centered, earth-fixed</td>
</tr>
<tr>
<td>ECI</td>
<td>Earth centred inertial</td>
</tr>
<tr>
<td>eFDE</td>
<td>Extended fault detection and exclusion</td>
</tr>
<tr>
<td>EIRP</td>
<td>Effective isotropic radiated power</td>
</tr>
<tr>
<td>EKF</td>
<td>Extended Kalman filter</td>
</tr>
<tr>
<td>eLoran</td>
<td>Enhanced long range navigation</td>
</tr>
<tr>
<td>FCC</td>
<td>Federal communication commission</td>
</tr>
<tr>
<td>FDE</td>
<td>Fault detection and exclusion</td>
</tr>
<tr>
<td>FG</td>
<td>Factor graph</td>
</tr>
<tr>
<td>FIM</td>
<td>Fisher information matrix</td>
</tr>
<tr>
<td>FOC</td>
<td>Full operation capability</td>
</tr>
<tr>
<td>FOG</td>
<td>Fibre optic gyroscope</td>
</tr>
<tr>
<td>FSPAWN</td>
<td>Federated SPAWN</td>
</tr>
<tr>
<td>GAGAN</td>
<td>GPS aided geo augmentation navigation</td>
</tr>
<tr>
<td>GDP</td>
<td>Gross domestic product</td>
</tr>
<tr>
<td>GEO</td>
<td>Geostationary orbit satellites</td>
</tr>
<tr>
<td>GLONASS</td>
<td>Globalnaya Navigatsionnaya Sputnikovaya Sistema</td>
</tr>
<tr>
<td>GM</td>
<td>Gauss Markov</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>GNSS</td>
<td>Global navigation satellite system</td>
</tr>
<tr>
<td>GPS</td>
<td>Global positioning system</td>
</tr>
<tr>
<td>IMU</td>
<td>Inertial measurement unit</td>
</tr>
<tr>
<td>INS</td>
<td>Integrated navigation system</td>
</tr>
<tr>
<td>IOC</td>
<td>Initial operating capability</td>
</tr>
<tr>
<td>ITS</td>
<td>Intelligent transportation system</td>
</tr>
<tr>
<td>LBS</td>
<td>Location based services</td>
</tr>
<tr>
<td>LC</td>
<td>Loosely coupled</td>
</tr>
<tr>
<td>LNF</td>
<td>Local navigation frame</td>
</tr>
<tr>
<td>LOS</td>
<td>Line of sight</td>
</tr>
<tr>
<td>MDPC</td>
<td>Measurement directed progressive correction</td>
</tr>
<tr>
<td>MDSPAWN</td>
<td>Measurement directed SPAWN</td>
</tr>
<tr>
<td>MEMS</td>
<td>Microelectromechanical systems</td>
</tr>
<tr>
<td>MSS</td>
<td>Multi Sensor System</td>
</tr>
<tr>
<td>NAVSTAR</td>
<td>Navigation by satellite ranging and timing</td>
</tr>
<tr>
<td>NED</td>
<td>North east down</td>
</tr>
<tr>
<td>OBDII</td>
<td>On-board diagnostics II</td>
</tr>
<tr>
<td>OCX GPS</td>
<td>Next generation operational control segment</td>
</tr>
<tr>
<td>OFDM</td>
<td>Orthogonal frequency-division multiplexing</td>
</tr>
<tr>
<td>OSM</td>
<td>Open street map</td>
</tr>
<tr>
<td>P(Y)</td>
<td>Precise(Y)</td>
</tr>
<tr>
<td>PDF</td>
<td>Probability density function</td>
</tr>
<tr>
<td>PF</td>
<td>Particle filter</td>
</tr>
<tr>
<td>PPS</td>
<td>Precise positioning services</td>
</tr>
<tr>
<td>PRN</td>
<td>Pseudorandom noise</td>
</tr>
<tr>
<td>PSD</td>
<td>Power spectral density</td>
</tr>
<tr>
<td>PVA</td>
<td>Position, velocity and attitude</td>
</tr>
<tr>
<td>QAM</td>
<td>Quadrature amplitude modulation</td>
</tr>
<tr>
<td>RAIM</td>
<td>Receiver autonomous integrity monitoring</td>
</tr>
<tr>
<td>RF</td>
<td>Radio frequency</td>
</tr>
<tr>
<td>RFID</td>
<td>Radio frequency identification</td>
</tr>
<tr>
<td>RIG</td>
<td>Rate integrating gyroscope</td>
</tr>
<tr>
<td>RLG</td>
<td>Ring laser gyroscope</td>
</tr>
<tr>
<td>RMS</td>
<td>Root mean square</td>
</tr>
<tr>
<td>RMSE</td>
<td>Root mean square error</td>
</tr>
<tr>
<td>RNSS</td>
<td>Regional navigation satellite system</td>
</tr>
<tr>
<td>RSS</td>
<td>Received signal strength</td>
</tr>
<tr>
<td>RSU</td>
<td>Road side unit</td>
</tr>
<tr>
<td>RTK</td>
<td>Real time kinematic</td>
</tr>
<tr>
<td>RW</td>
<td>Random walk</td>
</tr>
<tr>
<td>SA</td>
<td>Selective availability</td>
</tr>
<tr>
<td>SF</td>
<td>Scale factor</td>
</tr>
<tr>
<td>SGE</td>
<td>Slow growth error</td>
</tr>
<tr>
<td>SPA</td>
<td>Sum product algorithm</td>
</tr>
<tr>
<td>SPAWN</td>
<td>Sum product algorithm over wireless network</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>SPP</td>
<td>Single point positioning</td>
</tr>
<tr>
<td>SSA</td>
<td>Solid state accelerometer</td>
</tr>
<tr>
<td>SVN</td>
<td>Space vehicle number</td>
</tr>
<tr>
<td>TC</td>
<td>Tightly coupled</td>
</tr>
<tr>
<td>TEC</td>
<td>Total electron count</td>
</tr>
<tr>
<td>U.S.</td>
<td>United States</td>
</tr>
<tr>
<td>UERE</td>
<td>User equivalent range error</td>
</tr>
<tr>
<td>UKF</td>
<td>Unscented Kalman filter</td>
</tr>
<tr>
<td>UWB</td>
<td>Ultra-wideband</td>
</tr>
<tr>
<td>V2I</td>
<td>Vehicle to infrastructure</td>
</tr>
<tr>
<td>V2V</td>
<td>Vehicle to vehicle</td>
</tr>
<tr>
<td>V2X</td>
<td>Vehicle to infrastructure/vehicle</td>
</tr>
<tr>
<td>VANET</td>
<td>Vehicular ad hoc network</td>
</tr>
<tr>
<td>VSS</td>
<td>Vehicle speed sensor</td>
</tr>
<tr>
<td>WAVE</td>
<td>Wireless access in vehicular environments</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless local area network</td>
</tr>
<tr>
<td>WLS</td>
<td>Weighted least squares</td>
</tr>
<tr>
<td>WSN</td>
<td>Wireless sensor network</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Motivation and background

Road transportation injuries and traffic congestions impose high human, financial and environmental costs on today’s society. The number of annual road deaths in Australia alone has averaged around 1,200 over the past 5 years while the estimated economic cost due to traffic congestion is $10 billion per annum [20]. Globally, it is estimated that around 1.2 million people are killed on the road every year, corresponding to an annual loss of 1-2% gross domestic product (GDP) [8,20,52]. The environment is also affected by road transportation, with one report [88] suggesting that 10.7% of global greenhouse gas emissions are caused by road transportation, of which road traffic congestions contribute a staggering 85% [21].

The introduction of intelligent transport systems (ITS) is seen as the solution to these problems. For example, [1] suggests that employing ITS could potentially affect 80% of road collisions in the U.S., and while a study conducted by Australasian road transport and traffic agencies (Austroads) and Monash University Accident Research Centre (MUARC) concludes that a 25-35% reduction in serious casualties could be achieved [113]. Furthermore, [115] estimates that ITS could allow highway capacity to be more efficient and increase by as much as 273%, hence also reducing the impact of greenhouse gas emissions.
1.1.1 ITS and Positioning Requirements

ITS is a term describing the use of information and communication technology in the transport network to improve transport outcomes. Some of its applications are collision avoidance, location based services (LBS), and fleet management. In some of the literature, such as in [82], the term cooperative-ITS (C-ITS) is introduced to emphasis a particular subset of ITS, where the different elements of the network, i.e., vehicles, roads and roadside units (RSUs), share information with each other. A network of vehicles is often called a vehicular ad-hoc network (VANET). C-ITS applications are often classified based on these types of interactions:

1. Vehicle-to-vehicle (V2V) communication :- where vehicles exchange information with each other.

2. Vehicle-to-infrastructure (V2I) :- where vehicles exchange information with the transport network infrastructure or RSU.

3. Vehicle-to-vehicle/infrastructure (V2X) :- a combination of V2V and V2I.

Figure 1.1: Illustration of V2X (adopted from eblog.mercedes-benz-passion.com)

The communications in ITS can be realised using dedicated short range communication (DSRC), which utilises the 5.9 GHz band of the spectrum. Its properties and avail-
1.1 Motivation and background

ability play a major role in its use for ITS throughout major jurisdictions such as in U.S., European Union, Japan, and South Korea. In Australia, the 5.9 GHz band is currently embargoed by the Australia Communications and Media Authority (ACMA), which has recognised its potential and has enabled planning for the use of ITS to progress [9, 82].

As reported in [9], prepared by Austroads, a key component of ITS is positioning. It proposes that emerging ITS applications would require vehicle positioning accuracy at one of three levels: the road level (metre), lane level (sub-metre), and where-in-lane level (decimetre). Furthermore, the report states that the emerging timeliness requirement for road level applications is about one second (1 Hz), while for lane level and where-in-lane level applications is 0.1 second (10 Hz). Moreover, other performance requirements for vehicle positioning systems are also critical, such as continuity, availability, integrity and interoperability. These proposed requirements are in line with the recommendations put forward in the SAE 2735 DSRC implementation plan, prepared by the Society of Automotive Engineers [104].

The global navigation satellite system (GNSS), is the most widely used positioning system due to its global availability and the fact that it is free of charge, is the most attractive candidate for providing absolute position to ITS users. However, GNSS alone could not satisfy the required accuracy and latency of all ITS applications as it is well known that standalone GNSS can typically only achieve accuracy of between 5 and 15 metres [54]. Hence, it cannot be used for safety related ITS applications such as collision
avoidance and lane level traffic guidance: these applications require lane or where-in-lane positioning information, which falls in the sub-metre and decimetre level of accuracy. Furthermore, GNSS signals are highly susceptible to the effect of multipath, making GNSS impractical in providing accurate and reliable positioning in urban environments. These two shortcomings mean that standalone GNSS fails to meet two fundamental requirements of critical ITS applications: accuracy and availability. To counteract this problem, the concept of C-ITS is often further exploited. In addition to the sharing of absolute positions and safety related information, participating vehicles within VANETs can also share information that can help them solve for and potentially improve their positions in a collective manner. This is commonly termed as cooperative positioning (CP).

1.1.2 Cooperative Positioning

According to [7], in principle, the concept of CP is not entirely new. For example, the surveying community has long applied this concept on a network of GNSS receivers to provide better positioning output, also known as differential GNSS (DGNSS) [54]. This is sometimes called conventional CP. Using the same concept, vehicles can share information such as GNSS related information, inter-vehicle ranges, and acceleration to collectively solve for and enhance their positioning solutions. Due to the incorporation of various sensors and estimation algorithms, it is appropriate to refer to CP for ITS as modern CP.

The potential of CP to help mitigate the shortcomings of GNSS for ITS has been widely investigated, such as in [29, 86, 90]. However, most of these studies have tested CP in open sky and semi-urban environments, as their main aim has been to improve positioning accuracy. In contrast, few studies have considered deploying and tested CP in dense urban environments, which naturally poses new and more difficult positioning problems. For example, GNSS might not be viable in most parts of urban environments due to its susceptibility to multipath, and even if it is still able to provide positioning solutions, the error could be up to hundreds of metres. When this is the case, even if inter-ranges are available to form CP, conventional algorithms such as the Kalman filter usually cannot produce accurate solutions due to the high uncertainty of the GNSS positions. Furthermore, urban environments often hamper the connectivity between vehicles
due to physical obstructions, thus reducing CP availability. These examples demonstrate that in urban environments, like standalone GNSS, CP alone might not be able to meet the stringent requirements of critical ITS applications. Hence, this study proposes the use of a multi-sensor system (MSS) and enhanced estimation algorithms to help improve CP in difficult GNSS environments.

1.1.3 Multi Sensor System

In this study, MSS is a self-contained navigation system that comprises a set of inertial sensors, known as an inertial measurement unit (IMU); a speed sensor from OBDII; and a navigation processor. Using the IMU, its measurements are integrated over time to provide continuous positioning and attitude solutions. Similarly, the speed sensor allows for the estimation of position from the measured velocity. Due to their self-contained operation, it is not susceptible to external jamming and is able to operate in all environments, thus providing uninterrupted positioning and attitude solutions [62]. However, due to its nature of operation, the system does have a major disadvantage: parallel to its measurements integration, the error also accumulates over time, which in turn causes positional drifts. These drifts, if unbounded by an aiding sensor, result in significant positional error. Therefore, in contrast to GNSS and CP, MSS is only deemed to be reliable in the short term. The amount of error is completely dependent on the quality of the used inertial sensors. The conventional navigation-grade gyroscope, which is mainly based on mechanical gyroscopes, typically has bias instability ranging from $10^5$ to $10^3 \degree/h$, whereas typical navigation and tactical-grade accelerometers based on mechanical and precision resonator accelerometers have a bias stability ranging from 0.1mg to 1g. Although they are highly accurate, these inertial sensors are not suitable for use in most day-to-day commercial applications, primarily due to their cost, size, and availability [118]. In this study, only low-cost sensors are considered to be part of the MSS due to their practicality for VANET.
1.2 Research Aim, Hypothesis and Objectives

Limitations of current approaches to fulfilling the most stringent ITS positioning requirements remain a challenge, particularly in dense urban environments. Therefore, it is the aim of this research to improve CP for VANET in challenging environments by incorporating MSS. In line with the aim, the proposed hypothesis is:

*The development of MSS and enhanced estimation algorithms in CP for VANET enables a continuous and accurate (lane-level) positioning in challenging GNSS environments.*

To realise the research aim and hypothesis, the following objectives are formulated:

1. To investigate the capabilities of existing technologies, in the context of CP, to meet the requirements of ITS. This also includes the investigation of the network design, i.e., the effects of RSU placement, VANET distribution, and geometry on the achievable accuracy.

2. To investigate existing WSN estimation algorithms with regard to their viability for CP. Limitations of these algorithms must be addressed and improvements, if required, must be developed to satisfy ITS positioning requirements.

3. To investigate and improve existing MSS implementation. In turn, this is used as part of CP to strengthen its performance in terms of accuracy, reliability, and continuity.

4. To test the improvements of CP when incorporating MSS in varying conditions, from open sky conditions to a more challenging environment such as a dense urban area. Due to the limitations of resources allocated for this study in terms of manpower and equipment availability, it was not always possible to conduct large-scale real world experiments. Therefore, realistic simulations are used to test the developed CP improvements.
1.3 Research Contributions

The contributions of this thesis reflect the materialisation of its aforementioned objectives. These contributions are outlined below.

1. This thesis includes the in-depth study of the CP network design in open sky, semi-urban, and dense urban environments. In particular, the effects of vehicle distribution, geometry, distance/hops from anchors, and dynamics are studied using the Cramer Rao bound (CRB). The output of this simulated study is then used as a guide to describe the expected achievable positioning accuracy for any particular CP network.

2. The study presents the implementation of an algorithm that has not been tested for CP, called the measurement directed progressive correction particle filter (MDPC). This estimation algorithm is particularly useful when dealing with CP with large initial (prior) positioning errors, which typically occur in urban environments due to corrupted GNSS signals. The algorithms effectiveness is compared to conventional centralised algorithms such as the Kalman filter and particle filter. It is shown that unlike MDPC, these conventional algorithms are not able to provide accurate positioning solutions due to the large prior errors.

3. Chapter 3 shows that the current distributed algorithm, sum product algorithm over wireless network (SPAWN), has limitations in providing accurate positioning solutions in CP. This is mainly due to its inherent nature of not being able to capture the posterior of a given vehicular network, but only the marginal distribution for each participating vehicle. In other words, SPAWN is unable to capture the whole distribution of a vehicular network, which results in a non-optimal positioning estimation. To circumvent this problem, two novel algorithms are introduced and examined in this study:

   i. The first algorithm is a partially distributed algorithm (FSPAWN) where vehicles are federated, thereby creating sets of grouped vehicles. This allows for the
calculation of the posterior distribution for each group of vehicles, resulting in a more accurate positioning solution compared to SPAWN.

ii. The second algorithm is an adaptation of the MDPC in a distributed manner, where the measurements are used to direct the particles, using the Kalman update, before their probability is evaluated. The results show that similar to the FSPAWN, this algorithm is able to provide better positioning estimation, closer to that of a centralised algorithm.

4. The conventional (EKF, UKF) and proposed algorithms (MDPC, FSPAWN, MDSPAWN) are comprehensively tested using both simulation and real datasets. The simulation is set-up to be as realistic as possible, reflecting typical open sky, semi-urban, and urban environments.

5. The study proposes more realistic inertial stochastic parameters using dynamic Allan variance (DAV), which allows the estimation process to be improved for MSS. Unlike conventional studies, which rely on the standard Allan Variance (AV) to extract stochastic parameters, this study shows using experimental datasets that these parameters are not valid when a vehicle is in motion. Using DAV, it is possible to obtain a more accurate characteristic of the inertial sensors, hence providing a more accurate integrated solution.

6. Chapter 2 shows that one of the major contributions to GNSS errors in urban environments is multipath. Hence, this study takes the initiative to mitigate this problem by improving the receiver autonomous integrity method (RAIM). The following two methods are proposed.

   i. Cooperative positioning: It is well known that any additional aiding measurement would provide redundancy, and could thereby potentially improve RAIM. This study investigates how this can be realised in the scope of CP, where vehicular inter-ranges are used to provide redundancy and subsequently improve RAIM in harsh GNSS environments.

   ii. Non-holonomic constraint: This constraint has been widely used for low-cost
MSS, particularly when GNSS observability is low. This study investigates the application of this method to further improve RAIM, i.e. to correctly detect and reject faulty GNSS signals due to multipath in low GNSS observability conditions. The application of the non-holonomic constraint can not only provide a better positioning estimate, but naturally also limits the growth of the navigation solutions level of uncertainty. This is an important improvement as RAIM correctness is dependent on two conditions: MSS estimated position and level of uncertainty.

7. The study develops and tests a tightly integrated V2V MSS/CP based on differenced pseudoranges (non-radio ranging); this is discussed in chapter 5. The inclusion of MSS in CP is shown to provide marked improvements over standalone CP during GNSS outage. Furthermore, due to the presence of MSS, this CP method could also work when only two common visible satellites are present.

8. The study develops and tests a tightly integrated V2X MSS/CP based on UWB ranges in semi-urban and dense urban environments. This is elaborated in detail in chapter 6. As with the previous study, all of the algorithms are tested and compared. However, more emphasis is placed on the federated and decentralised algorithms due to their inherent capability of handling scalability of larger networks, as opposed to the centralised algorithms. Wherever possible, the proposed methods are tested using real datasets. Otherwise, realistic simulations of VANET, generated by VEINS, are used.

1.4 Thesis Outline

This dissertation consists of seven chapters, outlined below.

1. Introduction: The first chapter states the problem of dealing with positioning for vehicular navigation in general and the necessity to overcome this problem to fulfil ITS requirements. The objectives are detailed and research contributions are summarised in this chapter.
2. Intelligent Transportation System: This chapter presents the background and reviews the literature that is relevant to ITS. First, the concept of ITS and recent developments are discussed. Then, the navigation components, namely GNSS and vehicular sensors, and their error sources are discussed. A case study of GNSS performance in urban environments is also presented. The final part of this chapter reviews wireless communication and ranging for CP, discussing the viability of implementing various wireless sensor network (WSN) ranging measurements.

3. Cooperative Positioning Algorithms: Chapter 3 starts with the investigation of CP performance expectations in various environments. The standard CRB is explained first, followed by an alternative CRB when dealing with singular FIM. Then, the chapter reviews current CP algorithms, both centralised and decentralised namely, weighted least squares (WLS), the extended Kalman filter, the Unscented Kalman filter and SPAWN. To overcome the limitations of these conventional algorithms, two novel algorithms are developed and discussed here. The final part of this chapter presents a thorough examination of these algorithms using simulated and real datasets.

4. Improving MSS: Chapter 4 focuses on improving the current implementation of MSS. The first part of the chapter elaborates on the improvement of inertial sensor stochastic coefficient extraction using DAV, while the second half details the improvement of satellite rejection in high-multipath areas by implementing the non-holonomic constraint and adding a speed sensor as part of the MSS.

5. V2X Implementation Using GNSS, MSS, DSRC, and UWB: The fifth chapter discusses the implementation of V2X using MSS, DSRC, and UWB. The V2X implementation takes into account all the developed improvements for CP and MSS, as discussed in earlier chapters. This chapter comprises two parts: the first focuses on V2V implementation using GNSS, MSS, and DSRC only; while the second discusses the same implementation but with an added sensor, UWB. It highlights how MSS and developed estimation algorithms could improve CP in GNSS-denied environments.
6. V2X Implementation in Large-Scale Datasets: The analysis in chapter 5 considers only a small-scale dataset; this is due to the logistical challenges faced in this research when trying to collect large-scale real datasets. Chapter 6 overcomes this issue by simulating realistic VANET datasets, which are then analysed using the same methodology as the previous chapter. This chapter highlights how MSS and developed estimation algorithms benefit CP, where the proposed system can not only heighten the positioning accuracy in GNSS-difficult environments, but it can also reduce the traffic of wireless communication due to the redundant measurements provided by MSS.

7. Conclusion and Further Research: The final chapter summarises the dissertation and highlights the major contributions of this research. Finally, the chapter proposes further research opportunities to improve CP.

1.5 Publications

At the time of this thesis submission, the author has produced a total of eight publications and is the main author of five of them, including two journals, one peer reviewed conference paper, and two non-peer reviewed conference papers.


Chapter 2

Intelligent Transportation System

2.1 Introduction

This chapter presents the literature review and background of this research, and discusses the concept of CP and its components to realise C-ITS. The chapter presents the navigation components utilised in this study for C-ITS: GNSS, IMU, VSS, and the wireless communication system. To maintain brevity, only highly relevant information is discussed for each component, due to the vast availability of information available in the existing literature.

As depicted in figure 2.1, two distinct subsets of navigation systems are introduced and used throughout this study. The first is called local-level observations, where each vehicle is able to provide its own navigation output, independent of other vehicles. This system consists of GNSS, IMU, and VSS. The second is network-level observations, where vehicles share information amongst each other to form a network, using DSRC and UWB, to provide a more robust navigation output in GNSS-denied environments. As will be discussed in this chapter, the choice to use DSRC and UWB to form CP between vehicles and infrastructure in this study comes down to their interoperability with each other and GNSS. Their allocated frequency bands do not interfere one another.

This chapter is structured as follows. First, it discusses ITS and its current and future developments. Then, each of the navigation components used in this study to realise C-ITS are covered. GNSS, the most widely used navigation system for road vehicles, is presented first. The result of a case study on GNSS in urban environments, conducted as part of this research, is also discussed. The case study serves to justify the need for
CP in C-ITS, which is also the main objective of this research. The chapter subsequently presents the IMU, and its implementation for vehicular navigation, error sources, and measurement models. Then, the chapter briefly discusses the VSS using the OBDII interface, and, as for the two previous devices, its measurements and error models are described. One of the most important parts of this dissertation is then finally covered: wireless communication for CP. Here, the chapter discusses the viability of DSRC and UWB as a medium for wireless communication, along with ranging methods. Some parts of this chapter were presented at the 2013 Mobile Mapping Technology conference:


The contents from the published paper have been slightly altered to better fit the flow of this chapter.

### 2.2 Intelligent Transportation System

This section discusses ITS in terms of current implementations and developments, and its underlying technologies. As introduced in chapter 1, ITS can be broadly be defined as the use of technology to improve transportation systems. The scope of ITS is broad, involving multiple parties or users, ranging from personal and transit vehicles to management
centres for emission monitoring and traffic control. It also requires varying layers of communication mediums and methods, as depicted in figure 2.2.

According to [33], ITS delivers five key classes of benefits:

i) increasing safety;

ii) improving operational performance, particularly congestion;

iii) enhancing mobility and convenience;

iv) delivering environmental benefits; and

v) boosting productivity and expanding economic growth.

Specific to transportation, key areas such as safety, efficiency, and capacity have been widely developed. However, the introduction of ITS could potentially bring about new dimensions to these key components of transportation. For example, over the last five decades, much emphasis has been placed on vehicle safety, with developments designed to protect passengers in the event of a crash. The implementation of a fully functioning
ITS, on the other hand, could potentially prevent vehicular collision altogether. Furthermore, recent studies such as [33, 119] have shown that ITS could deliver superior cost benefit returns when compared to conventional investments in increasing road capacity. These studies have estimated that the benefit-cost ratio of systems-operations measures, enabled by ITS, is about 9 to 1. This is well above the addition of conventional road capacity, which has a benefit-cost ratio of 2.7 to 1. Moreover, a 2005 study on model ITS deployment in Tucson, Arizona estimated that the average annual benefits to mobility, environment, safety, and other areas combined total about $455 million annually, which is a 6.3 to 1 benefit-cost ratio [78].

Despite the significant potentials of ITS, its deployment in many countries is limited. This is due to the significant challenges in ITS adoption and implementation, ranging from technological development to policy standards and agreement and jurisdictional differences. Many ITSs are subject to network effect and scale challenges, which requires extensive coordination, often at the national level [119]. Moreover, ITS projects often have to compete for funding against conventional transportation projects such as building new highways or infrastructure and road maintenance. Even though these developments are important, particularly in new developing areas, they do not deliver higher long-term returns than ITS [33].

Although the implementation of ITS poses tremendous challenges, several countries have started to use it and successfully implemented it to some extent. Japan, South Korea, Singapore, Europe and U.S. are among the leading countries in ITS implementation.

### 2.2.1 Current Implementation and Future Developments

This subsection briefly discusses current ITS implementation in three leading countries: Japan, South Korea, Singapore, Europe and the U.S. Leading countries are selected as a function of the highest number of citizens benefiting directly from the implementation of ITS.

Enacted in 1994 and subsequently launched in 1996, Japan’s first initiative towards realising ITS was through the implementation of the vehicle information and communication system (VICS). During its infancy stage, the VICS was primarily used to provide
up-to-date traffic information which would then be available for drivers via an on-board telematics unit (OBU). In 2004, Japan launched Smartway, a more advanced ITS succeeding VICS. The system makes use of vehicle location and traffic information to provide more comprehensive traffic awareness on most roadways in Japan. Called a safe driving support, it is able to advise and provide warnings to drivers regarding any hazardous conditions ahead of time. In 2010, the system was fully deployed nationally. By March 2013, cumulative shipments of VICS OBU exceeded 37 million units. Apart from VICS or Smartway, the national electronic toll collection (ETC) system is also widely employed. It uses DSRC to establish communication between toll gates and vehicles. It is estimated that 41 million vehicles had been fitted with this system by 2013. Today, the three mentioned ITS services—up-to-date traffic information, safe driving support, and ETC—have been combined to serve as an all-in-one system, termed the ITS spot services [33,96,110].

South Korea’s first initiative in implementing ITS started in 1997, when the Korean government drew up an ITS master plan and started to implement yearly plans. Then, in 1999, the Transportation System Efficiency Act was introduced, which included articles about ITS and provided legal support to the ITS project. In 2001, the government decided that the establishment of ITS was one of the major agendas of Advanced Green City. Based on the national ITS master plan, South Korea built its ITS infrastructure on a city-by-city basis, establishing four initial ITS model cities that implemented adaptive traffic signal control, real-time traffic information, public transportation management, and speed violation enforcement. Currently, 44 local governments have established and operate ITS, which provides similar services as the initial four ITS model cities. Like Japan’s ETC system, South Korea too has established a national electronic toll system, called T-money. It covers more than 70% of highways and is used by 31% of vehicles (2013). Broadcast of traffic information, incident management, and freeway traffic flow control services are also provided on all expressways. Furthermore, South Korea is actively conducting national ITS research and development, with a main focus on realising a ubiquitous transport system through V2X communications [33,69,75].

Singapore began its full-scale planning and development of ITS in 1997, when its government approved the integrated transport management system (ITMS) project, which
aimed to integrate all ITS, including obtaining real-time travel time information on the surface street system, the interface with car parks, mass transit, bus transport, and the associated interchanges [99]. In 1998, electronic road pricing (ERP) was implemented, which is similar to Japans and South Koreas systems. Then, in 1999, ITMS was replaced by i-transport, which aims to create a more integrated ITS, consisting of traffic information, public transport information, and a multi-modal route advisory system. At the current stage, it is able to provide predictive traffic flow modelling based on the use of historic and real-time data. As part of the i-transport, Singapore has also deployed adaptive computerised traffic signals nationwide and a national parking guidance system. The effectiveness of Singapore ITS has managed to reduce traffic usage during peak hours by 21% in the morning peak and 27% during the evening peak period [33, 99].

Europe started implementing ITS much earlier than the previously discussed countries. In 1974, Germany introduced a system that broadcasts traffic information called the Autofahrer-Rundfunk-Informationssystem (ARI) using frequency modulation (FM) [85]. In 1986, the program for European Traffic with efficiency and unprecedented safety (PROMETHEUS) was established involving governments, companies and universities of 19 countries. The program developed several ITS technologies such as VAMORS where vehicles equipped with the technology is able to perform road and object detection [124]. The European government also initiated Dedicated Road Infrastructure for Vehicle Safety in Europe (DRIVE) in 1988 where it serves as a platform for research, development and assessment of a range of Road Traffic Informatics (RTI) technologies, the evaluation of strategic choices of candidate systems, and a significant amount of standardisation work [35, 73].

As discussed in previous paragraphs, much of current ITS implementations have been focused on the V2I type of ITS. Over the last few years, however, more emphasis has been placed on V2X to create a more fully integrated ITS and subsequently further exploit its benefits. For example, the most recent U.S. DOT ITS strategic plan outlines five major program categories that need to be developed in the near future, two of which are connected vehicles (CV) and automation. CV, as the name implies, refers to establishing communications between vehicles to create VANET, and integrating it in the ITS infras-
2.2 Intelligent Transportation System

The document outlines the need to continually research, develop, and eventually adopt the concept of CV to pursue the requirements set by the National Highway Traffic Safety Administration (NHTSA) for safety applications. The NHTSA announced in 2014 that vehicles need to exchange basic navigation data, such as position, velocity, direction, and relative position, at 10 Hz, to achieve better safety outcomes. This allows for the realisation of the second program, automation, which concerns automated road vehicle systems and related technologies. The overarching aim of this program is to eventually, to some extent, transfer some amount of vehicle control from the driver to the vehicle. This will have a significant impact on driving safety, mobility, energy consumption, and operating efficiency. Some prime examples of automation include limiting vehicle speed at particular road segments and taking over vehicular controls that might potentially result in a crash [13]. Other countries, particularly Japan, South Korea, Singapore, and Australia, have also devised plans that are similar to the U.S. DOT ITS strategic plan, where greater emphasis is placed on V2X to create a sounder ITS [69, 79, 96].

2.2.2 ITS Summary

The review of the existing literature on ITS has shown that an accurate and reliable positioning system is amongst the key components needed to fully realise a functioning ITS. While most studies mention the use of GNSS to provide global positioning and DSRC to relay relative information between vehicles, only few, such as [9], address the limitations of GNSS in providing required positioning accuracy and reliability. Thus, the focus of this study is to address and overcome the limitations of GNSS for use in ITS. As mentioned in chapter 1, the approaches taken in this research include the use of MSS to provide navigation continuation, implementing CP, and improving conventional algorithms for better navigation outcomes in GNSS-denied areas. The next few sections discuss the navigation systems, both local- and network-level measurements, used in this study to achieve its aims, as presented in chapter 1.
2.3 Global Navigation Satellite System

Perhaps the most widely used positioning system for ITS to date, it is appropriate that GNSS is discussed here before the other navigation systems. In this study, GNSS is used to enable vehicular global positioning, which is critical in CP.

2.3.1 Positioning with GNSS

Positioning with GNSS is achieved using the passive ranging technique. This can be realised by multiplying the speed of light by the time difference between the time of transmission from the satellite and the time of signal arrival at the receiver. The process provides a true range between a satellite and a receiver, where it is assumed that the signals times of transmission and arrival are perfectly known. In reality, however, the range is often distorted due to the signals transition time being corrupted by both satellite and receiver clock bias error. The range is also further corrupted by other sources, such as the multipath effect and the change of speed and direction when the signal passes through the atmosphere [44, 54, 62]. The next subsection discusses the various existing GNSS signals. Due to their full FOC, only global positioning system (GPS) and globalnaya navigatsionnaya sputnikovaya sistema (GLONASS) signals are listed. Furthermore, to maintain brevity, only the C/A ranging signal and L1/L2 carrier frequencies for both GPS and GLONASS, which were used in this study, are briefly outlined. More detailed discussions on the positioning systems can be found in [54, 62].

Signals

In its original plan, the GPS transmits two signals: the Course/Acquisition (C/A) and Precise-Y P(Y) code signals. The P(Y) signal is restricted to licensed precise positioning service (PPS) users only, and is therefore not elaborated further. The C/A code, on the other hand, is intended for public use. However, it is less accurate than the P(Y) code, comprising 1,023 chips that repeat every millisecond. Transmitted at 1.023 Mbps, the C/A code is made up of unique sequences of binary states which are generated using a pseudo-random noise (PRN) algorithmic process [62]. This then allows a GNSS receiver,
which knows the unique PRN of each satellite, to correctly identify individual satellites. Also, this allows for the receiver to deduce the signal transition time when its generated and satellite’s PRN correlate. The range between receiver and satellite is then solved using the measured transition time [44, 54, 62]. Similarly, GLONASS too has C/A and P code signals, with code frequencies of 0.511 and 5.11 MHz, respectively.

GPS signals are broadcasted or modulated onto three frequencies. Firstly, C/A code is modulated onto the L1 carrier, which has a frequency of 1575.42 MHz, using the Bi-Phase Shift Key (BPSK) technique. In addition to an L1 carrier, high-grade GPS receivers can also utilize an L2 carrier, which has a frequency of 1227.60 MHz. These two frequencies allow high-grade GPS receivers to quickly fix integer ambiguity to ultimately calculate a high-precision receiver-satellite range [44]. Furthermore, this retains high-precision ranging over long baselines, typically more than 20km [62]. In addition to signals, GPS satellites also broadcast navigational messages, which are superimposed on the C/A and P(Y) codes. The messages contain valuable information regarding satellite clock correction parameters and status/health, ionospheric model parameters, and broadcast ephemeris [26].

Measurements

GNSS measurements or observables are measured receiver-satellite ranges deduced from measured time or carrier phase differences by comparing receiver generated and satellite transmitted signals. As mentioned earlier, GNSS signal transition time is corrupted by both satellite and receiver clock biases, hence the term pseudorange. In total, GNSS receivers can output three types of measurements: code pseudoranges, phase pseudoranges, and Doppler measurements. These measurements are discussed in the following subsections.

Code pseudorange  The code pseudorange uses the C/A code to obtain signal transition time between receiver and satellite. As mentioned in section 2.3.1, the transition time is equivalent to the time taken for PRN generated by the receiver to align with PRN received from the satellite. Using C/A code-based signals to solve the position of a point
Intelligent Transportation System

at a particular epoch requires four simultaneously measured pseudoranges. This is to solve four unknowns: the three components of position and the receiver clock bias. From a geometrical perspective, the position is obtained by a sphere being tangent to the four spheres defined by the pseudoranges [54]. When GLONASS is used, the GPS-GLONASS time scale error term is added. Taking into account atmospheric, ephemeris, and multipath errors, the code pseudorange can be formulated as the following equation [43]:

\[ \tilde{\rho} = \rho(x, p) + E + c\delta t + c\delta t_{sys} - c\delta T + I + T + M + v \]  

(2.1)

where

- \( \tilde{\rho} \) Pseudo-range
- \( \rho(x, p) \) True-range
- \( E \) Ephemeris error
- \( c\delta T \) Satellite clock error
- \( c\delta t \) Receiver clock error
- \( c\delta t_{sys} \) GPS-GLONASS time scale error
- \( I \) Ionospheric error
- \( T \) Tropospheric error
- \( M \) Multipath error
- \( v \) Noise

**Phase pseudorange** Unlike the code pseudorange, which uses ranging codes to acquire satellite ranging, the carrier phase measurement uses the carrier frequency to calculate high-accuracy and -precision positioning solutions [100]. This is possible because carrier tracking is less noisy and suffers from smaller multipath errors than code tracking [44]. The high-accuracy positioning using the carrier phase is achieved with the relative positioning technique, which requires two receivers to simultaneously observe carrier phases to process a baseline vector. When real-time data transfer between the two receivers is possible, this allows for real-time computation of baseline vectors. This is known as the real-time kinematic (RTK) technique, with which cm positioning accuracy is achievable [54].

Although using carrier phase observables can produce high-accuracy positioning solutions, this process is far more complex than code-based positioning. For instance, it requires integer ambiguity resolution, which, when the number of visible satellites is low, can take several minutes to obtain the initial fix. Furthermore, it requires at least
five satellites to solve the integer ambiguity [54, 100]. Moreover, due to its use of the relative positioning technique, it requires a base station with accurately known coordinates [54]. One of the ways to obtain faster ambiguity resolution is through dual-frequency differencing. Several other methods are extensively discussed in [44, 54].

**Doppler measurement** The Doppler shift measurement is the rate of change of the carrier phase. It is obtained from the carrier tracking loop. The measurement can be transformed to a pseudorange-rate measurement when scaled by its carrier frequency (L1 for example). Subsequently, this can be used to calculate the receiver’s velocity [43]. The Doppler measurement is not used in this study, and is therefore not elaborated further.

This subsection has briefly presented the positioning technique using GNSS, outlining its signals and the different types of measurements utilised in this study. The next subsection briefly discusses GNSS errors and methods to mitigate them.

### 2.3.2 User Equivalent Ranging Error

GNSS user equivalent ranging error (UERE) is the combined effects of orbit, atmospheric, and clock errors and receiver noise projected onto a line between receiver and satellite.

**Ephemeris**

Ephemeris error $E^i$ or satellite orbital error is the error in the prediction of the satellite position. It has three components, cross-track, radial, and along-track errors, all of which are defined in its orbital frame cylindrical coordinates. The control segment monitors the satellite orbits and calculates the ephemeris parameters broadcast to the user by the satellites. The model is a curve fit to the measured orbit, which allows the receiver to compute the satellites position such that the distance between the true and calculated position is less than 10 meters in magnitude. Typically, ephemeris error contributes about 2 metres of a users position error [44, 54].
Clock

Satellite clock error $c\delta t_i$ is due to the cumulative effect of oscillator noise. It is mostly corrected using three calibration parameters broadcasted in the navigation data message. The control segment monitors and computes these parameters by fitting a polynomial correction to the satellite clock error. Thus, the residual clock error depends on the control segments monitor networks size. Furthermore, the latency of the corrections and the stability of the clock itself also affect the residual clock error. Satellite clock residuals typically contribute to about 2-3m of the receivers positioning error. However, because this error is independent of the receivers location, it can effectively be removed by employing the differencing technique [54]. On the other hand, the receiver clock error $c\delta T_i$ is a time-varying bias that affects all simultaneous range measurements in the same fashion. Therefore, if at least four simultaneous satellites range measurements are available, the clock bias and position can both be estimated [54].

Atmospheric

The errors resulting from the atmospheric effect are mostly due to ionospheric $I$ and tropospheric delay $T$. The ionosphere is the layer of the atmosphere that extends from 50 to 1,000 km above the earth's surface. Due to its nature of being a dispersive medium, the propagation velocity of the modulated signal (PRN code and navigation data) is delayed in this layer. However, the carrier phase is advanced at the same rate. This subsequently leads the carrier and modulated code signals to diverge, a phenomenon known as the code carrier divergence [44, 54]. Conversely, the troposphere is the lowest part of the atmosphere, extending from the Earth's surface up to 50 km above it. It is a non-dispersive medium but it affects the index of refraction due to atmospheric pressure, temperature, and humidity. Subsequently, this affects the measured GNSS signals time of propagation with the measured value being larger than the true range [44]. The differential method, used in this research, is usually employed to mitigate both ionosphere and troposphere effects due to their nature of being spatially correlated [34, 54].
2.3 Global Navigation Satellite System

**Multipath**

Multipath errors $M$ occur when the receiver receives reflected signals from a satellite along with the direct signals. This is typically common in a dense urban environment, where signals are reflected from buildings, land, and other artificial structures that have high reflective characteristics. The reflected signals always arrive after the direct path signal due to their longer distances travelled. However, the multipath error can still be both positive and negative: reflected signals can either interfere constructively, resulting in a positive range error, or destructively, resulting in a negative range error [44, 54].

![Deflected signals](image)

**Figure 2.3: Deflected signals (adopted from [34])**

The multipath error can be as large as tens of metres, depending on the environment and antenna design. Some of the considerations in antenna designs are having a low gain at small and negative elevations, and the use of absorbent materials and a choke ring. Moreover, low elevation angle signals are usually subjected to the greatest multipath interference. Therefore, the latter is usually avoided by changing the setting of the receiver to avoid using satellites at low elevations ($< 15^\circ$) [34, 54]. This error is of particular interest in this study as its effect in urban environments is rampant and usually dominates the positioning error. Chapters 4, 5, and 6 present methods to reduce this error in this study.

**Receiver Noise**

$\nu$ - In addition to the ranging errors listed in previous sections, some errors arise solely from the receiver itself. Thermal fluctuations, extraneous RF signals and noise, cross-correlation between CDMA codes and signal quantisation, and sampling effects all are
referred to as receiver noise. Since receiver noise is receiver specific, it is not possible to mitigate it using the differencing technique [43]. Therefore, like in any other noise processes, receiver noise is modelled as white and independent between both satellites and channels [44].

The total GNSS UERE is summarised in table 2.1. The expected accuracy of the point positioning solution can then be calculated by multiplying the UERE by the dilution of precision (DOP) value. The DOP is dependent on the geometrical arrangement of the GNSS satellites [44].

<table>
<thead>
<tr>
<th>Error Source</th>
<th>GNSS-SPP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Satellite Ephemeris</td>
<td>2.1</td>
</tr>
<tr>
<td>Satellite Clock</td>
<td>2.1</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>4.0</td>
</tr>
<tr>
<td>Troposphere</td>
<td>0.7</td>
</tr>
<tr>
<td>Multipath</td>
<td>1.4</td>
</tr>
<tr>
<td>Receiver Noise</td>
<td>3.0</td>
</tr>
</tbody>
</table>

Table 2.1: GNSS UERE Summary [26]

2.3.3 Navigation in Urban and Semi-Urban Environments - A Case Study

The purpose of this case study is to obtain statistics of GNSS behaviour in terms of positioning error and satellite availability in urban and semi-urban environments. These statistics are then primarily used as an argument for this research, and as the basis for creating a model for the urban environment GNSS simulation, used throughout in chapters 3 to 6. Two main findings are expected from this case study:

1. The characterisation of positioning error in semi-dense and dense urban environments. Due to the lack of adequate equipment needed for this test, the positioning errors were only visually approximated for the dense urban test.

2. The characterisation of satellite availability at a particular time in urban and semi-urban environments. The experiment was repeated for five days to observe whether
the number of satellites visible could be replicated on other days (taking into account the sidereal day).

Methodology

Using a Leica geodetic-grade GS10 receiver, six datasets were collected for this study on six different days in Melbourne, Australia. To compare GNSS availability in different environments, the experiment was segmented into two sets. For the first segment, the vehicle drove primarily within the central business district (CBD) area, which is a dense urban environment. As shown in figure 2.4, the starting point for this segment was at the corner of Exhibition and Collins Street. Then, the test vehicle went through Collins Street, and stopped at the corner of Williams and Collins Street. Subsequently, the vehicle proceeded to William Street, Bourke Street, and Elizabeth Street before ending back at the corner of Exhibition and Collins Street. The second segment was conducted along Rathdown Street, which is a semi-urban environment. The start and end points were located close to the corner of Rathdown and Pelham Street. For each stop, the vehicle stayed stationary for at least 3 - 5 minutes, to align the time with the dataset collected the previous day. Furthermore, to account for the difference between sidereal (used by GPS satellites) and solar days, the start/end times were adjusted to start approximately 4 minutes earlier than the previous day.

Results and Discussion

This section is segmented into two parts, according to the expected findings outlined earlier.

Positioning Error in semi and dense urban environments  Figures 2.7 and 2.8 plot the GNSS position (GPS and GLONASS), where the varying colours represent the different datasets used in this test. As reference, the dimensions for each block are 200 by 200 m and 200 by 100 m for segments 1 and 2, respectively. As mentioned earlier, the error present along segment 1 could not be quantified due to the lack of proper equipment
(High grade navigation IMU/GNSS solution). However, figure 2.7 reveals several positioning characteristics worth noting. First, large positioning gaps are repeated across the different datasets. This is simply due to the effect of signal shadowing: the receiver could only receive less than four satellites, and was hence unable to produce a positioning solution. In that same vein, increased availability seems to be concentrated along the first half of Collins Street, along Elizabeth Street, and at all intersections in between building blocks. Clearly, these areas have a better view of the sky, and hence increased satellite visibility. On average, the positioning availability is around 60% for this segment. Second, a number of the computed positions contain huge 3D errors, some in excess of 150 m, most
likely due to the effect of multipath. Specifically, approximately 30% of the positions have errors of more than 50 m, and the largest error is around 160 m.

On the other hand, segment 2 shows significant increased positioning availability as this trajectory mostly has open sky conditions, as shown in figure 2.8. Positioning availability is on average 98%. Errors are still present, but at a much more manageable scale. As seen in figure 2.9, the 3D errors are well within the acceptable range of below 10 m, with the exception of a few points where the errors exceed this value. The errors reflect the undesirable conditions in which the vehicle was travelling: a 16-storey building, some overhead foliage cover, and two- and three-storey buildings were nearby. On average, the percentage of error above 10 m is around 5%, with a maximum error of 70 m. Note
that the y-axis of the error plot has been limited to 25 m to better demonstrate the error values, i.e., maximum error is not seen here. Looking more closely at figure 2.9, the errors are consistent across the six datasets used, indicating a good repeatability of GNSS positioning error. The averaged root mean squared error (RMSE) from all datasets for this segment is 4.015 m.

Satellite availability in semi urban and dense urban environments This subsection presents the satellite availability and repeatability in more detail by observing the num-

Satellite availability in semi urban and dense urban environments This subsection presents the satellite availability and repeatability in more detail by observing the num-

ber of satellites at a particular time. Figures 2.10 and 2.11 show the number of satellites visible throughout segments 1 and 2 respectively. To ease the interpretation of the number of satellites, figures 2.10 and 2.11 show the averaged number of satellites across all datasets instead of plotting them individually in a single graph. In these graphs, the blue dots represent the averaged number of satellites, while the red dots represent the standard deviation of the number of satellites observed at a given time. The standard deviation allows for consistency/repeatability checks between the varying datasets used.

Both figures show expected results: when in dense urban environment, the average number of visible satellites is considerably lower than when in open sky environments. For example, the average number of visible satellites in the dense urban environment (segment 1) and semi-urban environment (segment 2) is 5 and 11, respectively. Similarly, in the urban environment, there are more than three satellites, thus enabling three-dimensional positioning, in only 60% of the cases, whereas in the semi-urban environment this is about 98%. As noted earlier, the number of visible satellites tends to be higher when the vehicle is located near an intersection, whereas it is always minimal be-
between intersections, or halfway down a block of buildings. This can be seen in figure 2.10, where the number of satellites increases as the vehicle approaches an intersection but decreases as the vehicle moves further away from it, hence showing a sinusoidal trend. Compared to segment 2, the repeatability of the observed number of satellites is much more difficult to replicate across the datasets for segment 1. This can be seen when comparing the standard deviation values for segments 1 and 2.

<table>
<thead>
<tr>
<th>Segment 1</th>
<th>%</th>
<th>Segment 2</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma &lt; 0.5$</td>
<td>1</td>
<td>$\sigma &lt; 0.5$</td>
<td>1</td>
</tr>
<tr>
<td>$\sigma &lt; 1.0$</td>
<td>14</td>
<td>$\sigma &lt; 1.0$</td>
<td>44</td>
</tr>
<tr>
<td>$\sigma &lt; 1.5$</td>
<td>19</td>
<td>$\sigma &lt; 1.5$</td>
<td>59</td>
</tr>
<tr>
<td>$\sigma &lt; 2.0$</td>
<td>67</td>
<td>$\sigma &lt; 2.0$</td>
<td>93</td>
</tr>
</tbody>
</table>

Table 2.2: Repeatability percentage of number of visible satellites

Table 2.2 lists the percentage of standard deviations within the threshold of 0.5, 1, 1.5, and 2. This serves as an indication of the capability and reliability of tracking identical GNSS constellations. Evidently, this is much more difficult in an urban environment than in a semi-urban environment. For example, the percentage of standard deviations that are less than 1 is only 14% for segment 1, while for segment 2 this is 44%. This is due to
the sensitivity of the receiver in the harsh environment and the varying traffic conditions across the different days of data collection, which make it very difficult to accurately replicate identical observation conditions at any particular time.

Summary

This simple case study has fully outlined the results of characterising GNSS positioning error and availability in both urban and semi-urban environments using six datasets collected on six different days. As expected, the positioning error can be as large as hundreds of metres in an urban environment, largely due to multipath. Although it could not be properly quantified due to a lack of appropriate equipment, visual inspection reveals that the error is similar to the findings in previous studies [45, 67, 86, 102]. The number of observable satellites in the urban environment is considerably lower than in the semi-urban environment. The results also show that the harsh environment makes data collection with identical observation conditions, or repeatability, difficult. It can be expected that more than three satellites are visible 60% and 98% of the time for urban and semi-urban environments, respectively. It is also important to note that in urban environment, satellite availability increases when the vehicle is in areas where the sky view is wider, i.e. at intersections, but decreases when in between intersections. These characteristics
are important and are used when simulating GNSS positioning in subsequent chapters. Overall, the results of this case study demonstrate the major drawbacks of navigating using GNSS in urban environments in trying to achieve ITS positioning requirements. This highlights the need to employ alternative positioning techniques such as CP and integrated navigation systems to improve positioning in dense urban environments, which is the overarching aim of the present study.

2.4 Inertial Measurement Unit

The second navigation system that is part of the local observation is the IMU. It is a self-contained three-dimensional navigation system that provides information on position, velocity, and attitude (PVA). The use of an IMU to provide three-dimensional positioning solutions was realised in the early 1950s, mainly for the use of seaborne and airborne applications. Since then, the range of applications utilising IMU has become more extensive, including navigation of aircrafts, ships, and spaceships, autonomous image georeferencing, vehicle stabilisation and control, and borehole surveying [19, 62].

This section presents the IMU. It begins by discussing IMU components providing brief insight into the different grades of commercially available IMU. Then, the section proceeds to examine four reference frames and the development of IMU mechanisation equations. Finally, it concludes with IMU errors and methods of compensating for these errors.

2.4.1 Introduction to IMU

IMU Components

Based on the dead-reckoning concept, IMU tracks changes in position, velocity, and orientation in a defined reference frame by utilising a processor and an array of acceleration and turning rate sensors. Its operation is based on Newton's law of motion, where every object in a state of uniform motion tends to remain in that state of motion unless an external force is applied to it [118]. The force, acceleration, and angular velocity are measured using the
2.4 Inertial Measurement Unit

IMU, which consists of three orthogonal accelerometers and gyroscopes.

i. An accelerometer measures acceleration- and gravity- induced reaction forces. It detects the magnitude of acceleration relative to the direction of its axis and generates its changes in vector quantity.

ii. A gyroscope measures angular velocity relative to its axis. This allows the system to update the information on which direction it is facing relative to its initial orientation.

To obtain navigation solutions, the IMU measurements need to be translated from its operating frame (body frame) to a defined reference frame. This is typically earth-centred-earth-fixed (ECEF) and local navigation frames (LNF). In this dissertation, the frame in which PVA are resolved is called the navigation frame. A processor is utilised to continuously track its orientation and perform the rotation of measured accelerations into the navigation frame. Combining the two systems (IMU and processor) together forms an inertial navigation system [118].

Miniaturised IMU

This subsection provides brief insight into MEMS-grade IMU. Due to the large variety of inertial sensor categories developed over recent decades, only common categories are discussed in the following. Table 2.3 lists the approximate performance parameters of several IMU grades, ranging from navigation to consumer grades.

Technological advancement in recent decades has made it possible to produce low-cost and micro-scaled inertial sensors. These are often referred to as micro electro mechanical systems (MEMS) IMU. Using silicon as the base material in its components has helped rapid mass production. This approach radically overcomes many of the issues with manufacturing conventional mechanical inertial sensors, such as the high part count, which is the requirement for many parts with high precision tolerances and complex precision techniques [118]. MEMS inertial technology makes use of the chemical etching and batch processing techniques commonly used in electronic integrated circuits.
Therefore, the resulting solid state sensors, being small in size and low in weight, power consumption, cost, and maintenance provide greater design flexibility compared to earlier technologies, making it suitable for a number of applications in both military and commercial sectors [83, 118].

Nevertheless, the reduction in terms of size and cost of the inertial sensors pose new challenges for providing accurate and high-resolution measurements. Most current MEMS-based inertial sensor measurements are noisier, less sensitive, and more temperature dependent compared to higher-grade inertial sensors. However, the performance of MEMS inertial sensors is anticipated to improve within the next few years. In fact, several companies, including Systron Donner Inertial Division, have managed to produced tactical-grade MEMS inertial sensors, with bias stability of 1°/h and 200μg for gyroscope and accelerometer, respectively [112]. This enhancement is mainly due to its ability to compensate for systematic errors, which is achievable in real time. In addition, the performance can be further improved by using accurate error models, which in turn allows accurate estimation of navigation errors. The raw measurements are then compensated with the estimated errors [118].

It is expected that in the coming years, MEMS-based inertial sensors could achieve similar performance to current optically and mechanically based inertial sensors; thus,
they could be graded as navigation-grade IMU. When integrated with GNSS, these sensors could provide continuous, reliable, and accurate positioning solutions. The advantages of MEMS inertial sensors, being their low cost, small size, and low power consumption, make it an attractive alternative to conventional inertial sensors.

2.4.2 IMU Mechanisation

IMU mechanisation refers to IMU utilising specific kinematic equations with respect to a coordinate frame. Before discussing the kinematic equations, three fundamental reference/coordinate frames are first outlined. Then, IMU mechanisation in the ECEF frame, used in this dissertation, is discussed. This is done only briefly as their algorithms are well established and understood within the inertial navigation community, as can be seen in numerous studies [34, 44, 118].

Coordinate Reference Frames

**Inertial frame** Inertial frame $i$ is the reference frame where Newtons laws of motion apply. It does not accelerate or rotate with respect to fixed stars. Its origin is arbitrary, and the coordinate axis may point in any direction in the universe. This frame serves as the fundamental frame for the inertial sensors operation, where all of its output measurements are produced relative to this frame. For navigational purposes, however, a more suitable frame is defined, called the earth centred inertial (ECI) frame, whose origin is centred at the earths centre of mass and oriented with respect to the earths spin axis and the stars [34, 44, 62].

**ECEF frame** The ECEF frame $e$ is similar to the ECI frame, except that all axes are fixed to the earth. The $z$-axis is always parallel to the earths mean spin axis, while the $x$-axis points from the centre to the intersection of the equator mean meridian of Greenwich. Finally, the $y$-axis is orthogonal to the $X$ and $Z$ axes to complete a right-handed orthogonal coordinate frame. The ECEF is commonly used as both a reference and a resolving frame [101].
**Body frame** The body frame $b$ is fixed to the IMUs body. Its origin is coincident to the origin of local navigation frame (LNF), and the orthogonal axes are fixed to the IMUs body. In this study, the axes are positioned such that the $x$-axis points forward, the $y$-axis points to the right, and finally, the $z$-axis points upwards. Therefore, adopting this method, the $x$-axis becomes the roll axis, the $y$-axis is the pitch axis, and the $z$-axis the heading axis. In a strapdown IMU, the body frame needs to be carefully defined, as it measures the motion of the body frame relative to the ECI frame [34,44,62].

![Figure 2.12: Body Frame](image)

**Mechanisation in the ECEF frame**

Having defined the fundamental coordinate frames, the following details the IMU mechanisation in the ECEF frame. Most of the equations listed herein are adopted from [44]. The ECEF frame is a common reference and resolving frame for GNSS. Therefore, IMU mechanisation in this frame is discussed in the following, as it is used to interface IMU with GNSS.

**Attitude update** Updating attitude in the ECEF frame uses the body angular rate sensed by the IMU, $\omega_{ib}^b$ to update the attitude solution, expressed in the ECEF frame. In addition, the constant Earth rotation rate, $\omega_{ie}^e$ is also accounted for. The value of $\omega_{ie}^e$ is listed
2.4 Inertial Measurement Unit

in appendix B.

\[
\dot{C}_b^e = C_b^e \Omega_b^b \\
= C_b^e \Omega_b^b - \Omega_e^e C_b^e
\]  

(2.2)

(2.3)

where \(\Omega_e^e\) and \(\Omega_b^b\) denote the skew symmetric matrices of the earth rotation and the turning rate in body frame vectors respectively. In this thesis, \(\Omega\) denotes a skew symmetric matrix of a vector and \(C\) denotes the rotational matrix. When integrating equation 2.3, the exponential component is truncated to the first order, assuming the change angular velocity is small and constant over the integration time. Therefore, integration in discrete time is given by

\[
C_b^e(+) = C_b^e(-) \left( I_3 + \Omega_{ib}^b \tau_i \right) - \Omega_e^e C_b^e(-) \tau_i
\]  

(2.4)

Velocity update The measured acceleration is first transformed into the ECEF frame using the rotation matrix obtained using equation 2.4

\[
f_{ib}^e = \hat{C}^e_b f_{ib}^b
\]  

(2.5)

Then, velocity in discrete time can be updated using

\[
\nu_{ob}^e(+) = \nu_{ob}^e(-) + \left[ f_{ib}^e + g^e_b(-) - 2\Omega_e^e(\nu_{ob}^e(-)) \right] \tau_i
\]  

(2.6)

where the gravity, \(g\) is the sum of the gravitational and centrifugal accelerations, transformed from the LNF to ECEF frame using \(C^e_n\).

Position update Accounting for the gravity and Earths rotation, the position update in the ECEF frame is straightforward, as shown below:

\[
r_{ob}^e(+) = r_{ob}^e(-) + \nu_{ob}^e(-) \tau_i + \left( f_{ib}^e + g^e_b(-) - 2\Omega_e^e(\nu_{ob}^e(-)) \right) \frac{\tau_i^2}{2}
\]  

(2.7)
2.4.3 Initialisation and alignment techniques

Since the IMU is based on the dead-reckoning concept, using measurements to solve its change in position over time, it needs to be initialised with a starting point. This process is called initialisation. Both position and velocity may be initialised using external information, such as GNSS or terrestrial radio navigation equipment. An alternative method of initialising position is to use the last known position from the previous operation, given that the position has not changed. It is important to note that the lever arm effect should be taken into consideration for high-accuracy applications, such as aircraft navigation. However, if the external informations (GNSS antenna for example) distance to the IMU is small, the lever arm effect is not significant, and may therefore be neglected [14, 44].

Initialisation of attitude is called attitude alignment. Similar to position initialisation, it may be initialised using the last known attitude from the previous operation, given that the orientation has not changed since then. However, when this is not the case, the IMU self-alignment procedure is carried out. This is only possible when using high-grade IMU; therefore, it is not discussed here, but can be found in [44, 47].

2.4.4 Error Sources and Models

The fundamentals of IMU operation were discussed in the previous sections. The present subsection examines IMU errors, which can be categorised into two groups: deterministic and stochastic errors. Deterministic errors include static bias, scale factor, and non-linearity. These errors can be determined by lab calibration procedures and the parameters are often stored in IMUs processor to correct its raw measurements. Unlike deterministic errors, on the other hand, stochastic errors cannot be corrected by the processor, as they are random in nature. These errors can only be estimated and corrected through integration with an external navigation system [118]. Common stochastic processes include random noise and in-run bias stability, and are discussed further in chapter 4. The error models of IMU are summarised below for the accelerometer and gyroscope, respec-
2.4 Inertial Measurement Unit

\[ f_{ib}^h = b_f + (I_3 + M_a)f_{ib}^h + v_a \] (2.8)
\[ \tilde{\omega}_{ib}^h = b_g + (I_3 + M_g)\omega_{ib}^h + G_g f_{ib}^h + v_g \] (2.9)

where:
- \( f_{ib}^h \) IMU-output acceleration
- \( f_{ib}^h \) IMU-true acceleration
- \( \tilde{\omega}_{ib}^h \) IMU-output angular velocity
- \( \omega_{ib}^h \) IMU-true angular velocity
- \( b_f, b_g \) Bias
- \( I_3 \) Identity matrix
- \( M_{f,g} \) Scale factor and misalignment
- \( G_g \) g-dependent bias
- \( v \) Noise

The subscript \( ib \) denotes the inertial frame while the superscript \( b \) denotes the body frame. The following subsection details each of the IMU error components as listed in the equations above except for cross coupling, as its effect is negligible in the sensors used in this study.

Bias

Sensor bias is the most dominant type of error in all inertial sensors [44,118]. It is defined as a constant error or offset from true measurement throughout an operation. It is also referred to as g-independent bias to distinguish it from g-dependent bias. Bias can be further separated into two components: static and dynamic biases.

\[ b_f = b_f + bd_f \] (2.10)
\[ b_g = b_g + bd_g \] (2.11)

where:
- \( b_{f,g} \) Static bias
- \( bd_{f,g} \) Dynamic bias

The static bias is also referred to as fixed bias, turn on bias, or bias repeatability; it varies from one turn on to another and stays constant from the beginning and throughout an
operation. The static bias is deterministic in nature [44,118]. Several methods such as the
local-level frame, six position static, and multi-position calibration methods have been
developed to determine static bias, as discussed in [111]. In this study, the bias is esti-
imated using the multi-position static method due to limited access to machinery that
could help determine it more accurately.

Unlike the static bias, the dynamic bias varies over time during operation. The temperature-
dependent bias is usually incorporated into this bias. It is also referred to as bias variation
or instability, and normally accounts for 10% of the static bias. Due to the random nature
of the dynamic bias, it is often desirable for it to be estimated in the integration algorithm
[44].

**Scale Factor**

The scale factor error results from the departure of the instruments input-output gradient
from unity after unit conversion. Accelerometer output error due to the scale factor error
is proportional to the true specific force along the sensitive axis. Similarly, gyro output
error arising from the scale factor error is proportional to the true angular velocity about
its sensitive axis. The scale factor error is deterministic in nature, and it is therefore possi-
ble to determine it through lab calibration procedures. However, it is also possible, and
often desirable, for it to be estimated in the integration algorithm, given that appropriate
stochastic modelling of the error is applied [43,44]. In this study, it is included as part of
the estimation process. The scale factor can be represented as the matrices below for both
the accelerometer and gyroscope, respectively.

\[
M_f = \begin{bmatrix}
    s_{f,x} & 0 & 0 \\
    0 & s_{f,y} & 0 \\
    0 & 0 & s_{f,z}
\end{bmatrix} \quad M_g = \begin{bmatrix}
    s_{g,x} & 0 & 0 \\
    0 & s_{g,y} & 0 \\
    0 & 0 & s_{g,z}
\end{bmatrix}
\] (2.12)

where:

\begin{align*}
    s & \text{ Scale factor}
\end{align*}
Random Noise

Random noise or stochastic error has a number of causes. One of these is electrical noise, which limits the resolution of inertial sensors. It is also caused by mechanical instabilities and vibrations, such as in pendulous accelerometers and spinning mass gyros. Random noise, when integrated, causes velocity and angle random walk [44]. Stochastic in nature, it is not possible to directly determine it using any of the static test methods previously mentioned. However, its parameters can be estimated using PSD and AV analyses. This study further enhances the parameter estimation process, as described in chapter 4. These parameters are then used in the integration algorithm to effectively minimise its effect.

2.5 Vehicular Speed Sensor

The third and final component of the local observation system within vehicles is the VSS. As the name suggests, VSS is used to measure the speed of a vehicle. It is a counting sensor which determines the speed by counting the revolutions of the transmission or transaxle output shaft over a specific period of time, typically tracked by the engine management computer. The computer then compares the signal from VSS against its internal clock to determine vehicle speed. In general, there are two types of VSS: the optical and magnetic sensor. The optical VSS was usually fitted in vehicles manufactured in the 1980s, while newer vehicles are most likely fitted with the magnetic VSS [71].

The optical VSS makes use of a photo cell, a light-emitting diode (LED), and a two-blade mirrored reflector to generate an electrical signal. This electrical signal, which produces pulsating current, is then converted into the number of photo cells to an electronic measurement of the vehicle speed. The magnetic VSS, on the other hand, is directly mounted on the transaxle case in the speedometer. The magnet rotates past a sensing coil which then generates a pulsating voltage. This voltage is then converted into a digital output that can be read by the vehicles computer.

In modern vehicles, VSS is used in many sub-systems. For example, it is used to regulate power steering pressures at low speeds, to ease parking manoeuvres. It is also commonly used for the anti-lock brake system (ABS) to prevent wheels from locking up.
when drivers apply sudden and hard braking and maintain directional stability. In this study, VSS is used as a velocity measurement as part of the MSS.

### 2.5.1 Measurement Characteristics

This subsection characterises the VSS. An experiment lasting over 10 minutes was conducted to test the VSS obtained from the test vehicle. In this test, the VSS measurement was compared to the velocity derived from dual-frequency post-processed GNSS solutions, here treated as a reference. Figure 2.13 shows the velocities from the VSS, marked as a red line, and the GNSS, marked as a green line. Comparison analysis shows that the velocities derived from these two systems differ, which can clearly be seen in the subplot of figure 2.13. This subplot shows the enlarged plot of seconds between 430 and 595, which reveals that velocity obtained from VSS contains a quantisation error, with a resolution of 0.28 m/s (1 km/h).

![Figure 2.13: VSS vs GNSS velocity](image)

Figure 2.13 shows the error plot and its corresponding histogram of the velocity obtained from the VSS compared to the reference velocity. From the error plot, it can be seen that during stationary moments, there are no discrepancies between the speed sensor and reference velocities. Conversely, error starts to appear when the vehicle is in motion, with a mean and standard deviation of about 0.15 m/s. This can also be seen in the histogram in figure 2.14, where the bin with the highest frequency is the error ranging from 0 to 0.2
m/s.

The VSS error characteristic is as expected, and it is largely due to the speed sensors quantisation error, as the measured speed is limited to a resolution of 0.28 m/s. In this study, the error is modelled as white noise. However, the measurements need to be readjusted before they can be used in the MSS. The measurements are adjusted such that, when in motion, they are added with half of the quantisation error value (an offset), resulting in $z = z + 0.14$ m/s. The resulting error characteristic is shown in figure 2.15, where the error is centred around 0 with unchanged standard deviation.
It should be noted, however, that this approach assumes that the actual velocity is randomly distributed in between the lower and upper bounds of the measured speed. The lower bound is defined as the measured speed, while the upper bound is the next quantisation step (+0.28 m/s). This assumption is not universally valid; for example, it does not hold when the actual velocity is concentrated in the first quarter of the bound. Subsequently, the measurements are no longer normally distributed along the assumed statistics. To alleviate this problem, this work employs a rather crude approach by slightly increasing the variance of the measurements. The results of several tests, such as the one shown here, show that this assumption does hold due to the small quantisation error when compared to the variation of the actual speed when navigating in a vehicle. A more proper approach to dealing with state estimation with quantisation error can be found in [28]. From here onwards, the study refers to the IMU and VSS as MSS.

2.6 Wireless Communication for Cooperative Positioning

This section discusses the network observation part of this study, reviewing the components to realise wireless communication and, subsequently, CP for VANET. In terms of hardware, both DSRC and UWB were used in this study; hence, they are discussed accordingly. The section also reviews the literature on deriving meaningful measurements, both radio and non-radio ranging, for CP using these wireless communication modules, and questions their viability. Finally, ad-hoc networking issues when dealing with inter-vehicle ranging are discussed.

2.6.1 Dedicated Short Range Communication

DSRC is a wireless communication channel designed specifically for vehicle-vehicle and vehicle-infrastructure communications. In the U.S., the Federal Communication Commission (FCC) has allocated DSRC a dedicated bandwidth of 75 MHz in the 5.8505.925 GHz band. The DSRC has six service channels (SCH) and one control channel (CCH). As seen in table 2.16, each channel is allocated 10 MHz with the option of having two channels merged, thereby creating two 20 MHz channels (channels 175 and 181). In the
PHY layer, the CCH has the highest priority and is used for “public safety applications involving safety of life and property” [5, 59, 72]. It is important to note that the allocation of DSRC frequency bands differ in other regions; for example, Japan has allocated 5.770-5.850 GHz while Europe has allocated 5.875-5.905 GHz bands. In this study, the FCC standard was used.

<table>
<thead>
<tr>
<th>Frequency (MHz)</th>
<th>5850</th>
<th>5855</th>
<th>5865</th>
<th>5875</th>
<th>5885</th>
<th>5895</th>
<th>5905</th>
<th>5915</th>
<th>5925</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel Number</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Guard Band</td>
<td>172</td>
<td>174</td>
<td>176</td>
<td>178</td>
<td>180</td>
<td>182</td>
<td>184</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 2.16: DSRC frequency allocation [72]

The DSRC is based on the IEEE 802.11p physical layer (PHY) and media access control (MAC) specifications for implementing wireless access in vehicular environments (WAVE). The IEEE 802.11p is an amendment to the IEEE 802.11 standard, which concerns implementing the wireless local area network (WLAN), making it more suitable for WAVE [127]. The amended standard uses the orthogonal frequency division multiplexing (OFDM) technique to transmit DSRC data. The OFDM is a multi-carrier modulation method, where digital data is encoded on multiple carrier frequencies, and thus on different channels or sub-carriers. The OFDM technique is advantageous compared to other techniques, such as the code division multiple access (CDMA) or any other single carrier system, as it is able to adapt to severe channel conditions, narrow band interference, and frequency fading due to multipath [5, 59]. However, it is susceptible to carrier frequency offset (CFO) and Doppler shift due to the higher Doppler spread and multipath delay spread. The modulation of the OFDM sub-carriers is done using varying techniques, including the binary phase shift keying (BPSK), quadrature phase-shift keying (QPSK), and 16 and 64 quadrature amplitude modulation (QAM) methods [72]. The transfer data rates for DSRC vary from 3 to 27 Mb/s, depending on the modulation techniques and coding rates [5].

DSRC can be configured to vary its transmitting power, which is categorised into four classes. The power is measured in effective isotropic radiated power (EIRP) units.
Nominally, the transmit power ranges from 0 to 28.8 dBm. However, for public safety applications, the transmitting power can be increased up to 44.77 dBm [11]. Using the maximum power setting, the communication range can be extended up to 1 km. Each of the channels in DSRC can be configured to transmit at different power levels, and can thus vary according to specific applications [5, 72]. For example, channel 178 or CCH can be configured to have the maximum power of 44.8 dBm as it is usually used for safety applications, while the other channels can have power transmitted at 20 dBm to communicate with other vehicles.

DSRC performance can be separated into several factors, such as bit error rate (BER), effect of packet length, limited link lifetime, Doppler shift, and latency and delivery rate. This is not the focus of this research, and is therefore not discussed in detail. However, a summary from [5] points out several key performance metrics for DSRC. First, the packet length does affect the successful packet delivery rate, and not in a linear function of packet size, where less payload leads to better success rate. Second, due to the varying mobility within VANET, the link lifetime between nodes is considered to be an important issue; it is considerably lower when vehicles are moving in an opposite direction than when they are travelling in the same direction. Finally, the latency and delivery rate for DSRC in varying traffic conditions was investigated. In a simulation for heavy traffic conditions, where each vehicle broadcasts 10 packets/s and 1 packet contains 200 bytes, the packet delivery rate is about 65% and has a latency of 1 ms. These performance factors are crucial and need to be taken into account, particularly when dealing with VANET for safety purposes. The viability of deriving range measurements from DSRC is discussed in subsection 2.6.3.

2.6.2 Ultra-Wide Band

The ultra-wide band (UWB) is a radio technology mostly used in low probability of detection radar and communications systems. In 2002, the U.S. FCC allocated the UWB frequency bands of 3.1-10.6 GHz and 22-29 GHz. Its signal is characterised by its centre frequency ratio being larger than 0.2 and a total bandwidth of more than 500 MHz. The high frequency requirement to operate the UWB ensures that it does not interfere
with other communication devices using lower frequencies, such as GNSS and cellular services. In addition, to further minimise interference with other carrier-based licensed users, the maximum mean radiated power spectral density in this band is limited to -41.3 dBm/MHz [12, 116].

In general, there are two methods of transmitting UWB signal. The first and more commonly used is the impulse UWB. It is also known a direct sequence (DS) UWB, where the UWB transceiver transmits a series of pulses that are short in duration and occupy a very wide band. The second type is the multi-band orthogonal frequency division multiplexing (MB-OFDM) method, where the signals are modulated on carrier frequencies with a fixed frequency of at least 500 MHz. Clearly, both methods require UWB to transmit very wide bandwidth. This leads to a superior performance compared to narrow band technologies such as GNSS in terms of more accurate ranging, ability to deal with multipath, and penetrating structures, thus increasing ranging availability [57,80,91,116].

Most UWB applications include indoor positioning and asset tracking. This is due to the limited range in which UWB signals can travel as a consequence of the low power signal transmission requirements set by the FCC. However, this can be overcome by employing the coherent pulse integration method, as described in [57, 116]. This technique allows for the data and pulse scans to be conveyed using a stream of pulses in a packet structure. In other words, it allows for the data to be spread over multiple pulses. Then, the receiver acquires this data and coherently aligns and tracks the transmission pattern, and finally integrates the multiple sequential pulses to decode the data. Apart from increased range, it can also increase SNR and dynamic range. The improvement comes at the cost of lower update rates. For example, the Time Domain PulsOn P410® UWB module specification, as shown in table 2.4, reveals that higher update rates require longer integration time. As seen in this table, the maximum range of 125 m is achievable when the update rate is set to 46.9 Hz. It is possible to extend the UWB range up to 354 m, but the update rate has to be decreased to 7.5 Hz.
2.6.3 Radio Range/Range-Rate CP Measurements

This subsection discusses the radio-based range/range-rate using DSRC and UWB. Time propagation measurements can be done in two forms. The first technique, the one-way propagation time measurement or time of arrival (TOA), allows for range estimation by measuring the time it takes from where the signal is transmitted to where it is received. This requires accurate time synchronisation of the transmitter and receiver, thus adding complexity to the ranging system. The second technique is the round-trip propagation time or two-way time of flight (TW-TOF), which measures the difference between the time when a signal is transmitted from a node to another and the time when it is sent back to itself. Given that the two nodes use the same clock, there is no need for accurate time synchronisation [5].

The time difference of arrival (TDOA) technique allows for ranging estimation by measuring the time difference between the time when the anchor nodes receive the transmitted signals from non-anchor nodes, also known as multi-lateration. The most widely used technique when implementing TDOA is the generalise cross-correlation method, which consists of integrating the lag product of two received signals for a particular time period. Once the time differences are obtained, one can compute the difference of angles and subsequently use the known baselines between anchor nodes to compute their ranges to the non-anchor node. One of the drawbacks of TDOA is that a severe effect of multipath can cause overlapping cross-correlation peaks, which makes time difference estimation impossible [80].

![Table 2.4: UWB performance using coherent integration technique](image)

<table>
<thead>
<tr>
<th>Pulse-Bit Ratio</th>
<th>Max range (m)</th>
<th>Data rate (bps)</th>
<th>Rate (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>16:1</td>
<td>35</td>
<td>632k</td>
<td>125</td>
</tr>
<tr>
<td>32:1</td>
<td>60</td>
<td>316k</td>
<td>103.1</td>
</tr>
<tr>
<td>64:1</td>
<td>88</td>
<td>158k</td>
<td>73</td>
</tr>
<tr>
<td>128:1</td>
<td>125</td>
<td>79k</td>
<td>46.9</td>
</tr>
<tr>
<td>256:1</td>
<td>177</td>
<td>39.5k</td>
<td>27.2</td>
</tr>
<tr>
<td>512:1</td>
<td>250</td>
<td>19.7k</td>
<td>14.7</td>
</tr>
<tr>
<td>1024:1</td>
<td>354</td>
<td>9.86k</td>
<td>7.5</td>
</tr>
</tbody>
</table>

Table 2.4: UWB performance using coherent integration technique
The received signal strength (RSS), which are readily available from most wireless devices, can be converted into ranges. This approach is attractive in WSN localisation as it requires no additional hardware and as wireless devices like wireless access points are abundant, particularly in urban areas. The first challenge in utilising RSS is mapping the available access points, which is an impractical and laborious task. Secondly, it is difficult to accurately model RSS, as the signals are easily affected by reflection, scattering, and diffraction, which change from one place to another (depending on wall thickness, reflective surfaces present in the area, etc.) [80].

Although the above radio ranging techniques seem promising, their viability using DSRC is questionable. For example, TOA requires complex time synchronisation, which is not readily supported by DSRC in that its base protocol, IEEE 802.11, is only accurate to micro seconds. To achieve acceptable ranging measurements for CP, timing accuracy in the order of nanoseconds is required. On the other hand, although TDOA does not require time synchronisation, it can only be realised using DSRC when two nodes use the same bandwidth. This significantly reduces the capabilities of DSRC, and is therefore not suitable for VANET. RSS by itself is also not suitable for CP in VANET due to its inaccurate ranging, as demonstrated in [5].

Range-rate measurements can be estimated based on Doppler shift or the integrated carrier phase difference between vehicles. Relating to CP, it is less used when compared to range based measurements due to the lower amount of location-related information. The range-rate between vehicles can be calculated if the carrier frequencies of the transmitted and received signals are known. Due to the crystal clocks used in DSRC, the CFO of its received signal, which consists of Doppler shift, is affected by clock drift. Using DSRC in particular, the CFO can be estimated with a resolution of around 100 Hz for 5.9 GHz frequency. One of the advantages when using the CFO-based range-rate is that Doppler shift is not affected by multipath as much as range-based techniques, which makes it more viable for CP. However, it also has a disadvantage in that it is only useful when relative mobility between vehicles is above the level of range-rate noise, which is not usually achievable when they are travelling in the same direction [5].

A more promising candidate for inter-vehicle ranging is the UWB, where TW-TOF
is often used to achieve accurate ranges. Using this technique, cm accuracy can be achieved even over long distances when the coherent pulse integration method is employed. Moreover, [57] have shown that even when UWB is deployed in a high-multipath area, it is able to maintain sub-metre accuracy 89% of the time. The RSS can also be used in UWB to derive ranges, but as expected, would only yield non-accurate measurements. However, [27] have shown that by combining TW-TOF and RSS in a network containing multiple vehicles, KF can be used to reduce the ranging error from tens of metres to the sub-metre level. This technique not only reduces the ranging error, but can also alleviate the scalability issue when dealing with a dense wireless network. This is discussed further in subsection 2.6.5.

2.6.4 Non-Radio Range CP Measurements

Ranges in VANET can also be derived from sources that do not rely on radio signals of the communication transceivers (DSRC). For example, as will be elaborated in a later section, code-based pseudoranges can be shared amongst the participating vehicles. This means that DSRC acts only as a medium to transfer data between vehicles, which may include vehicles GNSS pseudoranges, positions, and variances.

[4] has demonstrated how the non-radio ranging CP technique can improve relative positioning by exchanging low-level GNSS pseudoranges using DSRC. Their study shows that non-radio ranging between two moving vehicles can achieve better accuracy of relative positioning than differential GPS (DGPS). In this work, the pseudoranges are used to derive code-based double differences, which results in the elimination of spatially correlated GNSS errors such as ionosphere, troposphere, satellite orbit, and clock drifts effects. The proposed system eliminates the infrastructure costs required by conventional systems such as DGPS. The derived double differences are tightly integrated and are then used to estimate the relative positioning between vehicles. The results of the proposed technique show that its CRB improves by up to 30% when compared to that of DGPS. Similarly, its relative positioning RMSE also improves by 37%, which supports its superiority to the conventional technique. However, a shortcoming of the proposed technique is that it needs at least four satellites to be functional. Hence, it might not be
2.6 Wireless Communication for Cooperative Positioning

viable in challenging GNSS environments such as in urban areas.

In trying to improve the non-radio ranging CP technique, this study, detailed in chapter 5, demonstrates a similar approach with an additional sensor, a low-cost MEMS-IMU. Like in the previous technique, in this approach the DSRC is used to transfer information between vehicles, and pseudoranges are used to derive code-based double differences. This work also uses the tight integration approach, where derived double differences, pseudoranges, and IMU measurements are tightly integrated. The tight integration with the IMU approach delivers several benefits, one of which is that the vehicles must only observe at least two common satellites for this technique to function. This system has shown vast improvements compared to conventional tightly coupled IMU/GNSS during limited GNSS availability. In this work, the CP systems performance is tested in three different scenarios: with full GNSS availability, limited availability of three satellites, and lastly, with two satellites. To further observe its performance, the duration of the simulated GNSS partial outages is varied from 60, to 180, and to 300 seconds. Under full GNSS availability, the proposed system provides limited improvement but significantly reduces the positioning error when GNSS availability is limited. For example, up to 60% and 40% improvements over standalone IMU/GNSS are reported when only three and two common satellites are available over 60 seconds, respectively. In other words, the system is able to maintain RMSE of below 2.5 m compared to IMU/GNSS RMSE of 5.7 m. Even after 300 seconds of partial availability, the proposed system continuously outperforms IMU/GNSS by 50% and 25% when three and two satellites are available, respectively.

However, the systems viability in dense urban environments is questionable when code pseudoranges are shared in a VANET. This is due to the shared pseudoranges being corrupted by multipath effects. Hence, this technique is only suitable to be deployed in areas where multipath is not an issue.

2.6.5 Deployment and Ranging in a Wireless Ad-Hoc Environment

The VANETs wireless mobile ad-hoc nature poses several implementation challenges: scalability, MAC delay, limited power sources, security, and quality of service [95]. Of particular interest in this study are scalability and MAC delay. Scalability is defined as
whether the VANET is able to provide the required service even in the presence of a large number of vehicles in the network. This ultimately affects the second problem, where larger networks results in greater MAC delay. The MAC delay is defined as the delay resulting from multiple vehicles wanting to access the same channel to communicate or acquire ranging. As it is not possible for them to access the same channel simultaneously, since this would result in packet collision, the vehicles have to wait for a successful pairing to be complete first, and only then can they access the same channel. Previous studies [39,76,105] have shown that MAC delay is a major factor in providing the required positioning accuracy. An important conclusion from these findings is that increasing the number of participants within a mobile network can sometimes degrade the positioning accuracy due to MAC delays. Hence, there needs to be a balance to optimise the trade-off between performance gains and MAC delay.
The process of successfully disseminating data throughout a network within a specified time frame is already complicated. However, this complexity is even higher when the network has to acquire inter-vehicle ranging via radio signals. For example, consider the situation depicted in figure 2.17, where some amount of data needs to be disseminated from the RSU, marked as a triangle, to all of the vehicles within its vicinity. Due to the building obstruction, the data cannot reach one of the vehicles directly. Therefore, the RSU has to transmit the data to intermediary vehicles first, and then one of them relays that data to the target vehicle. This is commonly known as the hopping method and is widely employed in VANET as a mean to share data. For simultaneous localisation, this becomes more complex than achieving successful data dissemination. As depicted in figure 2.18, all of the vehicles need to acquire inter-vehicle range simultaneously to achieve localisation. If a ranging method is used, TOA, TDOA, TW-TOF, or RSS, for example, these vehicles need to occupy the same channel for each pair of vehicles. As pointed out in [5], given the limited channels available and transmission rate when using DSRC, only a few of the vehicles can perform simultaneous inter-vehicle ranging. The hop method might be used for DSRC, but will result in considerable delay, which severely limits the capability of CP.

In this scenario, UWB might be a better candidate as it can provide faster update rates. [57] has shown that scalability and MAC delay can be improved using UWB through pseudo-noise (PN) code channelisation, where instead of monotonically sending sequential pulses, the delay between pulses is described by a pseudo-random code sequence, which helps to minimise collision with other links operating on the same or alternate PN code. Several other techniques to improve conventional message scheduling methods such as ALOHA and time division multiple access (TDMA) are described in [39,76]. The approach taken in the present work to alleviate the problems arising from scalability and MAC delay differs from past studies, in that it employs MSS. As shown in chapters 5 and 6, the introduction of MSS can help to reduce the dependency on and frequency of using radio ranging, which reduces packet collisions and in turn improves the performance of CP.
2.7 Summary

This chapter has presented a literature review and the background of ITS and its components. Two main parts of the C-ITS were introduced, the first being local-level measurements and the second, network-level measurements. The local-level measurements include GNSS and MSS, while the network-level measurements include ranges and exchanged information within VANET using DSRC and UWB. Note that in this dissertation, MSS refers to the collective system of IMU and VSS. A case study on GNSS performance was presented to further demonstrate the need of this research. Then, GNSS, IMU, and VSS were briefly discussed, and their background, measurements, and error models were presented. The final part of this chapter then presented the network-level observation. Its components were discussed first, followed by their viability to extract ranging measurements. Finally, the chapter discussed the difficulties of implementing a wireless mobile ad-hoc network using DSRC and UWB. The next chapter presents the CP estimation algorithms and evaluates conventional and novel algorithms using simulated VANET data.
Chapter 3

Cooperative Positioning Algorithms

This chapter discusses the estimation algorithms suitable for CP. Before the estimation algorithms are introduced, the chapter first presents the estimation bound, known as the Cramer Rao bound (CRB), which serves as a benchmark for the algorithms. Then, a brief review is presented of the Kalman filter (KF), unscented Kalman filter (UKF), particle filter (PF), and SPAWN. Subsequently, the chapter discusses a major contribution and one of the highlights of this research on improving current CP algorithms. Here, three improved algorithms are introduced: the first is the measurement directed progressive correction (MDPC), which is an improved centralised PF for cooperative positioning in difficult environments; the second is the measurement directed SPAWN (MDSPAWN), an improved SPAWN algorithm; and finally, the third is the federated SPAWN (FSPAWN), which is a semi-decentralised algorithm and has the benefits of both centralised and decentralised algorithms. Note that weighted least squares (WLS) is also used in this study but not elaborated on due to the wide availability of information on this subject, such as in [34,68]. The final part of this chapter evaluates the effectiveness of these algorithms using two-dimensional simulations. It addresses the problem when dealing with the effect of GNSS signal shadowing and multipath, resulting in large positioning uncertainties which severely limit the performance of conventional CP algorithms.

3.1 Performance Expectation Using CRB

The CRB is widely used in parameter estimation to set a lower bound on any unbiased estimator [24,128]. In other words, it allows for the determination of the best possible
outcome of any given scenario; hence, it is useful to efficiently test the correctness and effectiveness of a particular estimator. Naturally, it can also be useful when designing a particular set-up that is, to determine whether a set-up is able to meet system requirements. The normal distribution is chosen as it fits most of the measurements error characteristics dealt with in this thesis. To begin, a generic observation equation can be defined as

\[ z = h(x) + \eta \]  

(3.1)

where

- \( z \) observation vector
- \( x \) unknown parameter vector
- \( h(x) \) vector form function of the parameter vector
- \( \eta \) observation noise vector

The CRB for the \( m^{th} \) parameter of the position estimation vector \( \hat{x} \) is defined by

\[ \text{CRB}(\hat{x}_m) = \left[ F^{-1}(x) \right] \]  

(3.2)

where \( F(x) \) is the Fisher information matrix (FIM), whose elements are defined by

\[ \left[ F(x) \right] = -E \left[ \frac{\partial^2 \ln p(z|x)}{\partial x_m \partial x_{\bar{m}}} \right] \quad m = 1, \ldots, M \]  

(3.3)

where \( M \) is the length of the parameter vector and \( p(z|x) \) is the PDF of the observation for a given parameter vector. As mentioned earlier, the CRB can be used as a performance reference. Hence, it can be used to indicate whether a better estimation algorithm should be pursued when the root mean squared error (RMSE) derived from a particular estimation algorithm, such as LS, KF, or PF, is larger than the one defined by CRB. Consider a cooperative positioning scenario for a VANET, where vehicles \( (x) \), satellites \( (y) \), and inter-vehicle and satellite-vehicle ranges \( (z) \) are available,

\[ z = [z_1 \, \ldots \, z_T]^T \]  

(3.4)
3.1 Performance Expectation Using CRB

\[ x = \begin{bmatrix} x_1 & \cdots & x_M \end{bmatrix}^T \]  
\[ y = \begin{bmatrix} y_1 & \cdots & y_A \end{bmatrix}^T \]  
\[ h(x) = \begin{bmatrix} h_1(x) & \cdots & h_T(x) \end{bmatrix}^T \]  
\[ \eta = \begin{bmatrix} \sigma_1 & \cdots & \sigma_T \end{bmatrix}^T \]  

The distances between vehicles and vehicles, and between vehicles and satellites can be formulated as

\[ h_t(x) = \| x_m - x_\mu \| \]  
\[ h_t(x, y) = \| x_m - y_\mu \| \]  

The PDF \( p(z|x) \) is given as

\[ p(z|x) = \frac{1}{\sqrt{(2\pi)^T \prod \sigma_t}} \exp \left\{ -\sum_{t=1}^{T} \frac{1}{2\sigma_t^2} [z_t - h_t(x)]^2 \right\} \]  

Substituting eq. 3.11 into eq. 3.3, and differentiating respectively yields the FIM.

\[ [F(x)]_{m,m} = \begin{bmatrix} \sum_{t=1}^{T} \frac{(x_{m,x} - y_{a,x})^2}{\sigma_t^2 z_t^2} & \sum_{t=1}^{T} \frac{(x_{m,x} - y_{a,x})(x_{m,y} - y_{a,y})}{\sigma_t^2 z_t^2} \\ \sum_{t=1}^{T} \frac{(x_{m,x} - y_{a,x})(x_{m,y} - y_{a,y})}{\sigma_t^2 z_t^2} & \sum_{t=1}^{T} \frac{(x_{m,y} - y_{a,y})^2}{\sigma_t^2 z_t^2} \end{bmatrix} \]  

\[ [F(x)]_{m,\theta} = \begin{bmatrix} \sum_{t=1}^{T} \frac{(x_{m,x} - x_{\theta,x})^2}{\sigma_{z_t}^2 z_t^2} & \sum_{t=1}^{T} \frac{(x_{m,x} - x_{\theta,x})(x_{m,y} - x_{\theta,y})}{\sigma_{z_t}^2 z_t^2} \\ \sum_{t=1}^{T} \frac{(x_{m,x} - x_{\theta,x})(x_{m,y} - x_{\theta,y})}{\sigma_{z_t}^2 z_t^2} & \sum_{t=1}^{T} \frac{(x_{m,y} - x_{\theta,y})^2}{\sigma_{z_t}^2 z_t^2} \end{bmatrix} \]  

where the subscripts \( x \) and \( y \) denote the axes of the estimates. Finally the CRB can be calculated using equation 3.2.
3.1.1 On Singular FIM in Localisation

In some cases, the singular FIM might arise in localisation problems, for example when there is not enough information to estimate the parameters, or in a network that suffers from non-identifiable parameters. These problems, particularly the latter, typically occur when the VANET is operating in urban environments, where grid-like street layouts are present. As a workaround to this problem, this study adopts the approach proposed in [108]. Instead of using the pseudoinverse method, which often leads to an overly optimistic bound, the alternative approach derives the CRB as the solution to an unconstrained quadratic maximisation problem. For non singular FIM \( F \), the covariance matrix of any estimator is lower bounded by the CRB matrix such that

\[
E\{(\hat{x} - \mu)(\hat{x} - \mu)'^\prime\} \geq GF^{-1}G'
\]  

where

\[
\mu = E\{\hat{x}\} = \int_{R_T} \hat{x}p(z|x)dz
\]  
\[
G = \frac{\partial \mu}{\partial x'}
\]

For singular \( F \), the Moore-Penrose pseudoinverse \( J^\dagger \) can be used such that

\[
E\{(\hat{x} - \mu)(\hat{x} - \mu)'^\prime\} \geq HJ^\dagger H'
\]

where

\[
\frac{\partial b}{\partial x'} + \frac{\partial \mu}{\partial x'} \equiv H
\]

that needs to satisfy

\[
H = HJ^\dagger
\]

Note that the singular estimation problem is now handled using a biased estimator \( b \). Simply taking the pseudoinverse of \( J \) when it is singular, as seen in eq. 3.17, will generally result in an overly optimistic lower bound. Further manipulation to avoid this problem
is to utilise the eigenvector representation of $J$

$$J = U \Lambda U' = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Lambda_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_1' \\ U_2' \end{bmatrix}$$  \hspace{1cm} (3.20)$$

where $\Lambda_1 \in \mathbb{R}^{r \times r}$, $r$ is the rank of $J$ and

$$\begin{bmatrix} H_1 & H_2 \end{bmatrix} = H \begin{bmatrix} H_1 & U_2 \end{bmatrix}$$  \hspace{1cm} (3.21)$$

Without providing more details on its derivation, the lower bound can be defined as equation 3.22. Further information on the derivation can be found in [10, 108].

$$C \geq H J^\dagger H' + \sigma H_2 H'_2$$  \hspace{1cm} (3.22)$$

3.2 Review of Current CP Algorithm

This section provides a review of common estimation methods that are suitable for CP. These methods can be categorised as non-Bayesian and Bayesian. A non-Bayesian estimator assumes that the parameters to be estimated, which are usually unobservable, are non-random and constant during the observation window. However, the observations are noisy and thus have random components. A Bayesian model, on the other hand, assumes that the parameters are random variables that have a prior probability, and the observations are noisy as well [49]. In this dissertation, only one non-Bayesian estimator is used: the WLS. All other estimators are Bayesian-based.

3.2.1 Kalman Filter

The Kalman filter is a common estimator used in positioning problems. It extends the principles of WLS by including the dynamics of the state vector as part of the estimation process. This allows for better estimation of the state vector. This section provides an introduction to the KF and its extensions, EKF and UKF, applied to non-linear systems.
Kalman Filter

The KF is a recursive algorithm that uses a series of prediction and measurement update steps to obtain an optimal (in a minimum variance sense) estimate of the state vector [41]. As previously mentioned, the KF makes use of both the measurement and process or dynamics of a system to provide a better estimate of the unknowns. Therefore, the following focuses on describing the system process or dynamics. Firstly, the dynamic of a linear system must be defined. According to the linear system theory, this can be represented by a vector of differential equation in a continuous form

$$\dot{x}(t) = F(t)x(t) + G(t)\omega(t)$$  (3.23)

where
- $F(t)$: Dynamic matrix at time $t$
- $G(t)$: coefficient matrix to shape the noise input at time $t$
- $\omega(t)$: vector of system noise at time $t$, with spectral density $Q(t)$

and as before, the measurement equation is defined by equation 3.1, repeated here for convenience

$$z = h(x) + \eta$$  (3.1)

It is important to note that $\omega(t)$ and $\eta(t)$ are independent of each other, white, and normally distributed. White noise is characterised as having zero mean and no correlation between observations, i.e., $P(\omega) \sim N(0, Q)$ and $P(\eta) \sim N(0, R)$. $Q$ is the process noise and $R$ is the measurement noise variance matrices. Since estimation implementation is done with a computer, the continuous-time equation needs to be transformed into a discrete time equation

$$x_{k+1} = \Phi_{k+1,k}x_k + \omega_k$$  (3.24)

where
- $\Phi_{k+1,k}$: transition matrix from $t_k$ to $t_{k+1}$
- $x_k$: state vector at epoch $t_k$
- $\omega_k$: vector of system noise at epoch $t_k$ with covariance matrix $Q_k$
Having defined the process and measurement equations in discrete time, it is now appropriate to define the process of the KF. This process can be divided into two main groups: prediction and update. Essentially, the prediction group (equations 3.25 and 3.26) describes how the state vector and its covariance propagate through time, based on the current state and assumed system model. This is termed the prior estimate (denoted by a superscript negative sign). The update group (equations 3.27, 3.28 and 3.29) updates the Kalman gain, state vector, and system variance. The components in this group are referred to as posterior estimates. The Kalman gain, in a loose sense, weighs the process and observations accordingly, taking into account their respective variances. Using the Kalman gain, the state vector is updated with new observations. Finally, the system variance is updated, using both the Kalman gain and a priori variance [34,120]. The algorithm is then recursively applied to subsequent epochs. The following lists the discrete time equations involved in the KF.

**Prediction step**

\[
\hat{x}^-_k = \Phi_{k,k-1}\hat{x}^+_k
\]
\[
P^-_k = \Phi_{k,k-1}P^+_k\Phi_{k,k-1}^T + Q_k
\]

**Time update step**

\[
K_k = P^-_kH_k^T (H_k P^-_k H_k^T + R_k)^{-1}
\]
\[
\hat{x}^+_k = \hat{x}^-_{k-1} + K_k(z_k - H_k\hat{x}^-_k)
\]
\[
P^+_k = (I - K_kH_k) P^-_k
\]

As presented above, the KF assumes that the process and measurement models are linear. However, most real systems are best described by non-linear equations [120]. Such is the case when using MSS to navigate, where PVA calculations take into account the spherical earth. The non-linearity is also observed in the GNSS system model of processing ranging observations. To implement the KF into a non-linear system, the model needs to be linearised first. Only then can the KF be applied to estimate the state vector.
The non-linear version of KF, which linearises about the current mean and covariance, is termed the EKF [120]. The dynamic and measurement models of the EKF are

\[ \dot{x}(t) = f(x(t)) + G(t)\omega(t) \] (3.30)
\[ z(t) = h(x(t)) + \eta(t) \] (3.31)

where \( f \) and \( h \) are known non-linear functions replacing the dynamic and design matrices in the process and measurement models respectively. For linearisation, a nominal trajectory is selected, as given by

\[ x_{k+1} = x_{k+1}^* + \delta x_{k+1} \] (3.32)

where \( ^* \) represents the nominal state vector value and \( \delta \) represents the perturbation from the nominal value. Then, the first order Taylor series expansion is performed, assuming the perturbations are sufficiently small about the selected nominal trajectory, resulting in

\[ \delta x_{k+1} = \Phi_{k+1} \delta x_k + \omega_k \] (3.33)
\[ \delta z_{k+1} = H_{k+1} \delta x_{k+1} + \eta_{k+1} \] (3.34)

The first order partial derivatives matrix is also known as a Jacobian matrix. Equation 3.33 is the linearised process model, where the state vector is now replaced by the state-error vector. Similarly, equation 3.34 is the linearised measurement model, where the vector is replaced by measurement misclosures between the actual and predicted observations. The transition and design matrices in equations 3.33 and 3.34 are now a function of partial derivatives of the respective non-linear functions with respect to the state vector [43]. The linearisation is then followed by the steps of the KF (prediction and update).

**Unscented Kalman filter**

Unlike the EKF, the UKF does not require the explicit calculation of Jacobians. Instead, the approximation of the Gaussian distribution method is efficiently done by creating a discrete distribution of vector points, which allows a non-linear transformation to be ap-
3.2 Review of Current CP Algorithm

plied to each of the points independently [49, 130]. This approach alleviates the problem of dealing with non-linear systems, e.g. positioning systems, resulting in a better estimation process where the information lost resulting from neglecting the higher-order terms is avoided. Studies such as [126, 130] have shown that UKF does help to provide better performance in terms of accuracy and faster state convergence when compared to EKF.

The general procedure of UKF is as follows. First, the points, termed sigma points, are drawn:

\[
\chi_i = \hat{x}_{k-1}^+ + \left(\sqrt{nP_{k-1}^+}\right)_i, \quad \chi_{i+n} = \hat{x}_{k-1}^- - \left(\sqrt{nP_{k-1}^-}\right)_i
\]

(3.35)

where \(i = 1, \ldots, n\) and \(n\) denotes the dimension of the sigma points. The sigma points are then used to calculate the state and its corresponding covariance in the time update step.

\[
\hat{x}_k^- = \frac{1}{2n} \sum_{i=1}^{2n} f_{k-1}(\chi_i)
\]

(3.36)

\[
\hat{P}_k^- = \frac{1}{2n} \sum_{i=1}^{2n} [f_{k-1}(\chi_i) - \hat{x}_k^-] [f_{k-1}(\chi_i) - \hat{x}_k^-]' + Q_{k-1}
\]

(3.37)

Now, a new set of sigma points are drawn using the updated state.

\[
\chi_i = \hat{x}_k^- + \left(\sqrt{nP_k^-}\right)_i, \quad \chi_{i+n} = \hat{x}_k^- - \left(\sqrt{nP_k^-}\right)_i
\]

(3.38)

Subsequently, the measurements are predicted and used to calculate the covariances.

\[
\hat{z}_k = \frac{1}{2n} \sum_{i=1}^{2n} h(\chi_i)
\]

(3.39)

\[
P_k^{zz} = \frac{1}{2n} \sum_{i=1}^{2n} [h_k(\chi_i) - \hat{z}_k] [h_k(\chi_i) - \hat{z}_k]' + R_k
\]

(3.40)

\[
P_k^{xz} = \frac{1}{2n} \sum_{i=1}^{2n} [h_k(\chi_i) - \hat{x}_k] [h_k(\chi_i) - \hat{z}_k]'
\]

(3.41)

Finally, the UKF follows the normal KF steps where the Kalman gain, new state update,
and covariances are calculated.

\[ K_k = P_k^{zz} (P_k^{xz})^{-1} \]  
\[ \hat{x}_k^+ = \hat{x}_k^- + K_k (\hat{z}_k - \hat{z}_k) \]  
\[ P_k^+ = P_k^- - K_k P_k^{zz} K_k' \]  

3.2.2 Particle Filter

Like the KF, the PF uses the Bayes law to obtain the posterior conditional density of \( x \). Studies such as [105, 126] have shown that the PF does outperform EKF and UKF when used for CP, but at the expense of a greater computational requirement. The Bayesian approach assumes that a prior distribution exists for a state vector of parameters, \( x \sim \pi_0 \) [81]. The objective of Bayesian estimation is to find the posterior distribution, conditional on given observations. Using the Bayes rule, the posterior PDF is given as

\[ \pi(x) \propto l(z|x)\pi_0(x) \]  

The minimum mean squared error (MMSE) estimator of \( x \) is the posterior mean, and can be calculated using

\[ \hat{x} = E(x|z) = \int x \pi(x) dx \]  

where \( \pi \) is the posterior PDF (\( p(x|z) \)). The exact posterior mean is difficult to evaluate but can be approximated via importance sampling. This involves drawing examples of the parameter vector \( x \) from a given importance density \( q \). Then, the integral can be approximated using a weighted sum of samples

\[ \hat{x} \approx \sum_{i=1}^{n} w^i x^i \]  

where \( i = 1, \ldots, n, \); \( w^i \) is the weight for each sample and

\[ x^i \sim q \]  
\[ w^i \propto l(z|x^i)\pi_0(x^i)/q(x^i) \]
and the likelihood is calculated using

\[ l(z|x) = \prod_{t=1}^{T} N(z_t; h_t(x, z), \sigma^2) \]  

(3.50)

where \( T \) is the number of observations. There is a great deal of freedom in the choosing of \( q \), with the better choices generating sample values that lie in the region important for the value of the integral [49]. However, it is difficult to find an accurate representation of \( q \), particularly when the sample size is small but covers a large region. The simplest substitute is often the prior, \( q = \pi_0 \). Thus, the weights would simply be \( w^j \propto l(d|x^j) \). This approach might be straightforward, but it is unsuitable for difficult localisation problems where the prior is much more diffused than the true likelihood [81, 89]. The alternative method is discussed in later subsections. For now, the PF can be summarised as the following. First, draw samples based on the prior distribution and initial belief or weight for each sample.

\[ x^j \sim \pi_0 \]  

(3.51)

\[ w^j = 1/n \]  

(3.52)

Then, calculate the weight for each sample based on computed likelihood using equations 3.49 and 3.50. Finally, the posterior mean of the state vector is calculated using equation 3.47.

### 3.2.3 Sum Product over Ad-Hoc Wireless Network

The final conventional estimation algorithm used in this dissertation is SPAWN. It is a fully distributed estimation algorithm, which gives it an advantage over the other algorithms. Operating in a distributed manner allows for scalability that does not require fixed infrastructure in VANET [40]. SPAWN is based on the factor graph (FG) and sum product algorithm (SPA). The following presents the basics of these concepts, adopted from [22, 125].

The FG is used to graphically represent a factorisation and can be used in inference
problems, where the posterior distribution can be factorised. Here, each factor is denoted as \( \phi_J(\cdot) \). Then,

\[
p(x|z) = \frac{1}{J} \prod_{j=1}^{J} \phi_k(x)
\]

where \( J \) is the number of factors and \( Q \) is a normalisation factor. Assume that for every factor, a vertex is created and for every variable \( x_m \), an edge is created. When a variable is in a particular factor, it becomes an edge connecting the vertexes. In the context of CP in a static environment, two factors can be formed: the first describes the connection to available satellites, and the second, the connection to neighbouring vehicles. This can better be described by referring to sub-figure 3.1a where three vehicles are connected to each other: vehicles 1 and 3 are only connected to one satellite, while vehicle 2 is connected to two satellites. The corresponding FG for this network is depicted in sub-figure 3.1b.

The factorisation for this specific example can be written as

\[
\prod S_{a,m}(x_m) \prod h_{m,y}(x_m, x_{\neq m})
\]

where \( S_{a,m}(x_m) \) represents the pseudorange likelihood given the state of \( x_m \) and \( h_{m,y}(x_m, x_{\neq m}) \) represents the likelihood of the inter-range vehicular ranges given the state of \( x_m \) and \( x_{\neq m} \).

From equation 3.54, it is clear that the factors are the probability functions to calculate the likelihood of the measurements. The next step is to define the message update rule over
3.2 Review of Current CP Algorithm

As its name implies, the SPA is a message passing algorithm that efficiently computes $b_x$, which is defined as belief of $x$. The SPA consists of computing messages within the vertices (vehicles) and sending these messages over the edges to other vertices. Referring to the example in figure 3.1, there are three type of messages: messages from satellite factors, messages from adjacent vehicle factors, and messages to adjacent vehicle factors. Here, the message from satellite factors is denoted by $\eta_{S_{a,m} \rightarrow x_m}(x_m)$. Because satellite position is known from its ephemeris, the message is simply the likelihood of the satellite pseudoranges:

$$\eta_{S_{a,m} \rightarrow x_m}(x_m) = p(x_m | \rho_{a \rightarrow m}, r_a, \sigma_{a \rightarrow m}^2)$$  \hspace{1cm} (3.55)

where $\rho_{a \rightarrow m}$ is the pseudorange, $r_a$ is the satellite position and $\sigma_{a \rightarrow m}^2$ is the variance of the pseudoranges. Then, the messages from adjacent vehicles factors containing the likelihood of the inter-ranges and PDF of the adjacent vehicle, can be written as

$$\eta_{h_{\hat{g},m} \rightarrow x_k}(x_k) \propto \int h_{\hat{g},m}(x_{\hat{g}}, x_m) \times \eta_{x_m \rightarrow h_{\hat{g},m}}(x_{\hat{g}}) \partial x_{\hat{g}}$$  \hspace{1cm} (3.56)

where $h_{\hat{g},m}(x_{\hat{g}}, x_m)$ is the likelihood of the inter-vehicle range given the position $x_{\hat{g}}$, and $\eta_{x_m \rightarrow h_{\hat{g},m}}(x_{\hat{g}})$ is the message containing the PDF of $x_{\hat{g}}$. The proportionality symbol in equation 3.56 indicates that the message is normalised, such that $\int \eta_{h_{\hat{g},m} \rightarrow x_k}(x_m) \partial x_m = 1$.

The final message is the outgoing message to adjacent vehicle factors, written as

$$\eta_{x_m \rightarrow h_{\hat{g},m}}(x_m) \propto \prod_{s \in S_m} \eta_{S_{a,m} \rightarrow x_m}(x_m) \prod_{z \in M_m} \eta_{h_{z,m} \rightarrow x_m}(x_m)$$  \hspace{1cm} (3.57)

Coincidently, this is also the belief of $x_m$ which is the approximation of the marginals of $x_m$. It is important to note that there are two methods of representing a message in SPA: parametric and sample-based message representations [74]. The use of sample-based message representation, also known as non-parametric belief propagation, can better capture the true distribution at the cost of a larger sampling size. However, as shown in [81], this can be reduced by applying the measurement directed progression correction method. Hence, the sample-based message representation is used in this dissertation.
The outgoing message naturally has to have the updated position of the vehicles based on received messages. This is sometimes known as message multiplication. Since this study uses the sample-based message representation, the PF estimation algorithm, discussed in the previous subsection, is used to compute the updated vehicle positions.

The SPAWN process described so far has only involved static conditions. In the context of mobile vehicular networks, however, mobility can also be included as part of the SPAWN process. This involves the inclusion of a new factor \( f_{m}(x_{m,k}, x_{m,k-1}) \) in equation 3.54 and subsequently a new message, called the temporal message, which can be expressed as

\[
\eta_{f_{m}\rightarrow x_{m}}(x_{m}) \propto \int f_{m}(x_{m,k}, x_{m,k-1}) \times p(x_{m,k-1}|r_{1:k-1}, r_{1:k-1}) \, dx_{m}
\]  (3.58)

As a result, this message must be incorporated into equation 3.57. Rewriting this equation yields

\[
\eta_{x_{n}\rightarrow h_{\hat{m}, m}}(x_{m}) \propto \eta_{f_{m}\rightarrow x_{m}}(x_{m}) \times \prod_{s \in S_{m}} \eta_{S_{a,m}\rightarrow x_{m}}(x_{m}) \prod_{z \in M_{m}} \eta_{h_{z,m}\rightarrow x_{m}}(x_{m})
\]  (3.59)

Overall, the SPAWN process can be summarised as shown in pseudo algorithm 1.

Algorithm 1 SPAWN

1: for Node \( m = 1 \ldots, M \) do
2:  Draw \( x_{m} \sim \pi_{0} \)
3:  Set initial belief \( b_{m}^{i} = 1/n \)
4: end for
5: for Timestep \( k = 1 \ldots, K \) do
6:  for Node \( m = 1 \ldots, M \) do in parallel
7:     Compute temporal message, \( \eta_{f_{m}\rightarrow x_{k}} \)
8:     Broadcast predicted distribution, \( \eta_{x_{n}\rightarrow h_{\hat{m}, m}} \)
9:     Collect pseudoranges and inter-vehicle ranges
10: end for
11: for iteration \( it = 1 \ldots, IT \) do
12:  for Node \( m = 1 \ldots, M \) do in parallel
13:     Receive messages, \( \eta_{x_{n}\rightarrow h_{\hat{m}, m}} \), from adjacent neighbours
14:     Compute local factor to variable messages, \( \eta_{S_{a,m}\rightarrow x_{m}}, \eta_{h_{z,m}\rightarrow x_{m}} \)
15:     Compute and broadcast outgoing V2V messages, \( \eta_{x_{n}\rightarrow h_{\hat{m}, m}} \)
16:     Update and broadcast beliefs \( b_{m}^{i} \)
17: end for
18: end for
19: end for
3.2 Review of Current CP Algorithm

3.2.4 Algorithm Performance Based on 2D Simulation

This subsection tests the performance of the algorithms using a static 2D simulation model. It is well known that under well-defined conditions, such as when there are enough observations to correctly estimate the state vector, these algorithms perform comparatively. Hence, this subsection only presents results from a simulation with a challenging set-up. In particular, the algorithms are tested where none of the nodes or vehicles are connected to two anchors simultaneously, and are therefore unable to self-localise solely based on observations from the anchors. Instead, the nodes are only able to solve for their positions by cooperation. This subsection is structured as follows. As a baseline, the algorithms are first tested with nodes that have a relatively good prior distribution (∼10 m), and then again with a large prior distribution (∼100 m).

![Figure 3.2: Simulation setup](image)

Two simulations are used to evaluate the effectiveness of the algorithms discussed in the previous subsection. As shown in figure 3.2, both simulations consist of four anchors, but the first simulation includes only four nodes, while the second contains 10. The green lines represent the connections between anchors and nodes, while the blue lines represent connections between nodes. The first simulation is set up such that the four nodes are only connected to one anchor, and each node is connected to its adjacent
nodes. On the other hand, in simulation 2, none of these four nodes are interconnected directly; instead, they are connected to the remaining six nodes. It is clear from the plots that none of the nodes have two direct connections to any of the anchors, which makes localising these nodes quite challenging. The standard deviation of inter-ranges is set to \(0.5 \, \text{m} \, N(0, 0.5^2)\), reflecting UWB range noise obtained from a test done as part of this study. Both simulations contain 50 time steps. The computed CRBs for simulation 1 and 2 are 0.31 m and 2.30 m, respectively. The performance of these algorithms is quantified using position RMSE. Because SPAWN utilises sample-based approximation, its RMSE is computed using 50 Monte Carlo realisations.

Table 3.1 lists the averaged RMSE over the last 20 simulation time steps. In this table, simulations with small and large prior distribution are labelled as x.1 and x.2, respectively. Several notable conclusions can be drawn from this table. First, WLS has the worst performance and cannot solve for the positions when estimating with large prior distribution. Even when small prior distribution is used, its RMSE is large when compared to the CRB value. For example, its RMSE for simulation 1.1 is 8.89 m compared to the achievable CRB of 0.31 m. EKF and UKF perform comparatively for simulation 1, while UKF performs slightly better for simulation 1.2. On the other hand, the results from simulation 2.2 reveal that UKF significantly outperforms EKF due to its ability to better handle non-linear systems. However, both estimators are still unable to reach the achievable RMSE for both simulations listed by CRB. Among these conventional algorithms, the sample-based SPAWN achieves the best performance in terms of RMSE. In the worst-case scenario of simulation 2.2, its RMSE is 1.65 m and 20.06 m better than that of UKF and EKF, respectively. Yet, it is still unable to achieve an RMSE that is close to the CRB value for simulations with large prior distribution.

Observing the RMSE over the simulation time series reveals more in-depth error characteristics. Figures 3.3 and 3.4 plot the calculated RMSE of the tested algorithms for simulations 2.1 and 2.2. Note that WLS is not included in simulation 2.2 as it is unable to produce acceptable results. The left sub-figure shows the overall RMSE over time while the right sub-figure shows the magnified plot, where the y-axis is reduced to better demonstrate the RMSE values after convergence. Comparing figures 3.3 and 3.4, it can be seen
that the convergence is much faster when the prior distribution is small. While it takes two or three time steps for the RMSE values to converge for all algorithms except for WLS in simulation 2.1, they appear to take much longer to converge for simulation 2.2: EKF starts to converge at the 20 s mark, and UKF and SPAWN convergence start at around the 40 s and 30 s marks. Furthermore, while the algorithms can achieve acceptable RMSE values for simulation 2.1, they do not fare as well in simulation 2.2, particularly EKF, whose RMSE stays at 24.20 m. These figures also reveal that SPAWN performs the best among the algorithms. However, due to the challenging nature of the simulation, with none of the nodes able to self-localise using the anchors only, the algorithm still cannot reach the expected achievable performance as indicated by the CRB. This is further exacerbated when dealing with large prior distribution: in this case, none of the algorithms seem to achieve an RMSE close to the CRB value.
3.2.5 Effects of Large Prior Distribution

The results in the previous subsection clearly reveal that a large prior distribution severely hampers the ability of estimators to achieve the best possible performance. In the context of the VANET in urban navigation, this effect on vehicles is caused by GNSS signal shadowing and multipath. Although cooperative positioning has been extensively studied done, for example in [40, 125], no author has particularly highlighted the problem of dealing with a large prior distribution in difficult positioning set-ups, as demonstrated in subsection 3.2.4. This subsection visually explains this effect and discusses a method to overcome the problem.

Figure 3.4: RMSE for simulation 2.2 (large prior distribution)

Figure 3.5: Simulation with concentrated prior
Consider simulations 1.1 and 1.2, explained in the previous subsection, shown here as figures 3.5 and 3.6. One thousand normally distributed samples are generated with a standard deviation of 10 m, centred at the initial mean of around 10 m from the true position for simulation in figure 3.5. On the other hand, the samples are uniformly distributed across the whole region in simulation 1.2, seen in figure 3.6, simulating large prior uncertainty and distribution. Both simulations are then run through a sample-based estimation method (PF) where subsequently, only samples with high probability (top 20%) are redrawn in figures 3.5b and 3.6b.

It can be seen that the resulting samples with higher probability in figure 3.5b are concentrated at the true position, whereas the samples in figure 3.6b are still diffused or sparse. These figures clearly show the effect of a large prior distribution in difficult cooperative estimation scenarios. Essentially, the estimation process becomes easier when samples are closer to their true position, but becomes much more difficult when they are more diffused. This is due to the prior density or distribution of the parameters being much more diffused than the ranging measurements likelihood. To overcome this problem, the progressive correction method is adopted in this dissertation. The following subsection briefly describes the process involved when employing this method.
3.2.6 Progressive Correction

This subsection discusses the progressive correction method employed in this study to overcome the problem of dealing with large prior uncertainty. This method is largely adopted from [81]. First, recall that the likelihood of observations can be calculated as

$$ l(z|x) = N(z; h(x, y), \sigma_z) $$

(3.60)

where $y$ represents the nodes that are connected to node $x$. The progressive correction approach is implemented in stages ($S$), where at each stage, the likelihood becomes closer to the true likelihood. To implement this approach, equation 3.60 is rewritten as

$$ l_s(z|x, \hat{x}) = N(z; h(\hat{x}, y), \sigma_z)^{\Gamma_s} $$

(3.61)

where $l_s$ represents the likelihood at the $s$th stage and $\gamma$ is the progressive value such that, $\Gamma_s = \sum_{j=1}^{S} \gamma_j, \gamma_j \in (0, 1]$ and $\Gamma = 1$. It is clear from equation 3.61, that the likelihood at $s$ is greater than the true likelihood. Hence, this forces the samples drawn from the diffuse prior distribution to have a higher likelihood than they actually have, particularly at earlier stages. Gradually, the likelihood sharpens so that the samples become more concentrated in the area of the parameter space suggested by the true likelihood.

This process can be visualised in figures 3.7a-3.8b, where the red line is the likelihood of computed observations based on equation 3.60, the blue line is the proposed likelihood with progressive correction based on equation 3.61, and the black line is the actual density of estimates based on observations.

As seen in figure 3.7a, using equation 3.60 to compute the likelihood results in many prior samples being ignored at the first iteration. This could lead to an unwanted consequence: samples that seem unlikely at first, i.e. low likelihood, could be completely discarded. A more ideal approach is to not ignore those with lower likelihood all at once, but instead to discard them in a progressive manner as the likelihood sharpens. At each iteration, new samples are drawn based on the previous correction stages posterior, gradually becoming more concentrated within the area of the true likelihood. This is seen in figures 3.7b, 3.8a and 3.8b, where the progressive correction enables for the sample den-
3.2 Review of Current CP Algorithm

Although the progressive correction does help in perturbing the large prior uncertainty problem, its effectiveness in complex network, such as when dealing with large parameters i.e. high number of participating vehicles in VANET, is limited. This comes down to the choice of importance density \( q \), where so far, has been assumed to be \( \pi_0 \) (prior density). [81] has shown a better way of approximating \( q \) that would closely follow the true density. The technique, termed as measurement directed progressive correction (MDPC), uses observations, to direct the samples to move towards more probable positions. This is further discussed in the next section.

Figure 3.7: Probability density at iteration 1 and 2

Figure 3.8: Probability density at iteration 4 and 10
3.3 Improved CP Algorithm

The previous section has outlined conventional algorithms typically used in CP. EKF, UKF, and PF do provide a straightforward approach when dealing with CP, but they are often unable to provide accurate results, particularly when dealing with large prior uncertainty. This section showcases the improved algorithms for CP adopted from previous works and further developed in this dissertation. These are the measurement directed progressive correction (MDPC), which is a variant of PF; MDSPAWN, which adds the measurement directed method to SPAWN; and FSPAWN, which is a semi-centralised/decentralised algorithm. As shown in later subsections, these algorithms would overcome the aforementioned problem by improving conventional PF and SPAWN algorithms.

3.3.1 Measurement Directed Progressive Correction

The MDPC discussed here is adopted from [81]. The MDPC is a centralised, sample-based estimation algorithm. Unlike generic PF, the basis of this technique is to only use samples with refined likelihood by moving prior samples using a KF update and accordingly choosing the method of calculating weight for each sample. As highlighted in [81], one of the advantages of MDPC is its ability to achieve RMSE close to the derived CRB, even when using a small number of samples. Essentially, its construct is based on three principles:

- Utilising the progressive correction method to avoid sample impoverishment;
- Moving samples into a more likely place based on available measurements; and
- Calculating the appropriate importance density, $q$.

The progressive correction method has already been discussed in the previous subsection, and is therefore not elaborated further here. Naturally, appropriate power for the correction progressive correction stages is chosen first, such that $\Gamma_s = \sum_{j=1}^{S} \gamma_j$, $\gamma_j \in (0, 1]$ and $\Gamma = 1$, as discussed in subsection 3.2.6, before carrying out the procedures for the last
two principles, which are discussed in the following subsections. Furthermore, $s$ denotes the correction stage index and $i$ denotes the sample index in the following subsections.

**Kalman Update**

The term *measurement directed* measurement directed in MDPC stems from the fact that the samples are moved according to the available measurements or observations using the concept of the KF time update. In turn, the samples are placed in a more probable area instead of their original placement based on prior distribution. Like a normal KF, in the time update step, the predicted measurements $\hat{z}$, innovation covariance $S$ and Kalman gain $K$ are computed for each sample. Then, the sample update $\hat{x}$ takes place by using the Kalman gain, samples from the previous correction stage, measurements and predicted measurements. Finally, its corresponding posterior covariance $P$ is calculated.

$$\hat{z}_{s,i} = h(x^{s-1,i}, y) \quad (3.62)$$

$$S_{s,i} = H(x^{s-1,i})\Sigma_{s-1}H(x^{s-1,i})^T + \sigma^2 I / \gamma_s \quad (3.63)$$

$$K_{s,i} = \Sigma_{s-1,i}H(x^{s-1,i})^T(S_{s,i})^{-1} \quad (3.64)$$

$$\hat{x}_{s,i} = x^{s-1,i} + K_{s,i}(z - \hat{z}_{s,i}) \quad (3.65)$$

$$P_{s,i} = \Sigma_{s-1} - K_{s,i}H(x^{s-1,i})\Sigma_{s-1} \quad (3.66)$$

**Selection Weights $\psi$**

The next step is to calculate the weights of the moved samples, termed as selection weights. This process considers the weights from the previous correction stage and the likelihood of the measurements based on the predicted measurements and innovation covariance.

$$\psi_{s,i} \propto w^{s-1,i} \int l(z|x, x^{s-1,i})g_{s-1}(x - x^{s-1,i})dx$$

$$= D \cdot w^{s-1,i} N(z; \hat{z}_{s,i}, S_{s,i}) \quad (3.67)$$

where $D$ is the normalising constant and $g_{s-1}$ is a kernel density.
Re-sample

Next, both samples from the current correction stage and from the previous stage are resampled. This is done to avoid the degeneracy problem, where the weights are too concentrated only in a few samples while the majority of the samples have weights of 0. The resampling process avoids this problem by discarding samples with negligible weights and only retaining samples with higher weights, and some among these are duplicated, depending on their weights [49]. The present study employs the stratified resampling method. The samples are resampled based on computed $\psi^s$, and indexed as $j^s$. Note that both updated samples $\hat{x}^s$ and samples from the previous correction stage $x^{s-1}$ are re-sampled, resulting in $\hat{x}^s$ and $x^{s-1}$.

Parameter Sampling Density (Regularise)

The next step is to regularise the samples to avoid sample impoverishment, which occurs when the resultant sample set contains many repeated samples for any given weight [49]. The solution to this problem is to move the concentrated samples by adding controlled noise to regain diversity. The parameter sampling density can be expressed as

$$f^s(x) = \frac{\hat{I}(z|x, x^{s-1}) \gamma^s g_{s-1}(x - x^{s-1})}{\int \hat{I}(z|\bar{\xi}, x^{s-1}) \gamma^s g_{s-1}(\bar{\xi} - x^{s-1}) d\bar{\xi}} = N(x; \hat{x}^s, \sigma^s)$$ (3.68)

where it makes use of the resampled samples and posterior covariance. Then, it is used to regularise the resampled samples.

Sample Weights $w$

The final process in each correction stage is to calculate the sample weights $w$ using

$$w^s = C[I(z|x^s)/\hat{I}(z|x^s, x^{s-1})]\gamma$$ (3.69)
where the linearised likelihood of $x$ about the point $\hat{x}$ given measurement $z$ is

$$
\hat{L}(z|x, \hat{x}) = N(z; h(\hat{x}, y) + H(\hat{x})(x - \hat{x}), \sigma^2 I_T)
$$ (3.70)

Equation 3.69 is derived from

$$
\hat{L}(z|x_s, x_{s-1}, j^s_i) = N(z; h(x_{s-1}, j^s_i, y) + H(x_{s-1}, j^s_i)(x_s - x_{s-1}, j^s_i), \sigma^2 I_T)
$$ (3.71)

Note that when computing the measurement likelihood, it is more convenient to work with the natural logarithm of the likelihood function when dealing with a large set of measurements. This is further discussed in appendix D.

**Algorithm 2 Measurement Directed Progressive Correction**

1. Select $\gamma_1, \ldots, \gamma_S$
2. for Sample $i = 1, \ldots, n$ do
3. Draw $x_0, i \sim \pi_0$
4. Set initial weight $w_0, i = 1/n$
5. end for
6. for Correction stage $s = 1, \ldots, S$ do
7. Move samples using the KF update, $\hat{x}^s_i$
8. Compute selection weights - $\psi^s_i$
9. Re-sample samples, based on $\psi^s_i$
10. Find indices of $j^s_i$
11. Move both $\hat{x}^s_i$ and $\hat{x}^{s-1}_i$ resulting in $\hat{x}^s_{i,j^s_i}$ and $\hat{x}^{s-1}_{i,j^s_i}$
12. Regularize based on updated $f^s_{i,j^s_i}$
13. Compute sampling density $f^s_{i,j^s_i}$
14. Draw new $\hat{x}^s_i \sim f^s_{i,j^s_i}$
15. Compute weights $w^s_i$
16. Compute $h(x^{s-1}_i, j^s_i, y), H(x^{s-1}_i, j^s_i)$
17. Compute weights $w^{s,i}$
18. end for
19. Compute parameter estimate $\hat{x} = \sum_{i=1}^n w^{s,i} x^{s,i}$
3.3.2 Measurement Directed SPAWN

The second proposed algorithm for CP is the MDSPAWN, which adopts the technique of MDPC in a decentralised way. This provides a mean to handle scalability better than MDPC does. Similar to MDPC, both the progressive correction and measurement directed methods are added to the conventional SPAWN. By employing these techniques, discussed in sections 3.2.3 and 3.3.1, more meaningful measurements can be derived in SPAWN. In particular, using the progressive correction method relaxes the computation of measurement likelihood, which helps in dealing with large prior uncertainties, and using the KF update step to move the samples using available measurements helps more samples to be more meaningful in the estimation process. Because these processes have been explained in previous sections, they are not elaborated here; however, a summary of this method is presented as pseudo-algorithm 3.

**Algorithm 3 Measurement Directed SPAWN**

1: Select $\gamma_1, \ldots, \gamma_S$
2: for Node $m = 1, \ldots, M$ do
3:    for Sample $i = 1, \ldots, n$ do
4:        Draw $x_{0,i}^{m} \sim \pi_0$
5:    Set initial belief $b_{0,i}^{m} = 1/n$
6: end for
7: end for
8: for Timestep $k = 1, \ldots, K$ do in parallel
9:    for Node $m = 1, \ldots, M$ do
10:       Compute temporal message, $\eta_{x_m \rightarrow x_m}$
11:       Broadcast predicted distribution, $\eta_{x_m \rightarrow h_{\neq m}}$
12:       Collect pseudoranges and inter-vehicle ranges
13: end for
14: for Correction stage $s = 1, \ldots, S$ do in parallel
15:    for Node $m = 1, \ldots, M$ do
16:        Move samples using the KF update, $\hat{x}_m^s$
17:        Receive messages, $\eta_{x_m \rightarrow h_{\neq m}}^s$, from adjacent neighbours
18:        Compute local factor to variable messages, $\eta_{x_m \rightarrow x_m}^s, \eta_{h_{\neq m} \rightarrow x_m}^s$
19:        Compute and broadcast outgoing V2V messages, $\eta_{x_m \rightarrow h_{\neq m}}^s$
20:        Update and broadcast beliefs $b_{m}^s$
21: end for
22: end for
23: end for
3.3 Improved CP Algorithm

3.3.3 Federated SPAWN

The final improved algorithm is the FSPAWN. In this algorithm, the SPAWN algorithm is configured such that the nodes are group into clusters, where one node would act as a central node for processing. Connections between clusters, or nodes between different clusters, are treated like regular SPAWN. The objective of this approach is to alleviate the ineffectiveness of SPAWN to capture the true posterior density. Recall that only marginal posterior is calculated in SPAWN due to its distributed nature. By having clusters within a network, the posterior could be calculated for each cluster, which then should enable better estimation for localisation. Also, FSPAWN is better in handling scalability as the VANET is segregated into clusters. An important note when using this method is that after the creation of cliques, the connections between vehicles need to be segregated into two types; within cliques and in between cliques. This is to avoid using the measurements twice when executing MDPC for nodes within cliques and SPAWN for nodes between cliques. Like the MDPC algorithm, it makes use of the measurement directed and progressive correction methods.

Let \( x = [x_1, \ldots, x_M]' \) denote a collection of nodes, \( cl = [cl_1, \ldots, cl_U]' \) denote a collection of cliques and \( z = [z_1, \ldots, z_T]' \) denote a collection of measurements. Then, let \( m, u, \) and \( t \) denote the indices of nodes, cliques, and measurements respectively. As previously mentioned, the measurements need to be grouped into ones that exist within cliques and the rest that are between cliques, which are denoted as \( z_{t,a} \) and \( z_{t,b} \). In particular, \( z_{t,a} = h(x_{m \in cl_a}, x_{\# \notin cl_a}) \) and \( z_{t,b} = h(x_{m \in cl_a}, x_{\# \notin cl_a}) \). Figure 3.9 is presented to better demonstrate the cliques and the two types of measurements. This figure, which uses simulation 1s configuration, depicts two cliques: nodes 1, 2, and 4 are grouped into the first clique, while node 3 is in the second clique. Then, the \( z_{t,a} \) is represented by the solid blue lines, and \( z_{t,b} \) by the dashed blue lines.

3.3.4 Improved Algorithm Performance Based on 2D Simulation

In this subsection, the improved algorithms are tested using the same simulation configurations discussed in subsection 3.2.4. The performance is first quantified using calculated
RMSE, particularly for set-ups with large prior distribution. Then, the set-ups are compared to each other regarding how fast convergence is achieved. As in the previous tests, MDPC, FSPAWN, and MDSPAWN are run through 50 Monte Carlo realisations to obtain their respective RMSE. The size of cliques for FSPAWN is chosen to be two nodes and four nodes per clique for simulations 1 and 2, respectively. Table 3.2 lists the achieved RMSE for all algorithms. As expected, the improved algorithms produce better RMSE than conventional algorithms do. For simulations 1.1 and 1.2, the RMSE for MDSPAWN improves by 28% and 38%, for FSPAWN by 46% and 59%, and for MDPC by 65% and 80% compared to SPAWN. Furthermore, MDPC achieves an RMSE that is very close, and

<table>
<thead>
<tr>
<th>Alg.</th>
<th>1.1</th>
<th>1.2</th>
<th>2.1</th>
<th>2.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRB</td>
<td>0.31</td>
<td>0.31</td>
<td>2.30</td>
<td>2.30</td>
</tr>
<tr>
<td>WLS</td>
<td>8.89</td>
<td>- 8.32</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>EKF</td>
<td>1.51</td>
<td>3.80</td>
<td>3.81</td>
<td>24.20</td>
</tr>
<tr>
<td>UKF</td>
<td>1.72</td>
<td>1.90</td>
<td>3.16</td>
<td>5.79</td>
</tr>
<tr>
<td>SPAWN</td>
<td>0.82</td>
<td>1.50</td>
<td>3.29</td>
<td>4.14</td>
</tr>
<tr>
<td>MDPC</td>
<td>0.29</td>
<td>0.29</td>
<td>3.01</td>
<td>3.08</td>
</tr>
<tr>
<td>FSPAWN</td>
<td>0.44</td>
<td>0.61</td>
<td>3.10</td>
<td>3.09</td>
</tr>
<tr>
<td>MDSPAWN</td>
<td>0.59</td>
<td>0.92</td>
<td>3.27</td>
<td>3.18</td>
</tr>
</tbody>
</table>

Table 3.2: Averaged RMSE for varying algorithms
3.3 Improved CP Algorithm

Algorithm 4 FSPAWN

1: Select $\gamma_1, \ldots, \gamma_S$
2: for Node $m = 1 \ldots, M$ do
3:   for Sample $i = 1 \ldots, n$ do
4:     Draw $x_m^{0,i} \sim \pi_0$
5:     Set initial weight $w_m^{0,i} = 1/n$
6:     Set initial belief $b_m^{0,i} = 1/n$
7:   end for
8: end for
9: for Timestep $k = 1 \ldots, K$ do
10:   Form cliques $u = 1, \ldots, U$
11:   for Correction stage $s = 1, \ldots, S$ do
12:     for Clique $u = 1, \ldots, U$ do in parallel
13:       for Node $m = 1, \ldots, M$ do in parallel
14:         if $z(x_m^{\in cl_u}, x_m^{\in cl_u})$ then
15:           Perform SPAWN
16:         end if
17:       end for
18:     end for
19:     for Clique $u = 1, \ldots, U$ do in parallel
20:       Update weight $w^{s,i}$ from belief
21:       if $z(x_m^{\in cl_u}, x_m^{\in cl_u})$ then
22:         Perform MDPC for $x \in cl_u$
23:       end if
24:     end for
25: end for
26: Compute parameter estimate $\hat{x} = \sum_{i=1}^{n} w^{S,i} x^{S,i}$
27: end for

in fact slightly better, than the CRB value. This shows that these algorithms are capable of providing better estimates that are closer to the CRB value. For the second simulation, the proposed algorithms are still superior to the conventional algorithms, but the differences are smaller. Unlike in simulation 1.2, in simulation 2.2, MDSPAWN, FSPAWN, and MDPC only manage to improve their RMSE by 23%, 25%, and 26%, respectively. The absolute RMSE for MDPC is about 0.78 m from the CRB value.

This contradiction is investigated and is discovered that the high RMSE relative to the derived CRB value is largely caused by one of the nodes in this particular simulation, as revealed in figure 3.10. It shows that the error for each node is suppressed to better than 2.5 m, except for node 3. By reviewing figure 3.2b, it reveals why this has occurred,
where unlike the other nodes, node 3 is only connected to one other node, which makes estimating for its correct position particularly difficult.

Figures 3.10 and 3.11 reveal the trends of RMSE over time. They show that compared to conventional algorithms, MDPC, FSPAWN, and MDSPAWN are able to provide faster convergence. MDPC can reach convergence as early as the 2 s mark, while FSPAWN and MDSPAWN reach convergence from the 10 and 12 s marks, respectively. This is particularly important when dealing with a high-mobility network, as the frequent changes of vehicles positions make accurate estimation more difficult. This is further discussed in later chapters, as the present chapter only focuses on the performance of CP algorithms in static networks. Overall, the results show improvements of the proposed algorithms over conventional ones. The MDPC, which is a fully centralised system, performs the best as it can capture the posterior density of the whole network, while MDSPAWN is
3.4 Summary

The third best due to its distributed nature, which can only capture the marginal posterior density within the network. The FSPAWN lies somewhere between MDPC and MDSPAWN, as it allows for the calculation of the full posterior density for nodes within cliques and marginals for nodes between cliques, resulting in a trade-off between the fully centralised and decentralised algorithms.

Although the centralised algorithm, MDPC, could provide the best result, it is important to question its feasibility for VANET as it naturally requires a more complex and fully synchronised message passing scheme. For example, in a case with a huge number of vehicles in a vicinity, the question is whether it is feasible for all those vehicles to transport their information to RSUs, which will in turn send the data to a centralised computer, which will then apply the CP algorithm. This becomes more complicated when not all of the vehicles can transmit their information to the RSUs but require other vehicles in between to do so, which results in massive message traffic delays. Hence, in large-scale VANET, the decentralised and semi-decentralised algorithms are more favourable due to their nature of distributed computing. This makes them more capable of handling scalability and reducing message exchanges between vehicles and vehicles, and vehicles and RSUs, hence creating a more efficient network.

3.4 Summary

This chapter has presented the algorithms suitable for CP. The derivation of CRB for CP was discussed first, before proceeding to conventional algorithms including EKF, UKF, PF, and SPAWN. These algorithms were then tested using two 2D simulations with challenging set-ups. The effect of large prior distribution, which simulates positioning in dense urban environments, was investigated. The results showed that not all of the algorithms could achieve RMSE close to the derived CRB, particularly with simulations with large prior distribution. This motivated the development of the novel algorithms MDPC, MDSPAWN, and FSPAWN. The RMSEs for these algorithms were better than for the conventional algorithms. MDPC performed the best, with its achieved RMSE close to the derived CRB, followed by FSPAWN and MDSPAWN. Although the centralised MDPC
provided the best result amongst the proposed algorithms, however, issues such as scalability make it less practical than FSPAWN and MDSPAWN. FSPAWN handles scalability better as its clique sizes become smaller, while MDSPAWN does not face this issue due to its fully decentralised nature. Hence, the FSPAWN and MDSPAWN are seen as better candidates for CP in a VANET. The next chapter discusses the improvements made to the local-level observation, which is the MSS.
Chapter 4
On Improving the Multi-Sensor System

4.1 Introduction

This chapter presents the research conducted to further improve the performance of MSS, which consists of IMU, GNSS, and VSS. It is divided into three sections. The first section presents the technique of IMU/GNSS integration. Then, the second section discusses IMU/GNSS improvement by employing a more accurate IMU stochastic error modelling technique in dynamic environments. Finally, the third section proposes a method of using MSS to improve faulty satellite detection and rejection method, which is useful when navigating in dense urban environments. The second section of this chapter has been published in the Geo-Spatial Information Science journal:


The contents from the published journal article have been slightly altered to better fit the flow of this thesis.

4.2 IMU/GNSS Integration

This section provides preliminary information on IMU/GNSS integration before proceeding to methods of enhancing this integrated system further in the following two
sections. Note that this section focuses on utilising only IMU and GNSS. Hence, the term MSS is not used. The section first presents the system architecture, followed by a discussion on system and measurement models employed in this dissertation.

4.2.1 System Architecture

This subsection discusses both the loosely coupled (LC) and tightly coupled (TC) integration architectures. They primarily differ in terms of the type of GNSS measurements used in the EKF. In this dissertation, the closed-loop correction is employed to ensure that the innovations between subsequent measurement updates are consistently minimised throughout an operation.

Integrated IMU/GNSS using LC architecture utilises GNSS positions as the measurement inputs in the integration algorithm. As depicted in figure 4.1, it is a cascaded architecture, where the GNSS receiver has its own filter (WLS or EKF) and the output is then transferred to the integrated IMU/GNSS filter [44]. The two main advantages of employing this architecture are its simplicity and redundancy when integrating IMU and GNSS. Its simplicity allows it to be implemented with any IMU and GNSS equipments, making
it practically suited for retrofit applications. In addition, this architecture allows monitoring of the stand-alone navigation systems, i.e. IMU or GNSS only solution, usually employed for observability purposes [34, 44]. However, the drawback of this architecture is that it needs GNSS navigation outputs and their corresponding variances from the GNSS filter to be appropriately passed on to the integrated IMU/GNSS EKF. This might disrupt the IMU/GNSS EKF state estimation as the GNSS filter outputs are time correlated whereas the IMU/GNSS EKF assumes that the input measurements are independent between epochs [44]. Another well-known issue is that the architecture needs signals from at least four satellites to maintain the GNSS navigation solution to provide aiding measurements to the IMU [42, 44]. Thus, this architecture does not fully exploit the capabilities of an integrated IMU/GNSS system.

![Tightly coupled schematic](image)

Unlike LC architecture, IMU/GNSS using TC integration utilizes the pseudorange measurements as inputs to the integration algorithm [44]. The benefit of the TC architecture is that it combines the previously two separate filters in LC architecture into a single filter. This effectively removes the problem of transferring variances from one filter to another. Another major advantage of using this architecture is that it does not need a
full GNSS (four satellites) solution to aid IMU. GNSS ranging data is still utilised even if only one satellite signal is available, as shown in [48]. The only disadvantage of the TC architecture is that there is no inherent standalone GNSS solution [44]. However, it is possible to obtain a GNSS-only solution by creating a separate process dedicated only to solving for a GNSS positioning solution.

Both LC and TC architectures are used in this study: LC in the next section, and TC in the last section. LC is used in the next section to exemplify the effects of IMU stochastic errors when implementing an IMU/GNSS integrated system. On the other hand, the reason for using TC in the last section, which discusses faulty satellite signal detection and rejection, is clear: that is, TC allows for all available satellite ranges to be used to aid IMU. The next subsection discusses the system and measurement models for the IMU/GNSS integrated system.

4.2.2 System Model

The system model refers to the information about how a state vector and its covariance propagate forward in time. Before the system model is introduced, it is important to note the difference between the use of a total-state and error-state EKF when implementing IMU/GNSS integration. Total-state EKF refers to the state vector being formed by the actual navigation output including position, velocity, attitude, acceleration, turning rates, and their respective biases. On the other hand, error-state EKF refers to the state vector consisting of the errors of the navigation output. In this dissertation, the error-state EKF is used as it is computationally less expensive and less complex in nature. For example, [34] shows that the amount of computation, measured as floating point operations per second (FLOPS) for the total state, is around 43,000 FLOPS, while the error state has a greatly reduced value of 1,500 FLOPS. The study also shows that there is no need to tune the system noise parameters when using error-state EKF as it is based on the IMU error characteristics, thus reducing the systems complexity. Furthermore, [121] shows that the performance, in terms of positioning output, of these two methods is similar, thus making the error-state method a more favourable choice in the implementation of the IMU/GNSS integrated system.
Three basic error states chosen in the EKF are positioning, velocity, and attitude (PVA) errors. Due to the large error when utilising MEMS IMU, it is also beneficial to include accelerometer and gyroscope static biases and bias drifts. Therefore, these errors are also included in the error-state vector. Altogether, the vector is referred to as the 21-error-state EKF. When employing TC integration architecture, however, two additional states are included: the receiver clock bias and the time scale difference between the GPS and GLONASS time systems, as shown in equation 2.1. The dynamics of the error states are obtained by linearising the IMU mechanisation equations presented in chapter 2. Then the biases are included, modelled as constant random variables for static bias and the Gauss-Markov (GM) model for bias drift. The receiver clock bias and time scale bias are also modelled as constant random variables.

The following lists the system model for the TC integration architecture employed in this study. When LC is used, the receiver clock bias and GPS-GLONASS time scale bias are dropped in the following equations. Note that the full derivations of the equations are extensive, and are therefore not included here. However, they are well documented and can be found in [14, 34, 44]. For convenience, the system dynamic model is stated here:

\[
\dot{x}(t) = F(t)x(t) + G(t)\omega(t) \tag{3.23}
\]

The state vector is formed as

\[
\dot{x}(t) = \begin{bmatrix}
\delta r_{eb}^e \\
\delta v_{eb}^e \\
\delta \psi_{eb}^e \\
\delta b_f \\
\delta b_g \\
\delta b_{df} \\
\delta b_{dg} \\
c\delta t \\
c\delta t_{sys}
\end{bmatrix} \tag{4.1}
\]

where

- \(\delta r_{eb}^e\) Position error
- \(\delta v_{eb}^e\) Velocity error
- \(\delta \psi_{eb}^e\) Orientation error
- \(\delta b_f\) Accelerometer bias
- \(\delta b_g\) Turning rate bias
- \(\delta b_{df}\) Accelerometer bias drift
- \(\delta b_{dg}\) Turning rate bias drift
- \(c\delta t\) Receiver clock bias
- \(c\delta t_{sys}\) GPS-GLONASS time scale bias

The dynamic matrix \(F\) is then obtained by linearising IMU mechanization equations 2.4-
2.9 with respect to the navigation estimates.

\[ F = \begin{bmatrix}
  F_e^1 \\
  F_e^2
\end{bmatrix} \quad (4.2) \]

\[ F_e^1 = \begin{bmatrix}
  0 & I & 0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & -\Omega & 0 & C_b^e & 0 & 0 & 0 \\
  0 & 0 & 0 & -\Omega & C_b^e & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & -1/\tau_{\delta b_d} & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & -1/\tau_{\delta b_r} & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 & -1/\tau_{\delta b_r} & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 & 0 & -1/\tau_{\delta b_r} \\
  0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \quad (4.3) \]

\[ F_e^2 = \begin{bmatrix}
  0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \quad (4.4) \]

where \( F_e^1 = \frac{2\sigma_0}{r_{eb} |r_{eb}|^3} (\hat{r}_{eb})' \), relates to the gravity component [44]. The terms \( \tau_{\delta b_d} \) and \( \tau_{\delta b_r} \) are the correlation time of the dynamic biases. This information is extracted using the AV analysis. The coefficient matrix \( G \) consists of

\[ G = \begin{bmatrix}
  G_1^e \\
  G_2^e
\end{bmatrix} \quad (4.5) \]

\[ G_1^e = \begin{bmatrix}
  0 & 0 & 0 & 0 & 0 & 0 \\
  0 & C_b^e & 0 & 0 & 0 & 0 \\
  0 & 0 & C_b^e & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \quad (4.6) \]

\[ G_2^e = \begin{bmatrix}
  0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & I & 0 & 0 & 0 \\
  0 & 0 & 0 & I & 0 & 0 \\
  0 & 0 & 0 & 0 & I & 0 \\
  0 & 0 & 0 & 0 & 0 & I
\end{bmatrix} \quad (4.7) \]

whereas \( \omega \) consists of the PSD of IMU measurement noise, receiver clock bias, and the
4.2 IMU/GNSS Integration

The time scale bias,

\[
Q = \begin{bmatrix}
    n_{V_{RW}}^2 & 0 & 0 & 0 & 0 & 0 \\
    0 & n_{ARW}^2 & 0 & 0 & 0 & 0 \\
    0 & 0 & n_{bdf}^2 & 0 & 0 & 0 \\
    0 & 0 & 0 & n_{bdg}^2 & 0 & 0 \\
    0 & 0 & 0 & 0 & n_{c_idf}^2 & 0 \\
    0 & 0 & 0 & 0 & 0 & n_{c_idg}^2
\end{bmatrix}
\]  

(4.8)

where the subscript \( V/ARW \) stands for velocity and angle random walk. These values are discussed in detail in the next section.

4.2.3 Measurement Model

This subsection briefly presents the GNSS measurement model and its respective partial derivatives, and discusses its implementation for both LC and TC system architecture. The general measurement model is restated here for convenience.

\[
z = h(x) + n
\]  

(3.1)

**LC GNSS Measurement Model**

In the LC architecture, the measurement is the GNSS positioning output

\[
z = \begin{bmatrix} r_{GNSS}^2 \end{bmatrix}
\]  

(4.9)

It is clear to see that the measurement model is linear, hence the \( H \) matrix is

\[
H = \begin{bmatrix} \mathbf{1} & 0_{3 \times 18} \end{bmatrix}
\]  

(4.10)

and the measurement noise is the GNSS position variance

\[
R = \begin{bmatrix} \sigma_{r_{GNSS}}^2 \end{bmatrix}
\]  

(4.11)
TC GNSS Measurement Model

As discussed in chapter 2, the GNSS pseudorange measurement is

$$ z = \rho(x, p) + c\delta t + c\delta t_{sys} + \nu $$

Note that the ephemeris, satellite clock, atmospheric and multipath error terms have been dropped as they are solved or minimised using techniques discussed in chapter 2. The Jacobian matrix $H$ is then

$$
H = \begin{bmatrix}
H_{GPS_m} \\
H_{GLO_n}
\end{bmatrix} = \begin{bmatrix}
r_{s_m,x} - r_{x_x} & r_{s_m,y} - r_{x_y} & r_{s_m,z} - r_{x_z} & 0_{m,18} & 1 & 0 \\
q_m & q_m & q_m & 0_{n,18} & 1 & 1
\end{bmatrix}
$$

where $r_{s_m,x/y/z}$ denotes the $m^{th}$ GPS satellite position in the $x, y, z$ axes. Similarly, $r_{s_n,x/y/z}$ denotes the $n^{th}$ GLONASS satellite position and $r_{x_x/y/z}$ denotes the position estimate. $q_{m/n}$ is the geometric range to the satellite. Finally, the measurement noise matrix consists of the pseudorange weighting factor, modelled as a function of the satellites elevation angle [101].

$$ R = \left( \frac{1}{\sin \alpha_{s_m/n}} \right)^2 $$

Having defined the IMU/GNSS integration method, the next section presents an improvement in deriving IMU stochastic error parameters.

### 4.3 Improving Stochastic Analysis for IMU

As discussed in chapter 2, IMU errors can be categorised into two groups: deterministic and stochastic errors. Bias, scale factor, and non-linearity are the major contributors of deterministic errors, and they can be removed by lab calibration procedures. Stochastic errors, however, are not directly obtainable. Therefore, it is in the interest of analysts to model or characterise stochastic errors and include the models in the integration algorithm to effectively remove noisy measurements. In trying to determine stochastic errors, several techniques have been developed. The traditional method of using a single
RMS value with associated correlation time proved to be inadequate to accurately characterise sensors performance. Therefore, alternative techniques using frequency-domain and time-domain approaches have been introduced in trying to determine different types of IMU stochastic errors [58].

Power spectral density (PSD) is an example of a frequency-domain approach. PSD can be calculated in a number of different ways, but it is generally defined as the Fourier transform of its autocorrelation function. It is a powerful tool for analysing and modelling stochastic processes but difficult for non-system analysts to comprehend. On the other hand, correlation methods enable analysis using the time-domain approach. An example of a correlation method is the autocorrelation function, which measures the dependence of the process value at one time with its value at other times. Another example of a correlation method is the autoregressive moving average process (ARMA), which relates the auto-covariance sequence to coefficients of a difference equation. The shortcoming of the correlation methods is that they require extensive data collection. Furthermore, correlation approaches are also reported to be model-sensitive and poorly suited to dealing with odd power law processes, higher-order processes, or a wide dynamic range [58].

The Allan variance (AV) is another example of a time-domain approach. It was developed in the mid-1960s for the characterisation of the phase and frequency of precision clocks and oscillators [58]. Recent studies [30, 31, 46, 47, 55, 129] have applied this technique to identify and model stochastic errors that exist in IMU due to the techniques simplicity and efficiency compared to PSD and autocorrelation methods. Using AV allows the identification of five stochastic errors that are inherent in an IMU: quantisation noise, angle/velocity random walk, bias instability, rate random walk, and rate ramp. The identified errors are then analysed to extract their coefficients, which are then used in the IMU/GNSS integration algorithm. In the aforementioned studies, the AV is applied to IMU static data, collected in a controlled laboratory environment. However, [123] has shown that IMU error varies depending on its motion and other environmental effects such as vibration. Thus, it is important that these effects be accounted for when analysing IMU data, and neglecting these effects will result in a non-optimal integrated
Due to the motion and environmental effects on IMU measurements, the AV is no longer suitable for identifying IMU errors, as it assumes that the measurements analysed are \textit{stationary}. Here, the term \textit{stationary} means that the measurement noise signals are time invariant [61]. To overcome this problem, the present study applies the dynamic Allan variance (DAV) to the IMU measurement errors during motion. The DAV is a modified version of AV which allows for a time-varying AV analysis [36–38]. This is useful as it enables IMU error detection and characterisation from a kinematic test. The DAV has mainly been used for clock characterisation testing, particularly in GNSS satellites. For example, the DAV was used for the clock characterisation on-board the first experimental Galileo satellites. The U.S. Naval Research Laboratory also uses the DAV to monitor and test GNSS clocks [38].

This section presents the results of IMU stochastic error detection during the kinematic test and its implications for IMU/GNSS integrated system performance. Firstly, the DAV methodology is presented. To maintain brevity, the AV is not outlined here, but it can be found in [31, 129]. Then, static and kinematic experiment set-ups are briefly presented, along with the sensors used during these experiments. The results section discusses the extracted error parameters using AV and DAV and their effects on the IMU/GNSS integrated systems performance.

\subsection{Dynamic Allan Variance (DAV)}

The DAV is an extension of the AV which can reveal any time-varying properties of a set of signals. The basic concept of DAV is that it repeats AV evaluation at different epochs which can then be visualised in a 3D graph [38]. This reveals any non-stationariness in noise behaviour over time, which is not possible using AV [61]. The DAV procedure can be summarised as follows [38]:

\begin{enumerate}
  \item Set the analysis epoch \( t = t_1 \)
  \item Set the desired analysis window \( N_w \)
  \item Evaluate the AV \( \sigma(t_1, N_w) \)
\end{enumerate}
iv. Set another analysis epoch and repeat from i. The new epoch can be set so that the data within its corresponding new analysis window overlaps with the previous epoch and analysis window.

More details on DAV can be found in [38]. The formulation for the discrete time DAV is:

$$ \sigma^2(i, n) = \frac{1}{2\tau^2(N_w - 2n)} \sum_{k=i-N_w/2}^{i+N_w/2-2n-1} \cdots [\theta_{k+2n} - 2\theta_{k+n} + \theta_k] $$  (4.14)

where $i$ denotes the analysis epoch and $N_w$ denotes the analysis window.

### 4.3.2 IMU Stochastic Noise Terms

The five main IMU error sources that can be detected using AV are the following.

**Quantisation noise** is a result of the process of encoding a signal from analogue to digital form. The process of signal encoding is limited by the systems bit resolution, which therefore results in error when trying to sample the actual amplitudes. The small differences between the actual amplitude and digital readout are the quantisation noise [53]. In an AV log-log graph, this noise is represented by a slope of -1 and the coefficient value can be read from the slope line at $\tau = \sqrt{3}$. The units for quantisation noise are $m/s$ and rad (radians).

**Angle/velocity random walk (A/VRW)** describes the deviation that occurs when signals are integrated. The origin of the noise comes is that signals from IMU are perturbed by a white noise sequence, which is caused by thermo mechanical noise fluctuating at rates greater than the actual samples. Integrating the white noise consequently results in zero mean A/VRW. Identifying and modelling A/VRW is essential as it can be a major source of error in IMU. It is represented by a line with a slope of -0.5, and the coefficient value can be read at the slope line $\tau = 1$. The units for A/VRW are $m/s/\sqrt{s}$ for velocity and rad/\sqrt{s} for angle [55, 107].
Bias Instability describes how bias changes over time due to random electronic flickering. It is usually observed at low frequencies and is associated with a correlation time. It is represented by a line with a slope of 0 and its coefficient value is at the lowest point of the AV curve. The units for this error are \( m/s^2 \) and \( \text{rad}/\sqrt{s} \) [55, 106].

Rate random walk (RRW) is of uncertain origin. It is possibly a limiting case of an exponentially correlated noise with a long correlation time. RRW is represented by a line with a slope of +0.5 and the coefficient value can be read at \( \tau = 3 \). Its units are \( m/s/\sqrt{3}\tau \) and \( \text{rad}/\sqrt{3}\tau \) [55].

Rate Ramp reveals any deterministic error that is present in the dataset. It is represented by a line with a slope of +1 and the coefficient value at \( \tau = \sqrt{2} \) [55].

<table>
<thead>
<tr>
<th>Noise Terms</th>
<th>Slope</th>
<th>Allan Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantisation (Q)</td>
<td>-1</td>
<td>( \sigma_Q(\tau) = Q\sqrt{3}/\tau )</td>
</tr>
<tr>
<td>Random Walk (N)</td>
<td>-0.5</td>
<td>( \sigma_N(\tau) = N/\sqrt{\tau} )</td>
</tr>
<tr>
<td>Bias Instability (B)</td>
<td>0</td>
<td>( \sigma_B(\tau) = 0.664B )</td>
</tr>
<tr>
<td>Rate Random Walk (K)</td>
<td>+0.5</td>
<td>( \sigma_K(\tau) = K\sqrt{\tau/3} )</td>
</tr>
<tr>
<td>Rate Ramp (R)</td>
<td>+1</td>
<td>( \sigma_R(\tau) = R\tau/\sqrt{2} )</td>
</tr>
</tbody>
</table>

Table 4.1: IMU stochastic Noise Terms

4.3.3 Experiment Set-Up

This subsection details the experiment set-ups for both static and kinematic tests. In both tests, the IMU used to validate this study is the Xsens MtG. It is an MEMS-based IMU/GNSS navigational device consisting of tri-axial accelerometers, gyroscopes and magnetometers, a GNSS receiver, and an on-board processing unit. Its GNSS receiver can make use of up to 50 channels of the GNSS C/A code on the L1 frequency and tracking up to 4 Hz, while its on-board processing unit is able to fuse the IMU and GNSS data in real time.
Static Laboratory Test

For the static test, two hours worth of stationary data was collected, taking into account several environmental factors. Firstly, the room temperature was ensured to be constant throughout the whole test, as it plays a critical role in affecting the IMUs performance [84]. Secondly, the test was conducted at night to avoid any possibility of unwanted vibrations.

Kinematic Test

The kinematic test was conducted in Nottingham, UK as part of a multi-sensor data collection campaign conducted by joint FIG working group 5.5 and IAG working group 4.1.1 [64]. The particular test used in this study was conducted on road sections of the A52 Clifton Blvd., near the University of Nottingham campus. In total, the test lasted more than two hours, but only half an hours worth of data is used here. Three main sensors were used in this test: a navigational-grade IMU (Honeywell CIMU), a geodetic-grade GNSS receiver (Leica GS10), and the MEMS-based IMU (Xsens MTi-G). The navigational-grade IMU, integrated with a dual-frequency GNSS solution, is treated as the reference data as it provides a high-accuracy navigation solution. The CIMU/GNSS navigation solution (position, velocity, and attitude) is then used to derive body acceleration and turning rates, which are subsequently utilised to compute Xsens MTi-G measurement residuals.

<table>
<thead>
<tr>
<th>Performance</th>
<th>Units</th>
<th>CIMU</th>
<th>Xsens MTi-G</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acc Noise</td>
<td>m/s/√Hz</td>
<td>N/A</td>
<td>2 × 10⁻³</td>
</tr>
<tr>
<td>Acc Bias</td>
<td>m/s²</td>
<td>3 × 10⁻⁶</td>
<td>2 × 10⁻²</td>
</tr>
<tr>
<td>Gyro Noise</td>
<td>o/s/√Hz</td>
<td>1 × 10⁻⁷</td>
<td>5 × 10⁻²</td>
</tr>
<tr>
<td>Gyro Bias</td>
<td>o/s</td>
<td>1 × 10⁻⁷</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 4.2: Specifications of CIMU and Xsens MTi-G IMU
4.3.4 Results

This subsection comprises two main parts. The first presents the extracted IMU stochastic error coefficients using both AV and DAV. The second compares the results of the IMU/GNSS performance using the extracted coefficients from AV and DAV.

IMU Stochastic Errors

Allan variance The AV is applied on the IMU output to determine the coefficients of various stochastic processes. It enables the extraction of five main stochastic errors that are typically inherent in an IMU. The error coefficients are extracted by analysing the log-log graph (Allan deviation $\sigma_{AV}$ vs. time sampling $\tau$), where the underlying stochastic processes can be identified by their slopes at different $\tau$. For example, quantisation noise has a slope of -1, whereas A/VRW has a slope of -0.5. Refer to table 4.1 to identify errors with associated slope on a log-log graph.

![Figure 4.3: Allan deviation plot of static accelerations](image)

Figures 4.3 and 4.4 show the resulting $\sigma_{AV}$ vs. $\tau$, both in log-log graphs. From these graphs, it is clear that the most dominant error is the A/VRW. For accelerations, the lines with -0.5 slope ranges from approximately $\tau$ 0.05 to 2, while turning rates range from $\tau$ 0.02 to 10. Bias instability can be seen in the $\tau$ 300-550 and 130-530 regions for acceleration and turning rates, respectively, and their coefficients are observed at the lowest point of the AV curves. Table 4.3 lists the extracted A/VRW coefficients while table 4.4 lists the
4.3 Improving Stochastic Analysis for IMU

Figure 4.4: Allan deviation plot of static turning rates

bias instability coefficients and the corresponding time correlations.

<table>
<thead>
<tr>
<th>Axis</th>
<th>Velocity RW</th>
<th>Angle RW</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>9.39E-04</td>
<td>6.57E-04</td>
</tr>
<tr>
<td>Y</td>
<td>9.33E-04</td>
<td>6.42E-04</td>
</tr>
<tr>
<td>Z</td>
<td>9.55E-04</td>
<td>6.20E-04</td>
</tr>
</tbody>
</table>

Table 4.3: IMU A/VRW coefficients

<table>
<thead>
<tr>
<th>Axis</th>
<th>Acc and Gyro Bias Instability</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>m/s^2</td>
</tr>
<tr>
<td>X</td>
<td>5.35E-04</td>
</tr>
<tr>
<td>Y</td>
<td>5.03E-04</td>
</tr>
<tr>
<td>Z</td>
<td>2.14E-04</td>
</tr>
</tbody>
</table>

Table 4.4: IMU bias instability coefficients

**Dynamic Allan Variance (DAV)** Unlike the AV, the DAV (eq. 4.14) allows the calculation of multiple Allan deviations from a single dataset, achieved by varying the analysis window \((N_w)\) and desired steps \((i)\). In this analysis, \(N_w\) is set to 60 seconds and \(i\) is set to execute every 10 seconds. It is noted that due to the short \(N_w\) window, it is expected that only quantisation and A/VRW will be visible in the DAV graph, as other processes such as bias instability require a longer \(N_w\) window. For example, in table 4.4, acceleration bias
instability is only seen in the $\tau$ of 300-550 region. The DAV is represented in a 3D log-log graph, where in addition to the $\sigma$ vs. $\tau$, it plots the time when the DAV is applied.

The DAV is firstly applied to the static data to check for its correctness. It is expected that the resulting stochastic coefficients will be similar to the ones obtained from AV as the same static dataset is used. Figure 4.5 shows the resulting DAV for acceleration X, while table 4.5 lists the averaged (taken from five samples) A/VRW for all axes. From figure 4.5, it is observed that there is minimal variation between calculated Allan deviations at different times. This is expected as the amplitude of white noise, which results in A/VRW when integrated, remains constant throughout the static dataset. Hence, the resulting A/VRWs coefficients will also be similar. From table 4.5, it is observed that the extracted A/VRW coefficients are very close to the ones obtained using AV, listed in table 4.3 which proves the correctness of the DAV analysis. The average difference between the coefficients obtained from AV and DAV is only 2.43E-05.

![Figure 4.5: DAV plot of kinematic acceleration X](image)

The DAV is also applied to the kinematic data residuals. The residuals ($\Delta$) are obtained from subtracting the accelerations and turning rates derived from CIMU/GNSS from the ones measured by Xsens Mti-G. The CIMU/GNSS accelerations and turning rates are derived from the velocity and attitude and heading reference system, differentiated with respect to time. These values are then rotated from the navigational frame to
4.3 Improving Stochastic Analysis for IMU

<table>
<thead>
<tr>
<th>Velocity RW</th>
<th>Angle RW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Axis</td>
<td>m/s/√s</td>
</tr>
<tr>
<td>X</td>
<td>1.053E-03</td>
</tr>
<tr>
<td>Y</td>
<td>9.816E-04</td>
</tr>
</tbody>
</table>

Table 4.5: A/VRW coefficients from DAV static

the body frame using its attitude and heading information. Figures 4.6 and 4.7 show the residuals between acceleration and turning rates derived from CIMU/GNSS and XsensMTi-G. From these two graphs, it is observed that the acceleration residuals vary much more than the turning rate residuals do. Furthermore, the regions that contain higher variations are observed when the vehicle is moving, indicating that there are motion dependency errors. It is also worth noting that the measured accelerations during the kinematic test contain higher noise amplitude, resulting from vibrations compared to the measured accelerations in the laboratory test. These effects are expected to be visible in the DAV graph.

![Figure 4.6: Δ Accelerations: CIMU/GNSS vs. MEMS XsensMTi](image)

Figures 4.8 and 4.9 show the DAV results for acceleration X and turning rate Z during the kinematic test. From figure 4.8, it is observed that the $\sigma$ varies greatly, reflecting the effects of motion-dependent errors, also seen in figure 4.6. Furthermore, when observing the lower peaks of the assembled $\sigma$, which reflect the stationary part of the kinematic test, the $\tau$ is higher than the ones in figure 4.3 which reflects the vibration effect. On the other
hand, only minimal variations of $\sigma$ are observed in figure 4.9. This suggests that unlike measured accelerations, the turning rates are not affected by environmental changes. In fact, when looking closely at the calculated $\tau$ values, they are almost identical to the ones in figure 4.4. Again, this suggests that the measured turning rates are not affected by vehicle motions and vibrations.

Table 4.6 lists the A/VRW coefficients from the kinematic test. The minimum column lists the averaged A/VRW obtained from the lower peaks while the maximum lists the averaged A/VRW obtained from the higher peaks of the calculated DAV. It is clear from
the results that the values for maximum are higher than those for minimum. On average, the VRW increased by \(-7.717\times10^{-3} \, m/s/\sqrt{s}\), which is almost a tenfold increase, whereas the ARW only increased by \(-1.587\times10^{-5} \, rad/\sqrt{s}\).

<table>
<thead>
<tr>
<th>Units</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>m/s/\sqrt{s}</td>
<td>rad/\sqrt{s}</td>
</tr>
<tr>
<td>Y</td>
<td>1.468E-03</td>
<td>6.505E-04</td>
</tr>
<tr>
<td>Z</td>
<td>2.689E-03</td>
<td>7.535E-04</td>
</tr>
</tbody>
</table>

Table 4.6: IMU A/VRW coefficients from the DAVAR kinematic test

The results obtained in subsections 4.3.4 and 4.3.4 show that the extracted A/VRW values vary depending on the test environment and vehicle motions. Thus, it is important that these effects are accounted for in the integration algorithm to obtain better IMU/GNSS navigation solutions. The next subsection presents a simple analysis of the IMU/GNSS navigation performance when these variations are accounted for in the integration algorithm.
IMU/GNSS Navigation Performance: AV vs. DAV

This subsection presents a simple analysis of the IMU/GNSS navigation performance when the A/VRW variations are accounted for in the integration algorithm. Here, the navigational-grade CIMU/GNSS integrated system is used as the benchmark to test the MEMS Xsens MTi-G/GNSS integrated system. The code-based GNSS or single point positioning (SPP) solutions are also included here as a navigation performance comparison. The Xsens/GNSS integrated system uses the LC EKF integration algorithm to fuse the IMU and GNSS data; details can be found in [34,44,47].

The analysis is categorised into two scenarios. The first scenario, called AV-EKF, uses the A/VRW and bias instability from the AV applied to the static laboratory dataset, while the second scenario, DAV-EKF, uses the maximum A/VRW extracted from the DAV applied to the kinematic dataset. The bias instabilities for both AV-EKF and DAV-EKF remain identical, as it is not possible to extract its coefficients from DAV due to its short time window \((N_{w})\). The analysis evaluates the accuracy of both scenarios using the RMSE of their respective positioning errors (figure 4.10). The analysis presents the results in two stages: first, the IMU/GNSS integration with full GNSS availability; and later, with four simulated GNSS outages, each lasting 30 seconds.

![Figure 4.10: IMU/GNSS positioning error](image)

**Full GNSS availability** Table 4.7 lists the RMSE of GNSS-SPP and IMU/ GNSS using AV-EKF and DAV-EKF. It is observed that the highest 2D RMSE is from GNSS-SPP.
4.3 Improving Stochastic Analysis for IMU

<table>
<thead>
<tr>
<th></th>
<th>2D (m)</th>
<th>3D (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GNSS-SPP</td>
<td>2.25</td>
<td>10.47</td>
</tr>
<tr>
<td>AV-EKF</td>
<td>1.83</td>
<td>10.62</td>
</tr>
<tr>
<td>DAV-EKF</td>
<td>1.79</td>
<td>10.22</td>
</tr>
</tbody>
</table>

Table 4.7: Error-RMS comparison

On the other hand, the IMU/GNSS using noise coefficients from AV applied on static data has a lower RMSE, with an improvement of 0.42 m. Its performance marginally improves when noise coefficients from DAV applied on kinematic data are used, with a further decrease of 0.04 m compared to AV. As with 2D RMSE, the 3D RMSE is the highest from GNSS-SPP. Interestingly, the RMSE increases slightly for IMU/GNSS AV-EKF. This might be caused by a number of factors, such as over-confidence in the IMU measurements (lower A/VRW values) resulting in incorrect estimation of the navigation solution. However, the RMSE decreases for IMU/GNSS DAV-EKF with an improvement of 0.24 m compared to GNSS-SPP, and an improvement of 0.39 m compared to AV-EKF.

**Simulated GNSS outages** Four 30-second GNSS outages are simulated to further assess the improvement of DAV over AV in the IMU/GNSS integration.

![Figure 4.11: AV-EKF vs. DAV-EKF 2D performance](image)

Figures 4.11 and 4.12 show the 2D and 3D errors of the IMU/GNSS. As expected, its navigation solution drifts unboundedly over time without the aiding of GNSS. From the
figures, it is observed that the DAV-EKF results in smaller 2D and 3D errors compared to AV-EKF. This is because the DAV is able to provide a more realistic estimation of IMU stochastic errors, thus allowing the EKF to better estimate IMU biases. As seen in tables 4.8 and 4.9, the improvement ranges from 1.8 m to 4.6 m, and from 1.47 m to 8.85 m for 2D and 3D errors, respectively. On average, the 2D and 3D positional errors improve by 3.30 m and 5.74 m, respectively. These values represent 16% and 18% improvements of DAV over AV.

<table>
<thead>
<tr>
<th>2D Max Error (m)</th>
<th>Out 1</th>
<th>Out 2</th>
<th>Out 3</th>
<th>Out 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>AV-EKF</td>
<td>20.58</td>
<td>10.71</td>
<td>21.31</td>
<td>32.09</td>
</tr>
<tr>
<td>DAV-EKF</td>
<td>17.74</td>
<td>6.10</td>
<td>17.41</td>
<td>30.22</td>
</tr>
<tr>
<td>Improvement</td>
<td>2.84</td>
<td>4.60</td>
<td>3.90</td>
<td>1.87</td>
</tr>
</tbody>
</table>

Table 4.8: 2D maximum error during GNSS outage

<table>
<thead>
<tr>
<th>3D Max Error (m)</th>
<th>Out 1</th>
<th>Out 2</th>
<th>Out 3</th>
<th>Out 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>AV-EKF</td>
<td>28.73</td>
<td>24.18</td>
<td>32.77</td>
<td>38.71</td>
</tr>
<tr>
<td>DAV-EKF</td>
<td>23.76</td>
<td>16.48</td>
<td>23.92</td>
<td>37.24</td>
</tr>
<tr>
<td>Improvement</td>
<td>4.97</td>
<td>7.70</td>
<td>8.85</td>
<td>1.47</td>
</tr>
</tbody>
</table>

Table 4.9: 3D maximum error during GNSS outage
4.4 Improving Faulty Satellite Detection and Rejection

This section proposes a technique to improve the conventional faulty satellite detection and rejection method. As discussed in chapter 2, positioning accuracy is severely affected by faulty signals due to multipath when navigating in dense urban environments. Hence, many studies, such as in [50, 67, 87, 102], have been conducted to try to correctly identify and subsequently remove these faulty signals. When conventional method of integrity monitoring is used on a single GNSS receiver, it is commonly known as receiver user autonomous integrity monitoring (RAIM), while in an integrated system, it is more commonly referred to as user autonomous integrity monitoring (UAIM). Several other studies have also termed this technique fault detection and exclusion (FDE) [67], which is the term chosen in this thesis. This section is structured as follows. The first part presents conventional FDE methods using WLS and EKF. Then, the proposed method to further improve this technique is presented, and the inclusion of VSS, the application of the non-holonomic constraint, and the extended FDE method are discussed. The final part then covers the results of the proposed technique tested with simulated multipath on a real dataset.

4.4.1 Conventional FDE Methods

This subsection first discusses the most common form of FDE, which uses WLS. This is followed by the second form of FDE, which is used when additional sensors, such as IMU, are available. This form of FDE uses the EKF to monitor satellite signal integrity and subsequently detect and remove faulty signals.

FDE Using WLS

The most widely employed form of FDE uses pseudorange innovation from WLS [44, 70, 97]. This approach requires a minimum number of five satellites to detect faults due to the need to estimate four parameters of position \( r_{x,y,z} \) and clock bias \( c\delta t \) when using GPS. Subsequently, at least six satellites are needed to correctly exclude any faulty signals. The
measurements and their variances can be written as

\[ z = [\hat{\rho} - \hat{\rho}] \] (4.15)

where \( \hat{\rho} \) denotes the measured while \( \hat{\rho} \) denotes the computed pseudoranges. Their variances are then

\[ V_z = V_\rho \] (4.16)

The optimal estimates for the state parameters \( \hat{x} \) and their variances are given by

\[ \hat{x} = (H'V_z^{-1}V)^{-1}H'V_z^{-1}z \] (4.17)
\[ V_{\hat{x}} = (H'V_z^{-1}H)^{-1} \] (4.18)

Subsequently, the innovations and their variances are then defined as

\[ v = z - H\hat{x} \] (4.19)
\[ V_v = V_z - HV_{\hat{x}}H' \] (4.20)

then tested using

\[ C = v'V_z^{-1}v \] (4.21)

which follows a chi-square distribution, with \( k \) degrees of freedom. The expected value of a chi-square random variable is its degrees of freedom. Hence, this can be compared to the value of \( C \); if it is larger than the expected value according to the chi-square distribution, then faulty measurements are detected. Assuming the measurements are modelled correctly and weighted appropriately, then one or more pseudoranges must be (an) outlier(s). Subsequently, once this statistics test has failed, the exclusion stage can take place. The simplest exclusion method is to normalise the innovation using

\[ v_t = \frac{v_t}{\sigma_{t,\rho}} \] (4.22)
where subscript \( t \) denotes the index of the measurement. The normalised innovation follows the unit normal distribution \( N(0, 1) \). Then, any residual that falls out of the confidence interval can be assumed to be an outlier, and subsequently excluded from the computation. Note, however, that only the largest residual is excluded.

**FDE Using EKF**

The inclusion of IMU can effectively provide more redundancy. By including IMU, only two satellites are needed to detect outliers and three to exclude any existing outliers [51, 56, 114]. FDE when using IMU/GNSS can have two forms. The first variant, discussed in [109] and [15], is a straightforward approach, where the EKF innovation is directly used for the statistics test. The second, as demonstrated in [51], is where the WLS principle is used and the EKF predicted estimates are stacked together with the pseudorange measurements. Both variants are considered here. For the first variant, the innovation is defined as

\[
v_k = z_k - h(\hat{x}_k)
\]  

The variance of the innovation comprises the sum of the measurement noise variance \( R \) and the variance of the state estimate \( P \) in the measurement space

\[
S_k = H_k P_k H_k^T + R_k
\]

Then, as in equation 4.22, the normalised innovation is defined as

\[
y_{k,t} = \frac{v_{k,t}}{\sqrt{S_{k,t,t}}}
\]

which again follows the unit normal distribution. Thus, any residuals that are not within the confidence interval are treated as outliers. The drawback of this method is that the detection of outliers is largely dependent on the innovation variance. Hence, when the variance is large, the system can often miss the presence of outliers.

In the second variant, the pseudoranges and position estimate are stacked together, both treated as measurements. Hence, the measurements and their variances can be re-
written as
\[
z_k = \begin{bmatrix}
\hat{\rho} \\
r_x
\end{bmatrix}
\] (4.26)

\[
V_z = \begin{bmatrix}
V_{\rho} & 0 \\
0 & P_r
\end{bmatrix}
\] (4.27)

\[
H = \begin{bmatrix}
H_{\rho} \\
H_r
\end{bmatrix}
\] (4.28)

where
\[
H_{\xi} = \begin{bmatrix}
I_{3 \times 3} & 0_{3 \times 1}
\end{bmatrix}
\] (4.29)

Subsequently, the FDE procedure using equations 4.19-4.22 can be conducted. While this variant provides better detection of outliers, it requires signals from at least two satellites to successfully execute FDE, thus, limiting its effectiveness in dense urban environments. Similar to the first variant, it too can wrongly identify faulty signals when the innovation variance is large. The following subsection provides details about the proposed method, which imposes an additional sensor and conditions on the conventional FDE process to overcome this problem.

### 4.4.2 Improved FDE Method

In this study, the satellite rejection method is improved for use in situations where there is limited satellite visibility and satellite signals are affected by multipath, which is typical in dense urban environments. The approach taken varies from that in previous studies discussed in the previous subsection: here, the VSS and the non-holonomic constraint technique are employed to further enhance FDE. This section first discusses the enhanced form of FDE, called extended FDE, followed by the VSS measurement and non-holonomic constraint method.
Extended Fault Detection and Rejection Method

The extended FDE imposes a simple additional condition on top of the conventional FDE. This additional step is executed by comparing the magnitude of the computed innovation with its expected value (threshold), which is empirically defined. Typically, this value is somewhere between 5 and 10 m. If the magnitude is larger than the threshold value, it is assumed that there are outliers. Then, the largest outlier is removed and the test is repeated iteratively. Thus, this technique allows for the detection of possible outliers even when there is only one satellite.

![Figure 4.13: Overall process of the extended FDE](image)

However, this technique comes with a caveat: it assumes that the IMU can provide highly reliable positioning output. Thus, when employing only low-cost IMU/GNSS, this technique might not be suitable as the resulting trajectory might wander off the true path, subsequently rejecting signals that are not actually faulty. Therefore, this study also employs the VSS, thus forming the MSS, and the non-holonomic constraint to enhance the output from MSS, thereby allowing the use of this technique. To further minimise the risk
of detecting the wrong faulty signals, an independent FDE, based only on pseudoranges, is also included as part of the proposed method. However, it is clear that this is only possible when enough redundancy is available.

The overall process of the extended FDE is shown in figure 4.13. The blue boxes in this figure represent the additional hardware and processes that together form the extended FDE. Furthermore, in the statistical test step, two FDEs run in parallel; the first is based solely on pseudoranges, whereas the second is based on the MSS. As discussed earlier, the FDE based on only pseudoranges is used to prevent the FDE based on MSS from incorrectly rejecting signals that are not faulty. The following subsections discuss the VSS measurements and non-holonomic constraint method.

**Vehicle Speed Sensor Measurement Model**

VSS and its measurement characteristics were already reviewed in section 2.5. Therefore, only its measurement model, variance, and subsequently partial derivatives are presented here. The VSS measurement model can be expressed as

\[ z_{VSS} = \left( \hat{v}_x^2 + \hat{v}_y^2 + \hat{v}_z^2 \right)^{0.5} + \nu_{VSS} \]  

(4.30)

where \( \hat{v} \) represents the computed velocity in the ECEF frame and \( \nu \) represents the VSS noise. As discussed in section 2.5, the value of the noise for the particular VSS sensor used in this study is increased from \( \sigma = 0.15 \) to \( \sigma = 0.25 \) m/s to compensate for the quantisation error. The partial derivative relating to this measurement is then formed as

\[ H = \begin{bmatrix} 0_{1 \times 3} & \frac{\nu_x}{\hat{v}} & \frac{\nu_y}{\hat{v}} & \frac{\nu_z}{\hat{v}} & 0_{1 \times 17} \end{bmatrix} \]  

(4.31)

where \( \hat{\nu} = \left( \hat{\nu}_x^2 + \hat{\nu}_y^2 + \hat{\nu}_z^2 \right)^{0.5} \)

**Non-Holonomic Constraint**

Non-holonomic constraint is a technique that assumes no slide slip effects in land-based navigation; hence, the velocity in the longitudinal direction is assumed to be zero. This
is usually true for land based vehicles as rarely does a vehicle experience side slips or drifts while travelling along a road, hence making the non-holonomic constraint a useful approach in reducing positioning error. Furthermore, it is also assumed that height does not change drastically over the trajectory; hence, the change in height is assumed to be close to zero. This technique has been employed widely in enhancing the IMU/GNSS integrated system, as shown in \([2, 43, 77]\). To successfully employ this technique, the computed velocity, which is resolved in the ECEF frame, needs to be rotated to the IMU body frame, using
\[
\hat{v}^b = C^b_e \hat{v}^e
\]

As previously mentioned, the fictitious velocities in the longitudinal and vertical directions are assumed to be zero. Thus, the measurement is formed as
\[
z_{nhc} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \nu_{nhc}
\]
where \(\nu_{nhc}\) is the noise of the assumed velocities. This value reflects the error resulting from the misalignment of the IMU body frame to the vehicles body frame, which varies according to the vehicles speed. For example, when the calculated speed in the IMU body frame is 20 m/s and the misalignment is 3\(^\circ\), the resulting error is around 1 m/s, using \(v^b \sin \theta\). It is clear that as the velocity increases, so too does the error. In the experiment described in a later subsection, it is assumed that the misalignment is 3\(^\circ\). Hence, the error value is conditioned by the calculated velocity in the IMU body frame. Finally, the partial derivative matrix relating to the non-holonomic constraint is formed as
\[
H = \begin{bmatrix}
0_{1 \times 3} & C_{1,1:3} & C_{1,2}v_x^e - C_{1,3}v_y^e & C_{1,3}v_y^e - C_{1,1}v_x^e & C_{1,1}v_y^e - C_{1,2}v_x^e & 0_{1 \times 14} \\
0_{1 \times 3} & C_{3,1:3} & C_{3,2}v_x^e - C_{3,3}v_y^e & C_{3,3}v_y^e - C_{3,1}v_x^e & C_{3,1}v_y^e - C_{3,2}v_x^e & 0_{1 \times 14}
\end{bmatrix}
\]
where \(i, j\) in \(C_{i,j}\) denotes the index of the ECEF to the IMU body frame rotation matrix \(C\).
4.4.3 Experiment Set-Up

Description of Test Trajectory and Sensors

To extended FDE method is tested using a dataset collected in Albert Park Drive, Melbourne, Australia. The same Xsens MTi-G IMU is used in this test, along with a VSS and a Leica GS10 geodetic GNSS receiver. Dual frequency positioning solutions, obtained from the Leica receiver, is treated as reference. The trajectory of this test lasts 15 minutes. As shown in figure 4.14, the average number of satellite for this test is 8 satellites. Note that in the subsequent analysis, only GPS satellites are used to simplify the process of simulating signal shadowing and multipath effects.

![Figure 4.14: Number of satellites (GPS only)](image)

Multipath Simulation and Analysis Strategy

The effects of signal shadowing and multipath are set up to evaluate the effectiveness of the proposed technique. Here, two main scenarios are considered, both reflecting dense urban environments. In the first scenario, only three satellite signals are available, two of which are affected by multipath, while in the second scenario, all three satellite signals are affected. For each scenario, the effects are simulated four times at different parts of the trajectory, each lasting 60 seconds. The multipath simulation is then divided into three classes, each varying in ranging error magnitude: small ($\sim 10$ m), medium ($\sim 50$ m) and large ($\sim 150$ m). These magnitudes are then multiplied with randomly distributed
values, which are varied three times within each simulation to better reflect the varying conditions in urban environments. To gather more meaningful results, this process is repeated 10 times, meaning there are 40 realisations of the signal shadowing and multipath effects for each scenario. To provide a more comprehensive analysis of the effectiveness

![Figure 4.15: Satellite ranging error resulting from simulated multipath](image)

<table>
<thead>
<tr>
<th>Codename</th>
<th>Navigation</th>
</tr>
</thead>
<tbody>
<tr>
<td>s-0</td>
<td>GNSS</td>
</tr>
<tr>
<td>s-1</td>
<td>IMU/GNSS</td>
</tr>
<tr>
<td>s-2</td>
<td>IMU/GNSS with FDE</td>
</tr>
<tr>
<td>s-3</td>
<td>IMU/GNSS/VSS/Constraint with extended FDE</td>
</tr>
</tbody>
</table>

Table 4.10: Given codename for each navigation technique

of the MSS with extended FDE, it is compared to three other navigation solutions. These are GNSS only, IMU/GNSS integrated, and IMU/GNSS with conventional FDE. Each navigation technique is given a codename, as listed in table 4.10. It should be noted that s-3 is essentially the MSS with non-holonomic constraint and extended FDE. Also, note that all of these variants use EKF as their base state estimator.

4.4.4 Results

This section presents the results of the proposed FDE method. It focuses primarily on the 2D results, since the third component can be easily solved via map matching, particularly
when the 2D errors are small [94,103]. It is structured like the previous subsection: results from scenario 1 are presented first, followed by those from scenario 2. As an overview of the comparison of the results presented in the following, the RMSEs for the GNSS, IMU/GNSS, and MSS with full GNSS availability are all around 0.9 m, as shown in figure 4.16.

![Figure 4.16: 2D RMSE for full GNSS availability](image)

**Scenario 1 Results**

Table 4.11 lists the mean and maximum 2D RMSE from each navigation output with varying error magnitudes. From this table, it is clear that when only GNSS (s-0) is used, the RMSE is notably large. For example, the mean RMSE for medium magnitude error is 83.07 m, while its corresponding maximum error is 637.71 m. When the IMU/GNSS (s-1) is employed, the mean and maximum RMSEs are reduced to 39.31 m and 88.58 m, respectively. Unlike s-0 and s-1, where faulty signals are continuously used in their navigation process, the RMSE for IMU/GNSS with FDE (s-2) is significantly reduced as it is able to reject faulty signals. The MSS with extended FDE (s-3) performs the best, providing a mean RMSE of 3.05 and a maximum RMSE of 6.82 m. This is due to the inclusion of an additional sensor and the application of the non-holonomic constraint, which allows for the application of the extended FDE. When comparing mean and maximum RMSEs, s-3 provides significant improvements of 75% and 80% over s-2.

It is interesting to note that when comparing the RMSE values between small and
4.4 Improving Faulty Satellite Detection and Rejection

<table>
<thead>
<tr>
<th>Codename</th>
<th>Mean 2D RMSE</th>
<th>Maximum 2D RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>small</td>
<td>medium</td>
</tr>
<tr>
<td>s-0</td>
<td>18.79</td>
<td>83.07</td>
</tr>
<tr>
<td>s-1</td>
<td>18.08</td>
<td>39.31</td>
</tr>
<tr>
<td>s-2</td>
<td>12.71</td>
<td>16.01</td>
</tr>
<tr>
<td>s-3</td>
<td>2.61</td>
<td>3.05</td>
</tr>
</tbody>
</table>

Table 4.11: Mean and maximum 2D RMSE at varying multipath error magnitude

large multipath error magnitudes, the RMSE is smaller when large multipath errors are present compared to small multipath errors for both s-2 and s-3. Note that this only applies for s-2 and s-3, as they have FDE (s-2) and extended FDE (s-3) as part of their integration process. The increased RMSE when smaller multipath errors are present is due to the slow growth error (SGE) effect, where the introduced error slowly grows over time, making it more difficult for the navigation system to correctly identify faulty signals. This effect can be seen in figures 4.17a and 4.17b, where each figure plots the number of satellites available and resulting satellites retained after rejection for s-2 and s-3 in the 652-712 s region, when small and large multipath errors are induced. As previously explained, two out of the three satellite signals are corrupted due to multipath. From these figures, it is evident that when the induced error is large (figure 4.17b), both s-2 and s-3 manage to reject faulty satellites almost immediately, whereas when the induced error is small (figure 4.17a), only s-3 manages to correctly identify and subsequently reject faulty signals. On the other hand, s-2 takes an additional 20 seconds, coinciding with the time when the magnitude of the error changes to a larger value. Recall that the error magnitude varies three times during each simulation. Note that in figure 4.17a, both s-2 and s-3 fail to identify one of the faulty signals in the 692-712 s region due to the SGE effect.

A further investigation of the effects on the MSS with extended FDE is subsequently conducted, extending the simulation to 180 and 300 seconds with a medium class multipath error magnitude. The results are listed in table 4.12, while figure 4.18 the number of satellites rejected for both s-2 and s-3. It is observed that s-2 performs poorly when the duration is extended to 180 seconds, and even worse for 300 seconds, with a maximum RMSE of 353.36 m and 1336.37 m, respectively. On the other hand, s-3 performs much
better when its maximum RMSE is kept below 10 m. Figure 4.18 could explain the resulting RMSE: it shows that without VSS, the non-holonomic constraint, and subsequently the extended FDE, s-2 cannot correctly identify faulty signals. While it does correctly do so at the beginning of the simulation, it begins to incorrectly accept the faulty signals at 370 s. Conversely, s-3 manages to correctly reject the two faulty signals throughout the entire duration of the simulation, hence producing better RMSEs.

Table 4.12: Mean and maximum 2D RMSEs at varying time durations

<table>
<thead>
<tr>
<th>Scenario 1</th>
<th>Mean 2D RMSE</th>
<th>Maximum 2D RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Codename</td>
<td>180 s</td>
<td>300 s</td>
</tr>
<tr>
<td>s-2</td>
<td>64.33</td>
<td>353.36</td>
</tr>
<tr>
<td>s-3</td>
<td>3.25</td>
<td>4.33</td>
</tr>
</tbody>
</table>

Scenario 2 Results

In scenario 2, all three available signals are faulty. Hence, if the navigation system correctly identifies all faulty signals, then the IMU for s-2 or MSS for s-3 continues to provide positioning solutions without any aid from GNSS. Table 4.13 lists the resulting RMSEs for s-1, s-2, and s-3. As expected, s-1 performs the worst, like in scenario 1: it continuously uses faulty signals throughout the simulations. In contrast, s-2 and s-3 are able to reject
these signals, thereby performing better. s-3 provides the best RMSE, on average achieving an improvement of 70% over s-2. Furthermore, the results of this scenario are consistent with those of scenario 1: when the induced error magnitude is larger, the RMSE is generally smaller as the errors are easier to detect. Furthermore, the RMSEs in scenario 2 are generally larger than those in scenario 1. This is expected as in this scenario, the IMU in s-2 and MSS in s-3 mostly operate independently without aid from the GNSS.

<table>
<thead>
<tr>
<th>Codename</th>
<th>Mean 2D RMSE</th>
<th>Max 2D RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>small</td>
<td>medium</td>
</tr>
<tr>
<td>s-1</td>
<td>24.70</td>
<td>50.30</td>
</tr>
<tr>
<td>s-2</td>
<td>14.36</td>
<td>11.57</td>
</tr>
<tr>
<td>s-3</td>
<td>3.90</td>
<td>4.22</td>
</tr>
</tbody>
</table>

Table 4.13: Mean 2D RMSEs at varying error magnitudes

Figures 4.19a and 4.19b show how well s-2 and s-3 perform in detecting and rejecting all three faulty signals due to multipath. They reveal that like in scenario 1, both s-2 and s-3 are unable to correctly detect and reject faulty signals when the multipath error magnitude is small. When the error magnitude is larger, in contrast, they can better reject the faulty signals, hence providing better results. In figure 4.19b, it seems that the s-2 provides better detection than s-3 does, beginning at 692 s. However, it should be noted that this plot represents one of the 40 simulation instants. This shows that at times, s-3
can also incorrectly detect faulty signals but, on average, it does produce better results than s-2.

![Graph](image)

**Figure 4.19: Rejected satellites for s-2 and s-3 in scenario 2**

Similar to the analysis for scenario 1, s-2 and s-3 are also tested when the duration of signal shadowing and induced multipath error is prolonged in scenario 2. The results are listed in table 4.14, revealing that RMSEs for s-2 and s-3 both increased dramatically compared to when the simulation was set to run for 60 seconds only. This is either because they incorrectly use faulty signals, or because they navigate without the aid of GNSS. Either way, the resulting RMSE is generally larger than the ones in scenario 1, as unlike in the latter, one of the signals is still correct. Figures 4.20a and 4.20b show the resulting number of satellites used and 2D RMSEs throughout the extended duration for one of the simulation instants. In this example, both s-2 and s-3 manage to exclude all faulty signals. The resulting RMSE shown in figure 4.20b is expected: the RMSE for s-2 continues to grow unbounded over time, typical of an IMU operating without GNSS aid. s-3, on the other hand, manages to perform significantly better, keeping its maximum RMSE below 30 m. This signifies its improvement even without aid from any GNSS signals. Even when the duration is extended to 300 seconds, the maximum RMSE is better than 35 m due to the presence of VSS and the application of the non-holonomic constraint.
### 4.5 Summary

This chapter has presented the improvement of the integrated navigation system via better modelling of IMU stochastic errors and the introduction of an enhanced method to reject faulty satellite signals. Before presenting the proposed methods, the first section of this chapter discussed the conventional IMU/GNSS integrated system.

The second section then introduced the application of the DAV to a real kinematic dataset to further improve the method to extract more realistic coefficients of IMU stochastic errors. The results showed that there was some gain in the coefficient values compared to the values obtained from the AV method. This is because IMU measurement errors are influenced by vehicle motion and the environment (vibration), and particularly measured accelerations. Next, the coefficients obtained from the AV and DAV methods were used in the IMU/GNSS integration algorithm. The results showed that there were slight improvements when the coefficients from the DAV were used, particularly during com-
plete GNSS outages. Therefore, it is recommended that the IMU/GNSS integrated system take into consideration the effects of vehicle motion and the environment, which can be estimated using DAV analysis.

Finally, the last section presented an enhanced method for detecting and rejecting faulty GNSS signals due to multipath. It first discussed the conventional methods of FDE using WLS and EKF. Then, it introduced the enhanced FDE. The section also presented the measurement model for VSS and the non-holonomic constraint technique, which are critical to the extended FDE method. These methods were then tested using real datasets with realistic simulation of induced multipath errors. The results showed that the enhanced FDE could provide significantly better detection and rejection of faulty satellite signals, subsequently providing better positioning solutions. An important note on this enhanced technique is that it can only be utilised when the VSS and non-holonomic constraint are used to limit the error growth of MSS.

Having completed the individual improvements of CP algorithms in the previous chapter and MSS in this chapter, the following chapter demonstrates how MSS is integrated into the CP scheme.
Chapter 5
Multi-Sensor System and Cooperative Positioning: Analysis Using Real Datasets

5.1 Introduction

The previous two chapters have presented methods of improving positioning using network-level (CP) and local-level observations (MSS). This chapter now discusses the combination of these two separate methods, with the CP scheme also including MSS as part of the overall estimation process. Specifically, the chapter focuses on how MSS can be exploited to improve CP in terms of accuracy, continuity, and reliability, which is the main objective of this dissertation. It is divided into two sections: the first presents the topic of improving the CP technique based on non-radio ranges, while the second section covers the CP technique using radio-based ranges. The first section of this chapter was presented at the ION 2013 technical meeting.


The contents from this paper have been slightly altered to better fit the flow of this thesis. Specifically, the discussions of the EKF algorithm, IMU, and GNSS pseudorange
measurements are excluded in this chapter as they have already been discussed in the previous three chapters.

5.2 Improving Non-Radio-Based Ranging CP Using MSS

This section details a new CP approach, which proposes a tightly coupled CP with an MSS integrated system without relying on radio-based ranging. As discussed in chapter 2, non-radio ranging is less complex than radio-based ranging, as all that is required is the connectivity between the vehicles in a VANET. The study in [4] shows that promising results can be obtained when low-level GNSS data is shared among VANET participants. Moreover, the study shows that relative positioning using a code-based double differencing technique between two moving vehicles can achieve better accuracy of relative positioning than conventional DGNSS can. However, the technique presented in that study can only work when four common satellites are observed simultaneously. This makes it less suitable for use when GNSS satellite availability is limited. The present work aims to overcome this limitation by employing MSS, which provides more redundancy to the CP system. This effectively allows the number of GNSS satellites to be reduced from four to two to successfully employ this technique. In this approach, the participating vehicles exchange pseudoranges, positions, and their respective variances, and the medium by which this is done is the DSRC, as depicted in figure 5.1. As shown in later subsections, the proposed technique is tested using real datasets. The results show that the proposed method outperforms GNSS CP in terms of continuity and standalone tightly coupled MSS by up to 60% during GNSS outages, which is particularly beneficial in urban environments. This section is structured as follows. First, it presents the background regarding the measurements involved in this technique. Earlier chapters have already discussed the background of IMU and GNSS; hence, these are not elaborated further here. However, the proposed technique does introduce a new set of measurements, the double differenced GNSS pseudoranges, discussed in the next subsection. The subsequent subsection then covers the TC CP algorithm, which comprises MSS and a relative positioning measurement. Then, a brief description of the dataset used to validate the
5.2 Improving Non-Radio-Based Ranging CP Using MSS

Figure 5.1: Schematic of the proposed TC CP approach

An improved CP technique is presented. Finally, the results section discusses the improvements of the proposed method over standalone MSS.

5.2.1 Code-Based Double Differencing

As mentioned in chapter 2, spatially correlated errors can be reduced using the measurement differencing technique. This is widely used in DGNSS and real-time kinematic (RTK) positioning [4, 54]. Differencing can be done in a number of ways, known as single, double, and triple differencing. Single differencing refers to two receivers observing a common satellite; double differencing refers to two receivers observing two common satellites; and triple differencing is differencing double differences between epochs [54]. The differencing technique allows for the creation of baselines between two receivers and is often termed relative positioning. The baseline is of interest here, as it can be used in the CP technique. In this work, the double difference technique is chosen for its balance between complexity, practicality, and performance for real-time implementation. Moreover, only code-based measurements are considered here as phase measurements are more sensitive to the effects of multipath, which is common in urban environments, and they are more complex to resolve due to the presence of integer ambiguity [54].

Consider a scenario where there are two vehicles and satellites, denoted M, P, i, and j, respectively, as depicted in figure 5.2. The code-based double difference can be formu-
lated as

\[ \tilde{\rho}_{ij}^{MP} = \tilde{\rho}_i^{M} - \tilde{\rho}_j^{M} + \tilde{\rho}_i^{P} - \tilde{\rho}_j^{P} \]  

(5.1)

Consequently substituting equation 2.1 into equation 5.1 yields

\[ \tilde{\rho}_{ij}^{MP} = \rho_{ij}^{MP} + \nu_{ij}^{MP} \]  

(5.2)

Equation 5.2 shows that all of the spatially correlated and satellite clock errors have been removed. \( \tilde{\rho}_{ij}^{MP} \) is calculated based on the observed pseudoranges, while \( \nu_{ij}^{MP} \) represents the uncorrelated errors that cannot be eliminated with the differencing technique, here treated as observation noise. The double difference of the distances between vehicles and satellites is given by

\[ \tilde{\rho}_{ij}^{MP} = \left[ \vec{\mu}_i^{M} - \vec{\mu}_j^{M} \right]' \vec{r}_{MP} \]  

(5.3)

from

\[ \rho_i^{M} - \rho_i^{P} = \vec{\mu}_i^{P} \vec{r}_M \]  

(5.4)

\[ \rho_j^{M} - \rho_j^{P} = \vec{\mu}_j^{P} \vec{r}_M \]  

(5.5)

where \( \vec{\mu} \) and \( \vec{r} \) denote the unit vector and relative position vector, respectively. Note that the unit vectors to each satellite are equivalent for all vehicles in an area of tens of
kilometres due to the large distances between vehicles and satellites [4]. Substituting equation 5.3 into equation 5.2 yields

$$\tilde{\rho}_{ij}^{MP} = \left[ \bar{\mu}_i^M - \bar{\mu}_j^M \right]' \tilde{r}_{MP} + v_{ij}^{MP} \quad (5.6)$$

To solve for the ranges, the receivers need to track at least four common satellites (three baselines). Equation 5.6 can be expanded to:

$$\begin{bmatrix} \tilde{\rho}_{ij}^{MP} \\ \tilde{\rho}_{im}^{MP} \\ \tilde{\rho}_{in}^{MP} \end{bmatrix} = \begin{bmatrix} (\bar{\mu}_i^M - \bar{\mu}_j^M)_x \\
(\bar{\mu}_i^M - \bar{\mu}_j^M)_y \\
(\bar{\mu}_i^M - \bar{\mu}_j^M)_z \\
(\bar{\mu}_i^M - \bar{\mu}_m^M)_x \\
(\bar{\mu}_i^M - \bar{\mu}_m^M)_y \\
(\bar{\mu}_i^M - \bar{\mu}_m^M)_z \\
(\bar{\mu}_i^M - \bar{\mu}_n^M)_x \\
(\bar{\mu}_i^M - \bar{\mu}_n^M)_y \\
(\bar{\mu}_i^M - \bar{\mu}_n^M)_z \end{bmatrix} \begin{bmatrix} (\tilde{r}_{MP})_x \\ (\tilde{r}_{MP})_y \\ (\tilde{r}_{MP})_z \end{bmatrix} \quad (5.7)$$

where subscripts $x, y, z$ denote the unit vector axes and superscripts $m, n$ denote satellite index.

### 5.2.2 Tightly Coupled CP and Measurement Model

This subsection presents the TC CP technique. Three types of measurements are used: MSS, GNSS pseudoranges, and baselines between vehicles using the code-based double difference technique. These measurements are tightly integrated (figure 5.3) to fully utilise all available GNSS pseudoranges. As mentioned in chapter 4, the TC allows for continuous positioning even when less than four satellites are observed. Similarly, it also allows for only two satellites to be observed simultaneously to solve for the baselines between vehicles, as enough redundancy is now provided by the MSS.

The work in this section utilises the decentralised architecture where, as depicted in figure 5.1, the vehicles in a VANET solve for their own position, velocity, and attitude while receiving pseudoranges and position information from other vehicles. This approach is employed due to its practicality for use in VANET in that it does not require a fixed infrastructure to enable CP. Here the EKF is used to integrate all of the aforementioned measurements used in this system, where the state vector is as presented in equation 4.1. The system model employed is the same as the one presented in subsection 4.2.2. Subsequently, the measurement model for the pseudorange is also the same as the
one discussed in subsection 4.2.3. The new measurement model for relative positioning is presented in the following paragraphs.

The relative position between vehicles is formed using the code-based double difference technique. As before, assume there are two vehicles, denoted as $P$ and $M$, where $P$ is the vehicle of interest (figure 5.2). The two vehicles are simultaneously observing three common satellites, denoted as $i$, $j$ and $k$. For vehicle $P$ to receive aiding from vehicle $M$, equation 5.3 is altered to

\[
\rho_{ij}^{MP} + \left[ \vec{\mu}_M - \vec{\mu}_i \right]'r_M = \left[ \vec{\mu}_M - \vec{\mu}_j \right]'r_P \tag{5.8}
\]

\[
\rho_{ik}^{MP} + \left[ \vec{\mu}_M - \vec{\mu}_i \right]'r_M = \left[ \vec{\mu}_M - \vec{\mu}_k \right]'r_P \tag{5.9}
\]

where $A_{ij}^M$ and $A_{ik}^M$ denote

\[
A_{ij}^M = \left[ \vec{\mu}_M - \vec{\mu}_i \right]' \quad A_{ik}^M = \left[ \vec{\mu}_M - \vec{\mu}_k \right]' \tag{5.10}
\]

then

\[
\tilde{z} = \left[ \rho_{ij}^{MP} + A_{ij}^M r_M \quad \rho_{ik}^{MP} + A_{ik}^M r_M \right]' \tag{5.11}
\]

\[
\tilde{z} = \left[ A_{ij}^M r_P \quad A_{ik}^M r_P \right]' \tag{5.12}
\]
The design matrix for the measurements is defined as

\[
H = \begin{bmatrix}
A_{ij}^{M} & 0_{1\times19} \\
A_{ik}^{M} & 0_{1\times19}
\end{bmatrix}
\] (5.13)

The measurement noise consists of noise from the double difference ($\rho_{ij}^{DD}$ and $\rho_{ik}^{DD}$) covariance and the position ($Ar_{M}$) covariance. Note that the superscript of $\rho_{ij}^{DD}$ will be replaced by $DD$ denoting double difference in the following equations. Using the propagation of variance rule, the variance of the aiding measurement is constructed as

\[
R = 2 \left( R(\rho_{ij}^{DD}) + R(Ar_{M}) \right) \] (5.14)

The double difference covariance can be calculated from the single difference (SD) covariance, as shown in the following equations [54].

\[
R(\rho_{ij}^{DD}) = BR(\rho_{ij}^{SD})B'
\] (5.15)

\[
B = \begin{bmatrix}
-1 & 1 & 0 \\
-1 & 0 & 1
\end{bmatrix}
\] (5.16)

\[
R(\rho_{ij}^{SD}) = CR(\rho)C'
\] (5.17)

\[
C = \begin{bmatrix}
-1 & 1 & 0 & 0 \\
0 & 0 & -1 & 1
\end{bmatrix}
\] (5.18)

\[
R(\rho_{M}) = UERE^{2} *
\begin{bmatrix}
\frac{1}{\sin \alpha_{Si}} & 0 & 0 & 0 \\
0 & \frac{1}{\sin \alpha_{Sj}} & 0 & 0 \\
0 & 0 & \frac{1}{\sin \alpha_{Sj}} & 0 \\
0 & 0 & 0 & \frac{1}{\sin \alpha_{Sl}}
\end{bmatrix}
\] (5.19)

where the $UERE$ denotes the user equivalent ranging error, discussed in chapter 2. Fi-
nally,
\[ R(Ar_M) = AR(r_M)A' \] (5.20)

where, \( R(r_M) \) is the variance of position \( M \).

### 5.2.3 Experiment Dataset

The dataset to test the proposed CP technique is the same as the one used in section 4.3. It was collected on a major road near the University of Nottingham campus using two vehicles. This study uses 16 minutes of data, covering over 12 kilometres. The trajectory had both low and medium dynamics in terms of velocity and turning profiles, depicting typical land-based vehicle dynamics. For example, velocity along the trajectory varied, reaching up to 90 km/h, and had several slow/slight and relatively fast turns. The satellite availability for this particular test was about 95%, and several partial GNSS outages were experienced while driving under bridges and passing low-rise buildings.

![Trajectory](image)

Figure 5.4: Trajectory of test site

Figure 5.5 shows the ranges between the two vehicles, calculated from the reference
5.2 Improving Non-Radio-Based Ranging CP Using MSS

solutions and double difference pseudoranges with full satellite availability. As depicted in the figure, the separation between the vehicles varied from 6 m to about 900 m. This variation served to observe if distance had any effect on the proposed CP performance.

![Figure 5.5: Pseudorange double differencing range](image)

5.2.4 Hardware Configuration

Four main sensors were fitted in each vehicle: a navigational-grade IMU (Honeywell CIMU), a geodetic-grade GNSS receiver (Leica GS10), an MEMS-based IMU (Xsens MTi-G), and a DSRC transceiver to enable communication between vehicles. The navigational-grade IMU, integrated with a dual-frequency GNSS solution, was treated as the reference data as it provides high-accuracy navigation solutions. The GNSS base station was located on the rooftop of the Nottingham Geospatial Institute building, about 1-5 km away from the test track, and it had a good open sky area to maximise satellite visibility. The IMU used to validate this research was the Xsens MTi-G. It is an MEMS-based IMU/GNSS navigational device consisting of tri-axial accelerometers, gyroscopes, and magnetometers; a GNSS receiver; and an on-board processing unit. Its GNSS receiver can make use of up to 50 channels of the GNSS C/A code on the L1 frequency and track up to 4 Hz, while its on-board processing unit is able to fuse the IMU and GNSS data in real time. Measurements from the Xsens MTi-G were automatically time synchronised using its internal GNSS receiver. Both the CIMU/GNSS and Xsens MTi-G were aligned carefully to enable direct comparison of their navigation solutions.

The CIMU/GNSS solution was processed using a well-known IMU/GNSS integra-
tion software, the Inertial Explorer by Novatel. On the other hand, the XsensMTi-G/GNSS solution was processed using Matlab and the code that was developed as part of this dissertation. The DSRC was used to transfer the shared information between vehicles, which included their positions, GNSS pseudoranges, and covariances, as depicted in figure 5.3. As discussed in [4], the bandwidth is more than enough to accommodate this data, even if the rate is more than 1 Hz. This makes the proposed CP approach possible to use in many critical positioning applications.

5.2.5 Results

This subsection presents the result of the tightly integrated CP combined with MSS, herein simply called CP, by comparing its performance to the standalone tightly integrated MSS using the collected dataset. In the analysis, one vehicle acts as the aiding source while the other receives information through its DSRC. The following subsections present two different scenarios in which CP is assessed. First, its performance with full GNSS availability is presented. In the second scenario, its performance during partial GNSS outages is validated; these outages are simulated to reflect a typical urban environment. As elaborated in the next subsection, the integrated system is aided by three and two GNSS satellites during simulated partial GNSS outages. To observe its performance further, the duration of the simulated GNSS outages is varied from 60 to 180 to 300 seconds. The results are presented as mean RMSE, maximum error, and percentage of improvement.

Full GNSS Availability

Figure 5.6 shows the 2D positioning error of the integrated systems over time, while table 5.1 lists their mean RMSEs and maximum errors with full GNSS availability. It is clear from the figure and table that the performance of MSS CP is comparable to that of the standalone MSS. It can be observed that the 3D error is higher than the 2D error due to the satellite geometry, affecting the height solutions. The CP technique has little influence in correcting the vertical error. The limited increase of performance, as quantified by ob-
serving the % of improvement when there is full GNSS availability, is expected. This is because even though the ranging derived from the double difference is relatively accurate, the CP performance is dependent on the aiding vehicle, which itself is influenced by positioning errors even when it has been accounted for in the CP algorithm.

![Figure 5.6: Position error comparison with full GNSS availability](image)

<table>
<thead>
<tr>
<th>Full GNSS</th>
<th>Mean RMSE (m)</th>
<th>Max Error (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2D</td>
<td>3D</td>
</tr>
<tr>
<td>MSS</td>
<td>1.55</td>
<td>10.44</td>
</tr>
<tr>
<td>CP</td>
<td>1.54</td>
<td>10.35</td>
</tr>
<tr>
<td>Improvement %</td>
<td>0.49</td>
<td>0.80</td>
</tr>
</tbody>
</table>

Table 5.1: Full GNSS coverage mean RMSE and maximum error

Partial GNSS Availability

This subsection details the performance of the proposed CP during partial GNSS outages. It first presents the results of the scenario when only three satellites were available and the duration of partial outages varied from 60 to 180 to 300 s.

As seen in figure 5.7 and table 5.2, the performance of CP improves significantly. The mean 2D RMSE improves by 60% and 50% when experiencing 60 and 300 s of partial outages, respectively. On the other hand, like with full GNSS availability, CP with three
satellites mean 3D RMSE only improves slightly. For example, the CP mean RMSE improves by 6% and 2% compared to MSS during the 60 and 300 s outages, respectively. The CP maximum error shows a similar trend as the mean RMSE, with its 2D maximum error improving more than its 3D maximum error. However, not all of the CP results are improvements compared to MSS. For example, the CP maximum error when experiencing 300 seconds of partial outage is higher than its MSS counterpart. The difference is observed at the later part of the simulated outage and is due to the effect of the aiding vehicles solution not being accurate.

The following presents the CP results for the scenario when only two satellites were available and simultaneously observed by both vehicles. As before, the duration of the outages varied from 60 to 180 to 300 seconds. Similar to when three satellites were available, the 2D results in this scenario show that an improved performance of CP compared to MSS. The percentages of improvements are 40%, 43%, and 24% for 60, 180, and 300 s of outages, respectively. On the other hand, CPs 3D performance decreases by 25% or 2m worse than its counterpart when experiencing 300 s of outage. The results for CP maximum errors are also similar to the mean RMSE: for the most part, they are significantly better than for MSS. For example, during 60 s of partial outage, the maximum error observed is reduced by 39% compared to MSS.

When observing the results listed in tables 5.2 and 5.3, it can be seen that performance degrades over time. This indicates that although ranging from neighbouring vehicles is
5.2 Improving Non-Radio-Based Ranging CP Using MSS

<table>
<thead>
<tr>
<th>3 Satellites</th>
<th>Mean RMSE (m)</th>
<th>Max Error (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2D</td>
<td>3D</td>
</tr>
<tr>
<td>60 seconds</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSS</td>
<td>5.66</td>
<td>9.98</td>
</tr>
<tr>
<td>CP</td>
<td>2.23</td>
<td>9.38</td>
</tr>
<tr>
<td>Improvement %</td>
<td>60.54</td>
<td>6.04</td>
</tr>
<tr>
<td>180 seconds</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSS</td>
<td>6.12</td>
<td>8.99</td>
</tr>
<tr>
<td>CP</td>
<td>2.73</td>
<td>8.82</td>
</tr>
<tr>
<td>Improvement %</td>
<td>55.43</td>
<td>1.89</td>
</tr>
<tr>
<td>300 seconds</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSS</td>
<td>6.74</td>
<td>9.07</td>
</tr>
<tr>
<td>CP</td>
<td>3.36</td>
<td>8.88</td>
</tr>
<tr>
<td>Improvement %</td>
<td>50.17</td>
<td>2.07</td>
</tr>
</tbody>
</table>

Table 5.2: Partial GNSS 3 satellites 2 RMSE and maximum error

utilised, CPs performance is heavily reliant on the performance of MSS. As numerous studies have shown [43, 47], its positioning accuracy declines over time in the absence of full GNSS availability. Furthermore, the performance of CP is dependent on the quality of the aiding vehicles positioning information. Hence, if the quality of the neighbouring vehicles positions is poor, this will affect the quality of the target vehicle. Nonetheless,
<table>
<thead>
<tr>
<th>2 Satellites</th>
<th>Mean RMSE (m)</th>
<th>Max Error (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2D</td>
<td>3D</td>
</tr>
<tr>
<td>60 seconds</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSS</td>
<td>3.14</td>
<td>9.36</td>
</tr>
<tr>
<td>CP</td>
<td>1.87</td>
<td>9.49</td>
</tr>
<tr>
<td>Improv. %</td>
<td>40.41</td>
<td>-1.39</td>
</tr>
<tr>
<td>180 seconds</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSS</td>
<td>5.12</td>
<td>8.79</td>
</tr>
<tr>
<td>CP</td>
<td>2.88</td>
<td>9.78</td>
</tr>
<tr>
<td>Improv. %</td>
<td>43.6</td>
<td>-11.24</td>
</tr>
<tr>
<td>300 seconds</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSS</td>
<td>7.15</td>
<td>9.48</td>
</tr>
<tr>
<td>CP</td>
<td>5.4</td>
<td>11.91</td>
</tr>
<tr>
<td>Improv. %</td>
<td>24.52</td>
<td>-25.59</td>
</tr>
</tbody>
</table>

Table 5.3: Partial GNSS 3 satellites mean RMSE and maximum error

the proposed CP approach shows significant improvements, particularly in its 2D solutions when GNSS is only partially available. This would be beneficial for applications requiring critical positioning in GNSS-hostile environments.

Overall, the proposed approach shows marked improvements over standalone MSS, particularly during partial GNSS outages. Unlike previous studies, such as [4], the TC approach also allows the CP approach to function even when less than four common satellites are available. Using the experiment data collected in Nottingham, the proposed CP approach was tested and results showed that it was able to improve RMSE by up to 60% during partial GNSS outages. Another highlight of the proposed system over other conventional techniques of CP is the independence of inter-vehicle radio ranging, such as RSS, TOA, and TDOA.

5.3 Improving Radio-Ranging-Based CP Using MSS

The CP approach presented in the previous section requires only the exchange of GNSS pseudoranges to enable cooperation between vehicles. These pseudoranges are then used
to form the double difference ranging measurements between vehicles. While this technique has shown promising results when vehicles experience low satellite visibility, its application is limited in dense urban environments, where the pseudoranges are adversely affected by multipath. Correct formation of the double difference measurements in this case is not possible. The only way to overcome this problem is to employ a radio ranging measurement that does not suffer from the effects of multipath. In this dissertation, the UWB is chosen to provide radio-based ranging between vehicles and vehicles, and vehicles and infrastructure, collectively termed V2X. As discussed in chapter 2, the allocated frequency band of UWB does not interfere with DSRC and GNSS, and its wide band transmission makes it less susceptible to multipath effects [6]. These two facts make the UWB a viable technology to provide for the V2X ranges in VANET.

This section discusses how to improve radio-ranging-based CP using MSS. It first presents a series of experiments conducted to test the viability of the proposed CP method, where UWB transceivers were employed to provide V2X ranges between two moving vehicles and two RSUs. Next it discusses the employed UWBs characteristics, such as its accuracy and error distribution, lever arm, and UWB-GNSS time synchronisation effects. This is followed by a discussion of the proposed CP systems architecture and, subsequently, the analysis methodology to prove the effectiveness of the proposed system. Finally, the results of the analyses are presented at the end of this section.

5.3.1 Dataset Description

Figure 5.9: Experiment platforms consisting of two moving vehicles and two RSUs

A series of experiments were conducted in the vicinity of Melbourne, Australia with
the aim of collecting datasets to test the validity of the CP concept. A network of platforms was employed, with two vehicles acting as moving rovers and two static platforms acting as RSUs, shown in figure 5.9. The tests were conducted in multiple environment scenarios, such as open sky, residential, and dense urban environments. However, only the test performed in the open sky area is used in this study due to the lack of equipment needed to obtain ground truth in harsh GNSS environments. In total, five main sensors were deployed and attached to these platforms, including GNSS, IMU, VSS, UWB, and DSRC (depicted in figure 5.10). The deployment of these sensors on each platform is described in table 5.4, and the general specifications of each sensor are listed in table 5.5.

![Image](image.png)

(a) Sensor platform on the RSU  
(b) Sensor platform on the vehicle

Figure 5.10: Sensor platform

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Vehicle 1</th>
<th>Vehicle 2</th>
<th>RSU 1</th>
<th>RSU 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>GNSS</td>
<td>Leica GS10</td>
<td>Leica GS10</td>
<td>Leica GS15</td>
<td>Leica GS15</td>
</tr>
<tr>
<td>IMU</td>
<td>GX3-45</td>
<td>MMQG</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>VSS</td>
<td>Vehicle OBDII</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DSRC</td>
<td>MK2</td>
<td>MK2</td>
<td>MK2</td>
<td>MK2</td>
</tr>
<tr>
<td>UWB</td>
<td>1 × P410</td>
<td>2 × P410</td>
<td>2 × P410</td>
<td>2 × P410</td>
</tr>
</tbody>
</table>

Table 5.4: Sensors deployed on each platform

In this experiment, the Leica GS 10 and 15 were used to internally record raw GNSS observations. These measurements were then used to obtain the dual-frequency GNSS positioning output, which was treated as the ground truth. The base station used to calculate the dual-frequency solutions was located around 1.6 km away from the test
5.3 Improving Radio-Ranging-Based CP Using MSS

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Interface</th>
<th>Data rate</th>
<th>Time sync</th>
<th>Recording</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leica GS</td>
<td>-</td>
<td>5 Hz</td>
<td>-</td>
<td>Internal</td>
</tr>
<tr>
<td>MMQG</td>
<td>USB</td>
<td>100 Hz</td>
<td>GPS</td>
<td>Laptop</td>
</tr>
<tr>
<td>3DM GX3-45</td>
<td>USB</td>
<td>100 Hz</td>
<td>GPS</td>
<td>Laptop</td>
</tr>
<tr>
<td>VSS</td>
<td>Bluetooth</td>
<td>1 Hz</td>
<td>n/a</td>
<td>Mobile phone</td>
</tr>
<tr>
<td>MK2</td>
<td>USB</td>
<td>10 Hz</td>
<td>GPS</td>
<td>Laptop</td>
</tr>
<tr>
<td>P410</td>
<td>USB</td>
<td>~1 Hz</td>
<td>n/a</td>
<td>Laptop</td>
</tr>
</tbody>
</table>

Table 5.5: Sensor specifications

The Systron Dronner MMQG IMU was attached to vehicle 1, and the InertiaLink 3DM GX-45 was attached to vehicle 2 to provide inertial measurements at 100 Hz. Both sensors are low-cost IMUs and are automatically GPS time synchronised via their internal GPS receivers. During the experiment, only one VSS was used, attached to vehicle 1 via its OBDII interface. It was connected to a mobile phone via bluetooth, which stored the speed measurements at 1 Hz. The Time Domain PulsOn UWB transceivers were deployed to obtain V2X ranges. On vehicle 1, only one UWB transceiver was attached, while two were attached to each of the other three platforms, as depicted in figure 5.11. The reason for this configuration is that the default transceivers firmware only allowed for the system to operate in TW-TOF direct connection mode, as opposed to network mode. In this mode, the receiving transceiver can only process one signal at any given time. Hence, one transceiver may transmit multiple signals, but the receiving transceiver can only receive, process, and return the signal from one other transceiver. They were set to operate at 1 Hz to minimise data packet collisions. Finally, the Cohda Wireless MK2 DSRC transceivers were deployed to transmit data between the platforms at 10 Hz. During the test, dummy data with a fixed packet length of 200 bytes were transmitted, while RSS and CFO information were recorded.

5.3.2 UWB Range Characterisation

This subsection presents the measurement characteristics of the UWB deployed in the experiment, tested in both static and dynamic environments. The UWB in the static test
is presented first, and its mean error and distribution are investigated. Then, the sec-
tion discusses the time synchronisation and lever arm compensation procedures to en-
able comparison with distances calculated from dual-frequency GNSS solutions in the
dynamic test.

Measurement Characterisation in a Static Environment

Prior to the experiment mentioned in the previous subsection, a separate test was con-
ducted to obtain the characteristics, in terms of error mean and distribution and percent-
age of data loss, of the UWB measurements in a static outdoor environment. The UWB
range measurements were compared to the ranges obtained from a Leica set5x total sta-
tion, which has a range accuracy of \((2 + 2 \text{ ppm} \times x)\) mm. The total station emits infrared
carrier signals to a prism reflector, which then reflects the signal back to the total station to
obtain range measurements. As seen in figure 5.12, six points were set up as targets. The
first point (P1) was placed where there was a clear line of sight between the transceivers.
The next two points (P2 and P3) were located next to a brick wall to observe any effects of
multipath on UWB ranging. Then, P4 and P5 were located in an environment with light
foliage obstructions. The final point was located some 109 m away from the transmitter,
with very light foliage obstruction between the transceivers. Note that the locations of
the points depicted in figure 5.12 are only approximated, and not their true positions.
During the test, 20 measurements were recorded and repeated three times for each point,
and the UWB transceivers were placed directly side by side with the total station and reflector to minimise the effect of lever arm offsets.

![Figure 5.12: Static UWB test](image)

<table>
<thead>
<tr>
<th>Point</th>
<th>Total station Range (m)</th>
<th>Total station Range (m)</th>
<th>% loss</th>
<th>Std (m)</th>
<th>Error (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>46.55</td>
<td>46.72</td>
<td>7</td>
<td>0.05</td>
<td>0.17</td>
</tr>
<tr>
<td>P2</td>
<td>22.30</td>
<td>22.39</td>
<td>5</td>
<td>0.01</td>
<td>0.09</td>
</tr>
<tr>
<td>P3</td>
<td>30.70</td>
<td>30.60</td>
<td>5</td>
<td>0.03</td>
<td>0.10</td>
</tr>
<tr>
<td>P4</td>
<td>29.60</td>
<td>29.39</td>
<td>2</td>
<td>0.02</td>
<td>0.21</td>
</tr>
<tr>
<td>P5</td>
<td>48.21</td>
<td>48.04</td>
<td>6</td>
<td>0.08</td>
<td>0.17</td>
</tr>
<tr>
<td>P6</td>
<td>109.20</td>
<td>109.03</td>
<td>6</td>
<td>0.13</td>
<td>0.17</td>
</tr>
</tbody>
</table>

Table 5.6: Total station vs. UWB distance measurements

Table 5.6 shows the results obtained from the static test. An average error of 0.15 m for the UWB ranging is observed, which is higher than the advertised accuracy of 0.02 m for LOS and 0.10 m for obstructed LOS conditions [117]. The error value for P1 is treated as the baseline, as the point was located in an ideal condition with no structure
nearby and a clear LOS between the UWB transmitter and receiver. As seen in the error values of P2 and P3, the UWB range was not affected by multipath. This is expected, as the wide band signals are not susceptible to multipath effects. However, the errors for P4 and P5 increased slightly to 0.21 m and 0.17 m, respectively. It was first thought that this was due to the presence of light foliage, but the error for P1 suggests that this might not be true, as the point had a clear LOS with a similar distance to P4. When the distance was increased to 109.2 for P6, the error valued stayed at 0.17, similar to P5. Thus, the increase in distance is assumed to have no effect on the accuracy of the UWB ranges. Table 5.6 also lists the standard deviation of the UWB ranges for each point. On average, the calculated standard deviation is 0.05 m. The standard deviation values for all points are observed to be better than 0.10 m except for P6, which suggests that distance did affect the precision of the UWB measurements. It is important to note that while the standard deviation is relatively small, the errors are not normally distributed along the true mean, i.e. systematic error or bias is present. For example, the standard deviation for the range to P1 is 0.05 m, but the actual error is 0.17 m, which is not within the standard deviation value. On average, the UWB transceivers suffered a 5% loss of data during the static test. This might be due to signal collisions or error in processing signals in the UWB transceivers.

**Time Synchronisation and Lever Arm Compensation**

In the dynamic test, the UWB measurements were compared to ranges derived from dual-frequency GNSS solutions. Two issues arose when employing this method: first, the measurements from UWB were not time synchronised to GPS time, as the time was provided by the laptops; and second, the lever arm offsets between UWB transceivers and GNSS needed to be compensated for to enable direct comparison between ranges from respective sensors. For the time synchronisation of UWB to GNSS time, the laptops needed to provide accurate and consistent timing better than the micro-second level. The laptops used to record the UWB data were all capable of achieving such accuracy and consistency, as most modern computers are. Their clocks were aligned to the Coordinated Universal Time (UTC), which had a 16 s offset when compared to GPS time as of
December 2014. This is known as leap seconds. To successfully align the time between UWB and GNSS, the ranges from UWB and GNSS were cross-correlated until the differences were minimised. Figure 5.13 shows the asynchronous and synchronised ranges before and after adding the time offset. The green lines represent ranges from UWB, while the blue lines represent GNSS-derived ranges of vehicle 1 to RSU 1. For vehicle 1, the time offset of the laptop was 15.5 s, while for vehicle 2, the laptop recording UWB data had an offset of 15.1 s, both later than GPS time. Evidently, the values were not exact to 16 s, which was due to the laptop clock not being perfectly aligned to the UTC.

![Figure 5.13: Time synchronisation compensation for vehicle 1 to RSU 1](image)

The second step to enable direct comparison of the UWB ranges to the GNSS-derived ranges was to compensate the lever arm offsets between the respective sensors. These
offsets were measured prior to the experiment, as depicted in figure 5.14. The sub-figure on the left shows the schematic of the RSU platforms, and the sub-figure on the right shows the schematic of the moving vehicle platforms. Note that the drawings are not to scale and are intended for illustration purposes only. Two types of information were needed to correctly compensate the effect of the lever arm offsets: the actual lever arm offsets, and the dynamically changing orientation of the board during kinematic testing, which was derived from the IMU/dual-frequency GNSS solutions. It was only by using the lever arm offsets and orientation information that the GNSS-derived ranges could be directly compared to the UWB measurements. It is important to note that the error of the orientation from the integrated system had a direct impact on the lever arm offset compensated ranges. Hence, to ensure that the orientation error was minimised, the MSS was integrated with dual-frequency GNSS solutions.

Figure 5.15 shows the UWB and GNSS-derived ranges before and after lever arm offset adjustment for vehicle 1 to RSU 1. In this figure, the green line represents the UWB range. The GNSS-derived range before lever arm offset compensation is represented by the red line, while the range after compensation is represented by the blue line. The left sub-figure shows the region of 138-153 s, which represents the static and beginning of the kinematic phases of the test. It can be seen that after the lever arm offsets were compensated, the ranges were shifted closer to the UWB range. The UWB measurements are consistent with those in the static test: they are precise but not accurate, i.e. the errors are not within the standard deviation. When the vehicle was moving, depicted in the right sub-figure, the differences between the UWB and derived GNSS ranges did not stay constant, instead varying across time. This had two causes: first, the UWB measurement bias itself, and second, the orientation error derived from MSS solutions.

Figure 5.16 shows the matched UWB- and GNSS-derived ranges for vehicle 1 and vehicle 2, and RSU 1 and RSU 2. It indicates that the ranges were generally well matched and the maximum error observed was 0.5 m. Moreover, the results show that UWB measurements were precise but were also affected by bias, which varied when the vehicle was in motion. The findings regarding the UWB measurement characteristics discussed here will be used in the CP algorithm.
5.3 Improving Radio-Ranging-Based CP Using MSS

![Figure 5.15: Corrected range after lever arm compensation](image)

![Figure 5.16: Matched ranges from UWB and GNSS](image)

### 5.3.3 UWB Measurement Model

In this study, the UWB measurements are modelled using the simple assumption that they are normally distributed centred along the mean. To compensate for the bias that varied across the kinematic test, the standard deviation is increased to 0.25 m instead of its true standard deviations of 0.05-0.13 m, as shown in the static test results. The UWB measurement model is formulated as

\[
\tilde{z} = \rho (x, y) + \nu \tag{5.21}
\]

where \( \rho (x, y) \) is the true range between vehicles and vehicles or vehicles and RSUs, while \( \nu \) is the measurement noise. The Jacobian matrix \( H \) is then formed as

\[
H = \begin{bmatrix}
    \frac{r_{x_y} - r_{y_y}}{\tilde{q}} & \frac{r_{x_y} - r_{y_y}}{\tilde{q}} & \frac{r_{x_z} - r_{y_z}}{\tilde{q}} & 0_{1,20}
\end{bmatrix} \tag{5.22}
\]
5.3.4 System Architecture

Considering the practical limitations of data dissemination using existing technologies such as DSRC and UWB, the proposed system is designed to minimise the amount of information being shared in a VANET. Two architectures are presented here: the centralised and decentralised systems.

Figure 5.17 depicts the proposed centralised CP system. In this figure, the orange lines represent raw measurements and information that will be used by the estimator, the blue lines represent correction feedback, and the green lines represent the solutions from the estimators. This figure shows that the proposed architecture does not share all available information centrally. Instead, all of the local-level measurements, including GNSS pseudoranges, IMU inertial, and VSS speed measurements, are internally solved for by each vehicle using EKF. The resulting navigation solutions, their variances, and the UWB ranges are then transferred to the DSRC, which in turn broadcasts this information to a central processor. Subsequently, the central processor uses the information to solve for the navigation solution and provide correction feedback to all participating vehicles in the VANET. It is clear that if the raw measurements are centrally shared, a better performance could possibly be achieved. For example, a better model of the GNSS pseudoranges correlation between vehicles could be formed if they used the same satellites, which would generally yield better estimation solutions. However, this is not practically achievable in large networks due to the physical limitations of V2X communications, as addressed in [23, 65]. Hence, the proposed architecture only considers the integrated solution from each vehicle and V2X ranges to be used in the CP algorithm to minimise the DSRC bandwidth usage and update rate.

A more suitable architecture for CP in a VANET is one that is decentralised in nature. The proposed architecture in this dissertation is depicted in figure 5.18, where the different colour lines represent different input/output similar to the previous architecture. It can be seen that in this system, each vehicle only uses connections with immediate neighbours, i.e., data does not need to be hoped to a central processor in the event that
direct connections between the central processor and vehicles cannot be established. As discussed in chapter 2, this makes a VANET with a decentralised architecture more efficient in handling scalability compared to a centralised system. However, as discussed in chapter 3, the nature of a decentralised system does not allow the computation of the network joint posterior, so that only marginal posteriors can be calculated. Thus, in terms of the parameter estimation process, it is expected that the decentralised architecture and algorithm will produce less accurate navigation output compared to the centralised architecture. Figure 5.18 shows that apart from the absence of a central processor, the decentralised architecture is identical to the centralised one. As before, the local-level observations are solved for first, before the resulting information is passed on to the CP processor, utilising SPAWN or MDSPAWN as the CP state estimator.

Although a centralised system might not be feasible for use in a large-scale VANET, it is still considered in this dissertation to provide a better understanding of what performance is achievable in the event that better technologies are available in the future which allow for better data sharing capabilities and message routing strategies. The next subsection presents the analysis methodology to test the viability of the proposed CP technique with MSS integrated in its system.
5.3.5 Analysis Methodology

Using the datasets elaborated earlier in this section, the next subsection presents the results of the proposed radio-ranging-based CP system. Parallel to the main objective of this dissertation, the section emphasises the enhanced performance of CP when MSS is integrated into its system.

The structure of the analysis and following results is partitioned into two scenarios. The first scenario tests the GNSS standalone with radio-ranging-based CP, called GNSS CP. This is tested using the algorithms discussed in chapter 3 namely, centralised EKF and MDPC, and decentralised SPAWN and MDSPAWN. This scenario is further divided into three sub-scenarios. In the first sub-scenario (1.1), full GNSS availability is assumed; in the second sub-scenario (1.2), a 60 second partial GNSS signal outage is simulated; and in the final sub-scenario (1.3), both partial GNSS signal outage and multipath effects are simulated in the same region as the simulation in scenario 1.2. In scenarios 1.2 and 1.3, only two GNSS signals are available and specifically in scenario 1.3, one of the two signals suffers from multipath effects. Here, the simulation of multipath effects is configured to use the same methods discussed in section 4.4.3, such that the magnitude of error changes three times throughout the multipath simulation to better reflect realistic urban environments. The overarching objective of the first scenario is to quantify the achiev-
able performance of the navigation systems when utilising CP over the GNSS standalone system. It also aims to quantify the performance of novel (MDPC and MDSPAWN) over conventional (EKF and SPAWN) estimation techniques using real world datasets.

In the second scenario, the MSS is included as part of the CP system, forming MSS CP. It is divided into two sub-scenarios, the first of which has the same set-up as scenario 1.3. In this scenario (2.1), the MSS CP system is tested when navigating in GNSS-difficult environments while receiving ranging aid from both RSUs (V2I) and between vehicles (V2V). The analysis exemplifies how MSS is used to overcome the shortcomings of GNSS CP, as analysed in scenario 1. To further showcase the benefits of MSS CP, the connection of V2I is reduced from two to one for each vehicle. This set-up is applied to the last scenario (2.2). Except for the reduced V2I connections, this scenario still has the same set-up as sub-scenarios 2.1 and 1.3, where part of the trajectory is affected by partial GNSS signal outage and multipath. The objective of this scenario is to exemplify how MSS can provide substantially better navigation performance compared to GNSS CP. By providing measurement redundancy and positioning continuity, the MSS CP can still provide reliable positioning solutions even when the CP connections are limited. It is worth noting that when possible, i.e. when there are enough measurement redundancies, the navigation systems of both GNSS-only and MSS-based CP are applied with FDE and enhanced FDE, respectively, to reject faulty satellite signals, as discussed in section 4.4. Table 5.7 summarises the aforementioned scenarios, while the trajectory of the dataset collection is depicted in figure 5.19. In this figure, the green dots represent the reference trajectory for vehicle 1, the blue triangles represent the RSUs, and the red area marks the parts of the trajectory that are affected by simulated GNSS signal outages and multipath.

Like in the analysis in the previous section, the performance of the navigation output from the vehicles is presented as mean RMSE and maximum error. To maintain brevity, the RMSEs from both vehicles are averaged in the following subsection. For the sample-based algorithms, i.e. MDPC, SPAWN, and MDSPAWN, the RMSE was averaged using 30 Monte Carlo realisations.
<table>
<thead>
<tr>
<th>Scenario</th>
<th>Navigation mode</th>
<th>CP ranging</th>
<th>GNSS outage and multipath</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>GNSS only</td>
<td>full V2I, full V2V</td>
<td>Full availability</td>
</tr>
<tr>
<td>1.2</td>
<td>GNSS only</td>
<td>full V2I, full V2V</td>
<td>2 satellites</td>
</tr>
<tr>
<td>1.3</td>
<td>GNSS only</td>
<td>full V2I, full V2V</td>
<td>2 satellites, 1 faulty</td>
</tr>
<tr>
<td>2.1</td>
<td>MSS</td>
<td>full V2I, full V2V</td>
<td>2 satellites, 1 faulty</td>
</tr>
<tr>
<td>2.2</td>
<td>MSS</td>
<td>lim. V2I, full V2V</td>
<td>2 satellites, 1 faulty</td>
</tr>
</tbody>
</table>

Table 5.7: Summary of the different types of scenarios

Figure 5.19: Reference trajectory and area with simulated outage and multipath

5.3.6 Results

Using the methodology discussed in the previous subsection, this subsection presents the results of the standalone systems and proposed CP integrated systems.
Table 5.8 lists the results of scenario 1.1, where all available GNSS signals and CP ranging measurements were utilised throughout the test trajectory. The results are as expected, with the navigation output generally improving slightly when CP was employed compared to the GNSS standalone system. When comparing the different algorithms used for CP, there are slight improvements for novel (MDPC and MDSPAWN) over conventional (EKF) algorithm. For example, the improvement of MDPC over EKF 2D RMSE reached 32% or 0.19 m in terms of its absolute value. There is virtually no difference between the results obtained from SPAWN and MDSPAWN, but both did show an improvement over EKF at around 45%. The difference between MDSPAWN and MDPC is minimal, averaging at 0.08 m for the mean RMSE. This is expected: as shown in chapter 3, their performances were similar in well-defined conditions, such as the full GNSS availability in this scenario. Furthermore, the maximum error also improved, but only at a marginal scale when compared to mean RMSE. On average, the 2D maximum error improved by 10% when CP was employed compared to the GNSS standalone system. Similar to the
2D mean RMSE, the MDSPAWN provided the best maximum error with 2.19 m.

<table>
<thead>
<tr>
<th>Scenario 1.1</th>
<th>Mean RMSE (m)</th>
<th>Max Error (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2D</td>
<td>3D</td>
</tr>
<tr>
<td>GNSS</td>
<td>0.86</td>
<td>1.94</td>
</tr>
<tr>
<td>GNSS CP EKF</td>
<td>0.60</td>
<td>1.35</td>
</tr>
<tr>
<td>GNSS CP MDPC</td>
<td>0.41</td>
<td>1.04</td>
</tr>
<tr>
<td>GNSS CP SPAWN</td>
<td>0.34</td>
<td>0.95</td>
</tr>
<tr>
<td>GNSS CP MDSPAWN</td>
<td>0.33</td>
<td>0.95</td>
</tr>
</tbody>
</table>

Table 5.8: RMSE derived from scenario 1.1

Table 5.9 lists the results of the navigation systems in a scenario where each vehicle experienced partial GNSS signal outages for 60 s, allowing for only two GNSS signals to be available. It is clear that with limited GNSS signals, the GNSS standalone solutions degraded over time, recording a mean 2D RMSE of 42.70 m, and a maximum error of more than 400 m. With the aid of three CP ranging, two from the RSUs and one from the neighbouring vehicle, the navigation system was able to maintain a relatively acceptable positioning solution as there were now enough measurements to solve for the unknown parameters. Similar to the results for scenario 1, the novel algorithms for CP performed better than EKF. Using MDPC, the RMSE improved by 45%, and using MDSPAWN, it improved by 67% compared to EKF. Unexpectedly, both SPAWN and MDSPAWN performed slightly better than MDPC.

<table>
<thead>
<tr>
<th>Scenario 1.2</th>
<th>Mean RMSE (m)</th>
<th>Max Error (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2D</td>
<td>3D</td>
</tr>
<tr>
<td>GNSS</td>
<td>42.70</td>
<td>46.94</td>
</tr>
<tr>
<td>GNSS CP EKF</td>
<td>1.17</td>
<td>2.32</td>
</tr>
<tr>
<td>GNSS CP MDPC</td>
<td>0.64</td>
<td>1.59</td>
</tr>
<tr>
<td>GNSS CP SPAWN</td>
<td>0.42</td>
<td>1.43</td>
</tr>
<tr>
<td>GNSS CP MDSPAWN</td>
<td>0.38</td>
<td>1.44</td>
</tr>
</tbody>
</table>

Table 5.9: Mean RMSE and maximum error derived from scenario 1.2

In scenario 1.3, a more challenging set-up was simulated: among the two available
GNSS signals, one was corrupted due to multipath. The results are presented in table 5.10. As expected, when the navigation system operated in GNSS standalone only, its solutions became highly erroneous as there were simply not enough redundancies to correctly execute FDE. The mean 2D RMSE was averaged at 98.45 m while the positioning error exceeded 272 m. When CP was employed, the error was substantially reduced. For example, mean 2D RMSEs were reduced to 8.13 m and 3.20 m when using EKF and MDPC, respectively. Similarly, both SPAWN and MDSPAWN also managed to reduce the error to an average of 8.56 m and 5.15 m, respectively. However, it is important to note that while the errors were reduced, they were still considerably larger than the ones obtained in the previous sub-scenario.

<table>
<thead>
<tr>
<th>Scenario 1.3</th>
<th>Mean RMSE (m)</th>
<th>Max Error (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2D</td>
<td>3D</td>
</tr>
<tr>
<td>GNSS</td>
<td>98.45</td>
<td>81.31</td>
</tr>
<tr>
<td>GNSS CP EKF</td>
<td>8.13</td>
<td>8.62</td>
</tr>
<tr>
<td>GNSS CP MDPC</td>
<td>3.20</td>
<td>5.16</td>
</tr>
<tr>
<td>GNSS CP SPAWN</td>
<td>8.56</td>
<td>15.51</td>
</tr>
<tr>
<td>GNSS CP MDSPAWN</td>
<td>5.15</td>
<td>7.60</td>
</tr>
</tbody>
</table>

Table 5.10: Mean RMSE and maximum error derived from scenario 1.3

There are two reasons for the problem of navigation solutions inaccuracy, even when CP ranges were available in scenario 1.3. The first is that there were simply not enough high-quality measurements to provide good estimates; CP only provided three additional ranging, two from RSUs and one from the neighbouring vehicle, as depicted in figure 5.20. It is important to note that the neighbouring vehicles position quality had a direct impact on the target vehicles position. As such, if the position of the neighbouring vehicle was of low quality, then it would adversely affect the quality of the target vehicles position. Another probable reason for this problem is that the CP set-up, particularly with regard to the placements of the RSUs, was not geometrically optimal, resulting in an unfavourable network geometry. Consequently, it was more difficult for the estimators to correctly estimate the true positions. This was confirmed by the computed CRB value, averaged across the trajectory with simulated outages and multipath effects, of
5.75 m and 34.63 m for the 2D and 3D axes, respectively. A better network geometry would require the RSUs to be placed elsewhere, where they would cover both sides of the vehicles. For example, if RSU 1 was placed on the other side of the road, as depicted in figure 5.21b, as opposed to the default set-up, depicted in figure 5.21a, the achievable RMSE calculated using CRB would be 2.74 m and 13.84 m for the 2D and 3D axes. Alternatively, higher accuracy could be achieved if the prior information was not impaired by the effects of multipath, i.e., the prior information was of high quality. This was seen in sub-scenarios 1.1 and 1.2, where the results obtained were significantly better.

Scenario 2

As shown in the analysis of scenario 1, the GNSS standalone solutions did improve when CP was employed. However, when there was limited GNSS availability and signals were corrupted due to multipath effects, the solutions degraded significantly. The analysis of the second scenario will prove that the inclusion of MSS could significantly alleviate this problem. As shown in the previous chapter, MSS could continuously limit the error growth of the positioning error by providing measurement redundancies and correctly rejecting faulty signals, hence improving the overall navigation solutions. In turn, MSS could help the CP algorithms to better solve for the estimates.

In scenario 2.1, apart from the inclusion of MSS, the same set-up as scenario 1.3 was used where only two GNSS signals were available, one of which was corrupted due to multipath and CP ranging from two RSUs and between vehicles. Table 5.11 lists the results of the various navigation modes in this scenario. The MSS recorded a mean 2D
RMSE of 2.59 m, an improved solution of 98% compared to the GNSS standalone solution. When CP was applied, the positioning solutions further improved, with the EKF estimator producing the best result of 1.21 m. The 2D RMSEs for MDPC, SPAWN, and MDSPAWN floated around the same values, ranging from 1.63 to 1.69 m. Overall, it can be seen that there was a huge improvement of CP when MSS was fused into the navigation system. The mean 2D RMSE improvements of MSS CP over GNSS CP (table 5.10), using the various estimators, averaged at around 70%. The maximum error also decreased by about 80%, thus proving the effectiveness of MSS in improving CP and in providing better navigation solutions in GNSS-difficult environments. When comparing the different estimators, however, on average the EKF unexpectedly performed better than the other estimators, including MDPC and MDSPAWN. This might be due to this particular set-up and network geometry. Figure 5.22 demonstrates the effectiveness of MSS CP, represented as the green line, over GNSS CP, represented as the red line in this scenario. In the region after 225 s, the solutions from GNSS CP indicate that CP did help in trying to pull the position back closer to the true position. However, because the CP operated at around 1 Hz while the GNSS operated at 5 Hz, the position uncertainty grew when CP was not present, hence resulting in the spiky trend of the red line. When MSS, which operated at 100 Hz, was utilised, it actively prevented the solutions from diverging from their true positions, and this resulted in a superior navigation solution.

<table>
<thead>
<tr>
<th>Scenario 2.1</th>
<th>Mean RMSE (m)</th>
<th>Max Error (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2D</td>
<td>3D</td>
</tr>
<tr>
<td>MSS</td>
<td>2.59</td>
<td>4.77</td>
</tr>
<tr>
<td>MSS CP EKF</td>
<td>1.21</td>
<td>2.22</td>
</tr>
<tr>
<td>MSS CP MDPC</td>
<td>1.69</td>
<td>3.19</td>
</tr>
<tr>
<td>MSS CP SPAWN</td>
<td>1.67</td>
<td>3.24</td>
</tr>
<tr>
<td>MSS CP MDSPAWN</td>
<td>1.63</td>
<td>3.09</td>
</tr>
</tbody>
</table>

Table 5.11: Mean RMSE and maximum error for scenario 2.1

In scenario 2.2, a more challenging scenario was set-up. Now, each vehicle could only receive ranging from one of the RSUs, while maintaining their V2V ranging, as depicted in 5.23. This results in the CP being less effective, as the aiding from RSUs were reduced.
from two measurements to only one for each vehicle. This scenario was created to observe the effects of reducing the number of RSUs on positioning solutions of VANET. This analysis might be helpful in trying to provide a brief insight when dealing with large scale networks as it is not always practical to allow for every vehicles to connect to all neighbouring RSUs and neighbouring vehicles due to physical communication limitations. Table 5.12 lists the resulting solutions of the navigations systems in this scenario. Note that MSS have been excluded as its solutions are invariably equal to the one from scenario 2.1. Overall, all of the estimators performed similarly, with the MDPC provided the best mean 2D RMSE of 1.60 m. When compared to the results from scenario 2.1, there was an increase in positioning error, particularly when using EKF where its mean 2D RMSE and maximum error increased by 38% and 30% respectively. Similarly, using SPAWN resulted in a positioning error increase of 7%. MDSPAwn on the other hand virtually did not have any changes for its mean 2D RMSE but did suffer a slight increase of its maximum error. Surprisingly, MDPC provided slightly better solutions, with an increase of 5% of both mean 2D RMSE and maximum error. In terms of absolute values however, this percentage equates to 0.09 m and 0.13 m, hence are not considered to be of any particular significance.

Scenario 2.2 was more challenging. Here, each vehicle could only receive ranging from one of the RSUs while maintaining their V2V ranging, as depicted in 5.23. This resulted in the CP being less effective, as the aiding from the RSUs was reduced from two measurements to only one for each vehicle. This scenario was created to observe
the effects of reducing the number of RSUs on the positioning solutions of a VANET. This analysis might provide insight when dealing with large-scale networks, as it is not always practical to allow every vehicle to connect to all neighbouring RSUs and neighbouring vehicles due to physical communication limitations. Table 5.12 lists the resulting solutions of the navigations systems in this scenario. Note that MSS has been excluded as its solutions are invariably equal to those from scenario 2.1. Overall, all of the estimators performed similarly, with the MDPC providing the best mean 2D RMSE of 1.60 m. When compared to the results from scenario 2.1, there was an increase in positioning error, particularly when using EKF, whose mean 2D RMSE and maximum error increased by 38% and 30%, respectively. Similarly, using SPAWN resulted in a positioning error increase of 7%. On the other hand, MDSPAWN showed virtually no changes to its mean 2D RMSE, but did suffer a slight increase in its maximum error. Surprisingly, MDPC provided slightly better solutions, with an increase of 5% of both mean 2D RMSE and maximum error. In terms of absolute values, however, this is equivalent to 0.09 m and 0.13 m, and is thus not considered to be of any particular significance.

Figure 5.23: V2X connections in scenario 2.2

Overall, the inclusion of MSS again proved to be beneficial in optimising CP in difficult GNSS environments. Unlike non-radio ranging, the radio ranging obtained using UWB transceivers is not susceptible to multipath effects, which increases the capability of
Table 5.12: Mean RMSE and maximum error for scenario 2.2

<table>
<thead>
<tr>
<th>Scenario 2.2</th>
<th>Mean RMSE (m)</th>
<th>Max Error (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2D</td>
<td>3D</td>
</tr>
<tr>
<td>MSS CP EKF</td>
<td>1.67</td>
<td>3.11</td>
</tr>
<tr>
<td>MSS CP MDPC</td>
<td>1.60</td>
<td>2.94</td>
</tr>
<tr>
<td>MSS CP SPAWN</td>
<td>1.79</td>
<td>3.31</td>
</tr>
<tr>
<td>MSS CP MDSPAWN</td>
<td>1.63</td>
<td>2.75</td>
</tr>
</tbody>
</table>

CP to operate in dense urban environments. Using real datasets collected in Melbourne, Australia, the proposed radio ranging CP integrated with MSS was tested in various scenarios. The results, which also compared various CP estimation techniques discussed in chapter 3, showed that MSS could not only reduce the positioning error of vehicles in a VANET, but it could also maintain adequate navigation solutions even when their connectivity to RSUs was reduced.

5.4 Summary

Chapter 5 has analysed the integration of MSS and CP using real datasets, including both non-radio-ranging and radio-ranging-based CP. The first part of this chapter showed that fusing MSS and shared raw GNSS observations to form V2V double difference ranging in CP could significantly enhance the positioning accuracies of the participating vehicles in VANET. MSS, which provided measurement redundancies and continuous positioning, enabled the CP to continuously operate even when there were fewer than the minimum number of satellites needed in a conventional CP system, which is useful in environments that suffer from GNSS signal shadowing. However, this technique is not suitable for use when GNSS signals are affected by multipath, which invariably affects the V2V ranging. This leads to the second part of this chapter, where radio ranging obtained via UWB transceivers was used to form the V2X ranges. This part of the chapter analysed MSS and CP utilising radio-based ranging instead of shared raw GNSS observations. The UWB was chosen as a candidate to form the V2X ranging as it does not suffer multipath effects, thus making it useful in dense urban environments. As with the results from the first
part of this chapter, the inclusion of MSS in CP did increase the positioning accuracy of the participating vehicles in VANET. Furthermore, the results showed that MSS enabled vehicles to reduce their reliance on V2X connectivity. In turn, this creates the possibility to optimise message passing scheduling in the VANET to acquire V2X ranging, thus further optimising CP. While promising results were obtained from the analysis in this chapter, it should be noted that they were only acquired using small-scale CP datasets. Hence, the next chapter presents the MSS CP analysis on a larger scale, using realistic simulated datasets.
Chapter 6
Multi-Sensor System and Cooperative Positioning: Analysis Using Realistic Simulated Datasets

6.1 Introduction

The MSS is clearly beneficial when integrated into CP, as discussed in the previous chapter. Using real datasets, promising results were obtained, where achieved mean RMSEs were all under 2 m when GNSS outage and multipath were present. However, the datasets only consisted of a small-scale VANET due to the logistical challenge faced in this study. Hence, the present chapter analyses the exact same implementations as the previous chapter but with a larger-scale VANET. This is achieved by simulation. The availability of large-scale data opens up possibilities for various kinds of analysis, for example regarding the effect of scalability and network geometry. However, to keep this thesis concise and to ensure that it is in line with the main issues raised in chapter 1, only the following core questions are addressed in this chapter:

i. Algorithms - How do the proposed algorithms fare compared to the conventional ones in a large-scale VANET? Does their performance match the results in the previous chapter?

ii. MSS effect on CP - The major focus of the study is how MSS can improve CP. It has been demonstrated in earlier chapters that MSS does indeed improve CP, but this was investigated using small-scale datasets. In this chapter, the potential improvement when employing MSS part of CP is discussed using larger datasets.
This chapter first elaborates on the programs involved in generating realistic VANET simulations. This is followed by the simulation set-up, which details the VANET V2X connections, vehicle trajectory, and GNSS outage and multipath characteristics. Similar to the other chapters, the analysis methodology used in this study is then outlined, followed by a discussion of the results.

6.2 Generation of a VANET Simulation

In this study, VANETs are simulated using two programs: SUMO and GEMV2. As depicted in figure 6.1, a base map is first extracted from OpenStreetMap. The map is then fed as an input into the SUMO program, where it will generate road networks and building blocks. The vehicular traffic can then be generated, and users are able to configure the simulation set-up such as the simulation period and traffic types, density, and speed. In this study, private cars, buses, and trucks are generated to add realism to the simulation. The addition of these types of vehicles makes signal strength variation become more realistic, as their bodies will be taken into account when calculating RSS. SUMO outputs three types of files: road network, vehicular traffic flow and routes, and building polygons. This information is fed into GEMV2 to generate realistic V2X connections. This generation is done using a realistic signal propagation model such as the multiple knife-edge based on the ITU-R method [3], which takes into account signal diffraction caused by surrounding vehicles and buildings. Subsequently, the output from SUMO and GEMV2 is treated as input in the CP program created by the author in Matlab. To complete the VANET simulation, IMU, VSS, and GNSS signals are generated for each participating vehicle in the network. The following subsections provide more informa-
6.2 Generation of a VANET Simulation

6.2.1 Simulation of Urban Mobility (SUMO)

SUMO is an open source microscopic traffic simulation suite that can simulate traffic and hence vehicular positions from a defined road network. It was first developed by the German Aerospace Center (DLR) in 2001 as a simple tool for traffic simulation, and it has since then evolved into a full suite of traffic modelling utility.

The simulation begins with an input of a predefined road network. SUMO is able to acquire map input from multiple sources namely, VISUM, Vissim, MATsim, and Open Street Map (OSM). In this study, the OSM is used as the map source for SUMO as it allows for the extraction of existing road networks. In addition to the road network, building blocks are also generated in SUMO, using the input from OSM. This adds further realism to the simulation and establishes connectivity between vehicles and RSUs. The input from OSM is then converted into a proprietary format of road network and polygons representing buildings.

The next step is to generate traffic within the established SUMO network; this can be done in several ways. The simplest method is to generate original/destination (O/D) matrices, where an origin and destination are assigned for each vehicle. For larger-scale
simulations, however, these O/D matrices are converted into vehicle trips, which allows for the movements of the vehicles to be more complicated in that they can change lanes and travel along multiple roads (referred to as edges). When starting a simulation, each vehicle is assigned a start/end position and a departure time, but with no explicit route information. The route for each vehicle is calculated by traffic assignment, using routing procedures such as the shortest path calculation with varying cost functions. Other routing procedures are also available, such as dynamic routing. Here, edges with a high density of vehicles are often avoided to negate traffic congestions. In this study, the simulation parameters are set so that the vehicles stay within the simulation by an average of more than 10 minutes through looping, to create traffic build-up at the later stages of the simulation.

The SUMO suite can provide various simulation outputs, which are generated during simulation run-time. In this study, the parameters are set such that SUMO outputs vehicle ID, position, velocity, and heading during each time step. To complete the VANET simulation, this information, along with the SUMO road network and building polygons, is then passed onto GEMV2 to generate V2X connections. This is further described in the next subsection.

6.2.2 GEMV2

GEMV2 is a geometry-based, efficient propagation model designed to establish realistic V2X connections for CP applications [16]. Unlike conventional geometry-based models, such as the ray tracing method, it uses geographical information about the simulated environment, which includes outlines of buildings, foliage, and vehicles, to generate links between the vehicles. This makes it more efficient in large-scale simulations where a large number of vehicles are present. The connections between vehicles are then classified into three groups: LOS, NLOS due to surrounding vehicles, and NLOS due to surrounding buildings and foliage. The distances of established connections in the respective classes are then calculated with propagation effects including transmission (propagating through material), diffraction, and reflection. In addition, path loss and large-scale and small-scale fading mechanisms are incorporated into the calculation of the connec-
6.3 Simulation set-up

In this study, two scenarios are simulated to test the developed algorithms discussed in the previous chapters. The first is an open sky environment, and the second is an environment with dense building blocks. Due to the nature of scenario 1, which allows for full GNSS availability and more V2X connections, its results are used as a benchmark for the subsequent scenario, where GNSS signals and V2X connections are typically lower due to signal obstructions caused by the surrounding buildings. The following subsections elaborate on the two simulated environments used to test the developed CP system and the generation of simulated MSS measurements.

6.3.1 Scenario 1 - Open Sky Environment

The vicinity of Albert Park, Australia was chosen as the simulation test bed for the open sky environment. The section of road chosen is a dual carriageway with a length of 1.1
### Table 6.1: Summary of the different types of simulated scenarios

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Open Sky Environment</td>
</tr>
<tr>
<td>2</td>
<td>Dense Urban Environment</td>
</tr>
</tbody>
</table>

The simulation lasts around five minutes, with the participation of a total of 182 unique vehicles. Four RSUs are placed along the trajectory, where a vehicle might only receive up to two V2I connections at any given time due to the distance of the vehicle from the four RSUs. The simulation starts with one departing vehicle entering from the trajectory’s north-west edge, while the subsequent vehicles enter from either edge.

As depicted in the left subplot of figure 6.4, the number of vehicles present increases over time until it reaches 74. From that point on, it fluctuates with a decreasing trend until the end of the simulation. The right subplot of the same figure shows the averaged velocity of all vehicles, and the maximum and minimum velocities of the fastest and slowest vehicles at any particular time step throughout the simulation. The average velocity of the vehicles is around 38 km/h, indicating smooth traffic along the trajectory. Figure 6.5 shows the number of vehicles with velocity within a given bracket: one is above 40
6.3 Simulation set-up

Figure 6.4: Number of vehicles and V2X connections for scenario 1

km/h, and the other below 10 km/h. This figure demonstrates that this trajectory has a relatively smooth traffic flow over the period of simulation.

Figure 6.5: Number of vehicles with velocity threshold

Figure 6.6 visualises the connected vehicles produced via GEMV2. The rings on each pillar represent the number of connections, while the colours indicate the signal strength of the V2X measurements. The vehicles in this scenario are highly interconnected, with an average number of around 10, due to the lack of obstructive objects such as buildings and thick foliage in the environment. The generated ranges are set to have simpler but similar characteristics to the ones discussed in subsection 5.3.2, i.e., measurements with no bias and a standard deviation of 0.25 m.

In both scenarios, seven GNSS satellites are simulated throughout the entire trajec-
6.3.2 Scenario 2 - Dense Urban Environment

For the dense urban environment, a segment of the Melbourne CBD was chosen as the area of interest. It has a total length of about 2.5 km, with a majority of dual carriageways and a few single ones. As depicted in figure 6.7, it has a typical dense urban grid-like structure and consists of four major intersections. The simulation lasts approximately 17 minutes. In addition, four RSUs are generated in this simulation and placed at the four major intersections. All of the RSUs are placed at the centre of the intersections and 7 metres above ground to maximise visibility. Similar to the previous scenario, this one starts with one departing simulation, and the number of participating vehicles increases over time. Unlike scenario 1, however, some of the vehicles in scenario 2 are introduced...
6.3 Simulation set-up

in the centre of CBD, i.e. they do not start at the outer edges of the road network. This introduces a new challenge: the initial positions of these vehicles are heavily corrupted due to signal shadowing and multipath. The following section investigates this effect using conventional and proposed CP algorithms. At the end of the simulation, the traffic count is high, resulting in massive congestion where most of the vehicles do not move. This configuration is set to simulate typical traffic in urban environments during peak hours. In total, 950 unique vehicles are generated in this scenario.

The vehicle kinematics in this scenario are designed to differ from those in the previous one: as depicted in the right sub-figure 6.8, the averaged velocity is gradually reduced from 30 km/h at the start of the simulation to almost 0 km/h at end of the simulation. Figure 6.9 reveals that the majority of vehicles have less than 10 km/h velocity, and only a few have more than 40 km/h velocity. As mentioned earlier, this is intentional as it depicts a scenario during peak hour traffic.

Figure 6.10 represents connected vehicles on Google Earth™. Unlike scenario 1, the connectivity is slightly higher here due to the concentration of vehicles within the CBD. Compared to figure 6.6, there are more red rings in this figure, indicating that there are
more signals with lower signal strength. This is expected due to the surrounding high rise buildings. Figure 6.11 shows the number of V2X connections, the averaged V2X distance in metres, the number of unique vehicles, and the connection density throughout the simulation. It can be observed that the number of V2X connections increases over time as more vehicles are introduced into the simulation. Interestingly, the averaged distance seems to decrease at the start and stabilise after 200 s. This is because the vehicles become more concentrated as there is massive traffic build-up; hence, the derived V2X distances become shorter.

Similar to in scenario 1, seven GNSS satellites are simulated here. However, the number of pseudoranges is significantly reduced when vehicles travel along routes that sur-
rounded by buildings, and some of the remaining satellites are corrupted due to multipath effects. The resulting number of satellites is simulated to be close to the findings in subsection 2.3.3, while the multipath effects are simulated using the same methodology discussed in subsection 4.4.3. The average number of GNSS satellites visible throughout
the simulation is depicted in figure 6.12, which shows the satellite visibility for vehicle ID 185. Figure 6.12 shows that the number of satellites visible in the first 30 seconds is relatively good compared to in the remaining time. This is due to the lower number of vehicles introduced in the simulation, the majority of which have adequate satellite visibility. The number degrades quite rapidly when 400 s is reached due to the large number of vehicles concentrated in the areas where high rise buildings are present. The average number of visible satellites is three.

Figure 6.12: Averaged satellite visibility throughout scenario 2

A closer look at figure 6.13a reveals a typical characteristic of satellite visibility when traversing dense urban environments. The trend in this graph is also similar to the one seen in the test conducted in Melbourne CBD, illustrated in figure 2.10, where the number of visible satellites was relatively low. Note that the vehicle is introduced at 185 seconds and removed at 920 seconds; thus, only data from this period is available in the figure. As mentioned earlier, multipath is also simulated in this scenario using the methodology discussed in subsection 4.4.3. Figure 6.13b depicts the resulting multipath effect on pseudoranges for a particular vehicle ID 185. It can be seen that in the first 60 seconds, all four pseudoranges are affected by multipath. At around 290 s, they are again affected by multipath for around 280 seconds, followed by the last set of multipath effects lasting around 200 seconds. The vehicle is outside the area with high rise buildings in the last 120 seconds, which results in better pseudoranges, as shown in the figure.
6.4 Analysis Methodology

Parallel to the aim of this thesis, this chapter highlights how the incorporation of MSS in CP could significantly aid the CP system. Similar to chapter 5, the analysis in this chapter focuses on two major parts for each scenario: the first is evaluating the developed CP algorithms, while the second is investigating the effect of adding MSS in the CP scheme. It is worth noting that the architecture and methodology of the MSS and CP systems are identical to the ones discussed in chapters 3 and 5.

As discussed earlier in this chapter, the analysis for scenario 1 assumes that there are no GNSS outages and multipath effects due to the open sky environment, while it is assumed that vehicles in scenario 2 do suffer from GNSS and V2X signal shadowing and multipath effects. For scenario 1, two outputs (1.1 and 1.2) are discussed. The outcomes highlight whether MSS adds any benefit to the CP system in an open sky environment. Like in the previous chapter, scenario 1.1 compares the achievable accuracy of EKF, SPAWN, MDPC, FSPAWN, and MDSPAWN. The sampling size for algorithms that utilise the sampling method is set to 200 samples for each vehicle. For FSPAWN, two clique sizes were chosen: one contains five and the other contains 10 vehicles. These numbers were chosen to reflect the practicality of forming cliques based on distances: even if the clique is set to have a large number of vehicles, this would be infeasible due to the distances between vehicles often being too large or requiring a large number of
hops when propagating information. Next, scenario 1.2 highlights whether MSS adds to the positioning accuracy of CP when it is utilised in an open sky environment. The characteristics of the satellite visibility and multipath effect in this scenario are described in subsection 6.3.1.

Similarly, scenario 2 is analysed using the same methodology as in scenario 1. This is the focus of this chapter, where a large-scale VANET using the proposed CP system is tested in a dense urban environment. Due to the nature of the environment in which the VANET operates, GNSS visibility is limited while the V2X connections increase, as discussed in subsection 6.3.2. Furthermore, an additional test is included here, called scenario 2.3: to further investigate the benefits of MSS, the number of V2X ranging for each vehicle is reduced to six and four connections only. This is done to examine whether adequate positioning accuracy can be achieved even when the number of V2X connections is reduced, as the system has redundant measurements provided by MSS. To maintain brevity in this thesis, the estimates for MSS are not discussed in detail. It is assumed that their measurement biases are correctly compensated for using the techniques discussed in chapter 4. The next section presents and discusses the results obtained from both scenarios, which are summarised in table 6.2.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Navigation mode</th>
<th>CP ranging</th>
<th>GNSS outage and multipath</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>GNSS only</td>
<td>full V2I, full V2V</td>
<td>Full availability</td>
</tr>
<tr>
<td>1.2</td>
<td>MSS</td>
<td>full V2I, full V2V</td>
<td>Full availability</td>
</tr>
<tr>
<td>2.1</td>
<td>GNSS only</td>
<td>lim. V2I, lim. V2V</td>
<td>Limited availability and faulty</td>
</tr>
<tr>
<td>2.2</td>
<td>MSS</td>
<td>lim. V2I, lim. V2V</td>
<td>Limited availability and faulty</td>
</tr>
<tr>
<td>2.3</td>
<td>MSS</td>
<td>red. V2I, red. V2V</td>
<td>Limited availability and faulty</td>
</tr>
</tbody>
</table>

Table 6.2: Summary of the various scenarios

Similar to the analysis in the previous chapters, the performance of the resulting output is quantified using mean RMSE and maximum error. To maintain brevity, most of the discussions in the results section focus on the 2D component. For sampling-based algorithms, the RMSE is averaged from 30 Monte Carlo realisations.
6.5 Results

This section presents the results obtained from the analysis of the two main scenarios introduced earlier. It discusses the results of scenario 1 first, followed by those of scenario 2.

6.5.1 Scenario 1

Table 6.3 lists the results of all outcomes for scenario 1.1. The RMSE of the standalone GNSS solution is included in the table as a comparison to the other techniques solutions. As expected, it has the lowest accuracy, with a 2D mean RMSE of 1.10 m. When CP is employed, the 2D mean RMSE is reduced marginally, with EKF and SPAWN achieving 0.90 m and 0.83 m, respectively. On the other hand, all proposed algorithms perform significantly better than their counterparts. The decentralised MDSPAWN, partially decentralised FSPAWN (sizes 5 and 10), and centralised MDPC achieve 0.51 m, 0.49 m, 0.48 m, and 0.41 m 2D mean RMSE, respectively. In terms of percentage, the average improvement is around 48%. Similar trends can be observed for the 2D maximum errors, for which the proposed algorithms all yield better values than the conventional ones.

<table>
<thead>
<tr>
<th>Scenario 1.1</th>
<th>Mean RMSE (m)</th>
<th>Max Error (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2D</td>
<td>3D</td>
</tr>
<tr>
<td>GNSS</td>
<td>1.10</td>
<td>1.94</td>
</tr>
<tr>
<td>GNSS CP EKF</td>
<td>0.90</td>
<td>1.60</td>
</tr>
<tr>
<td>GNSS CP SPAWN</td>
<td>0.83</td>
<td>1.56</td>
</tr>
<tr>
<td>GNSS CP MDSPAWN</td>
<td>0.51</td>
<td>1.47</td>
</tr>
<tr>
<td>GNSS CP FSPAWN (5)</td>
<td>0.49</td>
<td>1.48</td>
</tr>
<tr>
<td>GNSS CP FSPAWN (10)</td>
<td>0.48</td>
<td>1.40</td>
</tr>
<tr>
<td>GNSS CP MDPC</td>
<td>0.41</td>
<td>1.20</td>
</tr>
</tbody>
</table>

Table 6.3: Mean RMSE and maximum error derived from scenario 1.1

Table 6.4 lists the outcomes of this scenario when MSS is incorporated into the CP. It reveals that adding MSS in open sky environments only marginally increases the positioning accuracy of the VANET. This is expected since the MSS solution is ultimately
dependent on the external aiding measurements, which in this scenario are GNSS and V2X. The average percentage of improvement is only 5%. However, there is a decrease in the 2D maximum error by an average of 0.41 m, or 19%, which can be considered significant. This might be due to the MSS forcing the outputs to be smoother and constraining noisy solutions. This is depicted in figure 6.14b, where the RMSE over time for all algorithms can be seen to have less noise than the outputs from scenario 1.1. Note that in these plots, G is an abbreviation for GNSS, while M is an abbreviation for MSS.

<table>
<thead>
<tr>
<th>Scenario 1.2</th>
<th>Mean RMSE (m)</th>
<th>Max Error (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2D</td>
<td>3D</td>
</tr>
<tr>
<td>MSS</td>
<td>1.07</td>
<td>1.90</td>
</tr>
<tr>
<td>MSS CP EKF</td>
<td>0.85</td>
<td>1.10</td>
</tr>
<tr>
<td>MSS CP SPAWN</td>
<td>0.69</td>
<td>1.40</td>
</tr>
<tr>
<td>MSS CP MDSPAwn</td>
<td>0.50</td>
<td>1.40</td>
</tr>
<tr>
<td>MSS CP FSPAwn (5)</td>
<td>0.51</td>
<td>1.30</td>
</tr>
<tr>
<td>MSS CP FSPAwn (10)</td>
<td>0.45</td>
<td>1.35</td>
</tr>
<tr>
<td>MSS CP MDPC</td>
<td>0.38</td>
<td>1.28</td>
</tr>
</tbody>
</table>

Table 6.4: Mean RMSE and maximum error derived from scenario 1.2

The collective results from scenario 1 suggest that in an open sky environment, MSS does not provide much improvement in terms of accuracy. However, in line with the findings in previous chapters, there are significant differences in accuracy when different
algorithms are employed. For example, all of the proposed algorithms perform better than conventional algorithms. In addition, MDPC consistently outperforms the other algorithms in this scenario. The obtained results are optimistic, which is expected since this scenario simulates an open sky environment, and since four RSUs are present to further aid the CP system.

6.5.2 Scenario 2

Unlike scenario 1, the simulation in this scenario is more challenging: GNSS availability is limited and often severely corrupted, as discussed in subsection 6.3.2. The results of scenario 2.1 are listed in table 6.5. From the results, it is evident that none of the algorithms, even when CP is employed, can fulfil the required ITS lane-level positioning accuracy. The first clear reason for this is the lack of adequate GNSS measurements, and the second is that the remaining GNSS measurements available are mostly corrupted by multipath, making it more difficult to provide good estimates for the vehicles. The lack of MSS in the system makes it challenging to properly bridge GNSS outages and reject faulty measurements. As expected, the GNSS-only method cannot provide any reliable positioning solutions, with a recorded 2D mean RMSE above 50 m. In contrast, when CP is used, both EKF and SPAWN provide better solutions, but they are still beyond a reasonable level of RMSE. However, the proposed algorithms provide better positioning accuracy, with the best from MDPC at 3.61 m. Although this is a much better solution, it is still beyond the desired level.

Table 6.6 shows the results for scenario 2.2 where the MSS is incorporated into CP. Based on the table, it clear that simply adding MSS does not lead to a more correct positioning solution. The MSS solution, for example, only manages to achieve a mean 2D RMSE of 53 m, while its maximum error exceeds 230 m. This contradicts the findings in subsections 4.4.4 and 5.3.6, where MSS alone was shown to significantly improve the positioning solution. However, there is one major difference between the set-up in that test and the one in scenario 2.2: in the latter, some of the vehicles are not properly initialised due to lack of GNSS and V2I signals. These vehicles begin their trajectory within the central part of the map, which is heavily surrounded by high rise buildings. This makes
it more challenging to achieve ITS standard positioning accuracy. This is analogous to
the problem discussed in subsection 3.2.4, that large prior distribution of the vehicles
makes it more difficult to provide good estimates. This problem is further exacerbated
when the vehicles are mobile. However, as discussed in subsection 3.3.4, the results in
this scenario show that unlike conventional algorithms, the proposed ones performed
much better under this circumstance. For example, the fully decentralised MDSPAWN
can produce a substantially better estimate compared to conventional SPAWN. Similar
to the results from the previous scenario, the MDPC performs the best, with a mean 2D
RMSE of 1.31m. This in turn provides better estimates for the MSS to further improve the
system as a whole.

Table 6.5: Mean RMSE and Maximum Error derived from scenario 2.1

<table>
<thead>
<tr>
<th>Scenario 2.1</th>
<th>Mean RMSE (m)</th>
<th>Max Error (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2D</td>
<td>3D</td>
</tr>
<tr>
<td>GNSS</td>
<td>54.66</td>
<td>93.69</td>
</tr>
<tr>
<td>GNSS CP EKF</td>
<td>32.34</td>
<td>53.65</td>
</tr>
<tr>
<td>GNSS CP SPAWN</td>
<td>9.03</td>
<td>19.13</td>
</tr>
<tr>
<td>GNSS CP MDSPAWN</td>
<td>4.08</td>
<td>7.86</td>
</tr>
<tr>
<td>GNSS CP FSPAWN (5)</td>
<td>5.50</td>
<td>8.59</td>
</tr>
<tr>
<td>GNSS CP FSPAWN (10)</td>
<td>4.28</td>
<td>6.88</td>
</tr>
<tr>
<td>GNSS CP MDPC</td>
<td>3.61</td>
<td>6.71</td>
</tr>
</tbody>
</table>

Table 6.6: Mean RMSE and maximum error derived from scenario 2.2

<table>
<thead>
<tr>
<th>Scenario 2.2</th>
<th>Mean RMSE (m)</th>
<th>Max Error (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2D</td>
<td>3D</td>
</tr>
<tr>
<td>MSS</td>
<td>53.06</td>
<td>93.20</td>
</tr>
<tr>
<td>MSS CP EKF</td>
<td>25.94</td>
<td>51.05</td>
</tr>
<tr>
<td>MSS CP SPAWN</td>
<td>6.98</td>
<td>16.26</td>
</tr>
<tr>
<td>MSS CP MDSPAWN</td>
<td>2.39</td>
<td>2.88</td>
</tr>
<tr>
<td>MSS CP FSPAWN (5)</td>
<td>1.95</td>
<td>6.01</td>
</tr>
<tr>
<td>MSS CP FSPAWN (15)</td>
<td>1.76</td>
<td>6.12</td>
</tr>
<tr>
<td>MSS CP MDPC</td>
<td>1.31</td>
<td>5.58</td>
</tr>
</tbody>
</table>
6.5 Results

The left subplot of figure 6.15a shows the resulting 2D RMSE over time for all algorithms, while the right subplot shows a close-up, where the y-axis has been limited to a range of 0 to 25 m. As listed in table 6.6, the plot shows that the standalone MSS performs the worst while all of the proposed algorithms perform significantly better than the conventional ones. These figures illustrate an example of the effect of high uncertainty of new vehicles entering the simulation: at around 420 s, most of the algorithms react similarly, with the 2D RMSE increasing by some degree and affecting the networks solution. For EKF, the 2D RMSE does not recover to an acceptable level even after a prolonged period. On the other hand, the proposed algorithms, particularly MDPC, manage to minimally restrict the error and provide better navigation solutions.

![Figure 6.15: RMSE over time for scenario 2.2](image)

As discussed in the previous paragraph, the improvement when incorporating MSS into CP is not clear if observed briefly. However, by comparing the results of the proposed system when employing MDSPAWN, FSPAWN, and MDPC to the ones obtained in scenario 2.1, the significance of incorporating MSS becomes clearer. MDSPAWN, for example, improves by 41%; FSPAWN (5 and 10) improves by 64% and 58%, respectively; and MDPC by 63%. Similarly, the 2D maximum error also improves by an average of 19%. This clearly proves that MSS is able to heighten the positioning accuracy when incorporated in CP. There are several reasons for this phenomenon. First, the MSS is able to bridge data gaps between GNSS and V2X measurements as it operates at a 100 Hz, which is much higher than GNSS and V2X measurements, running at 1 and 5 Hz, respectively. This in turn provides a better confidence level, which is essential for the estimation pro-
cess, particularly when employing sample-based algorithms. Moreover, the redundant measurements provided by MSS are crucial in the extended FDE technique, which allows for faulty GNSS measurements to be correctly removed, as demonstrated in subsection 4.4.4.

<table>
<thead>
<tr>
<th>Scenario 2.3</th>
<th>2D Mean RMSE (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>8 Con.</td>
</tr>
<tr>
<td>MSS CP MDSPAWN</td>
<td>2.39</td>
</tr>
<tr>
<td>MSS CP FSPAWN (5)</td>
<td>1.95</td>
</tr>
<tr>
<td>MSS CP FSPAWN (10)</td>
<td>1.76</td>
</tr>
<tr>
<td>MSS CP MDPC</td>
<td>1.31</td>
</tr>
</tbody>
</table>

Table 6.7: Mean RMSE derived from scenario 2.3

Similar to the results obtained in subsection 5.3.6, the inclusion of MSS allowed for the reduction of V2X connections, while able to maintain adequate positioning accuracy. Table 6.7 shows the results of the MDSPAWN, FSPAWN and MDPC when the number of connections received for each vehicle was reduced from the default average of eight, to averages of six and four. It reveals that for this particular scenario, the accuracy only degraded slightly, by 15% when two connections were removed. As discussed earlier, this is due to the continuity provided by MSS measurements, which in turn enables the positioning system to maintain its level of accuracy. Again, this proves the effectiveness of including MSS in VANET in improving positioning solutions in challenging GNSS environments. However, when the connections were further reduced to an averaged four, the navigation solutions degraded rapidly, beyond the acceptable level. This shows that although MSS does provide positioning continuity, making the vehicles become less reliant on V2X connections, ultimately, it still requires adequate number of V2X connections for the networks positioning solutions to retain the desired accuracy.

6.6 Summary

This chapter has demonstrated that the proposed CP system, with MSS included, can alleviate the vehicular positioning accuracy issues that arise when traversing dense urban
environments. The analysis was conducted on two simulated datasets, with realistic simulated V2X ranging obtained using GEMV2. The generation of V2X ranging in GEMV2 accounts for signal propagation effects including transmission (propagating through material), diffraction, reflection, path loss, and large-scale and small-scale fading mechanisms. The first part of this chapter discussed the generation of the VANET simulation in detail. This included the generation of vehicle mobility using SUMO, whose output was passed on to GEMV2 to generate V2X connections. The second part elaborated on the simulation set-up, followed by the analysis methodology used in this chapter. Finally, the results from the analysis on the simulations were discussed. Some of the highlights of the results include the comparison of the proposed algorithms performance, the effects of MSS on improving VANET positioning accuracy, and MSSs ability to reduce the number of V2X connections needed to retain adequate positioning accuracy. In terms of satisfying the ITS positioning requirements listed in table 1.1, the results show that in open sky environments, the proposed techniques are able to meet the where-in-lane-level accuracy, while in dense urban environments only lane-level accuracy can be achieved.
Chapter 7
Conclusion and Further Research

7.1 Introduction

Efficient traffic management, collision avoidance systems, and autonomous vehicle navigation are all part of ITS. As mentioned in [9], one of the core components of a fully functional ITS is accurate positioning (at lane level), which remains a challenge particularly in dense urban environments. This study has presented several methods to try to improve positioning in GNSS-challenging environments using both CP and multi-sensor integration. Main improvements were made via improved CP algorithms, stochastic modelling, a faulty satellite detection and rejection method, and the deployment of multi-sensor integration as part of the CP scheme. Using both simulated and real datasets, the aforementioned improvements were tested, and the results showed that lane-level positioning accuracy is achievable in dense urban environments using existing technologies and techniques proposed in this thesis.

This chapter summarises the research findings and concludes the thesis. Based on these findings, future recommendations are identified and discussed in section 7.4.

7.2 Summary of Research Findings

The main aim of this study was to meet ITS requirements in dense urban environments using CP and multi-sensor integration. From the numerous studies conducted in this thesis, the following conclusions can be drawn:

- Several conventional and new algorithms for CP were tested in this study. Among
the conventional algorithms, the particle-based filter performed the best but, as expected, it was also the most computationally expensive. Three new algorithms, primarily differing in architecture, were proposed and tested. These were fully centralised, fully decentralised, and partially decentralised algorithms. Using simulated and real datasets, the results consistently showed that the centralised MDPC provided superior positioning accuracy compared to the other algorithms. However, it is less suitable for use in dense vehicular networks due to the limitations imposed by current technology in terms of bandwidth and communication protocols. Hence, a more suitable, decentralised algorithm was developed. The MDSPAWN, an improved version of SPAWN, resulted in a better positioning solution compared to conventional SPAWN and most of the centralised algorithms. Finally, the partially decentralised architecture, FSPAWN, is a hybrid of the previous two architectures. The test results showed that its performance was between a fully decentralised and fully decentralised algorithm. The size of formed cliques in this algorithm heavily influenced its performance: if larger cliques were formed, i.e. forming a more centralised network, a better outcome could be expected.

- Stochastic modelling of the integrated navigation system was also improved. This was achieved through better techniques of capturing IMUs stochastic coefficients, obtained when a vehicle is in motion. Unlike conventional methods, which use the AV, the DAV can be applied when the vehicle is moving, thereby better capturing the vehicles dynamics characteristics. Two tests were conducted to test the improvement of using coefficients obtained from DAV vs. conventional AV. When full GNSS coverage was available, there were virtually no improvements when using the coefficients obtained from the DAV. However, when GNSS was obscured, the positioning drift over 30 seconds was improved by several metres, or a 16% improvement, hence proving the DAVs superiority when compared to the results from AV.

- An improved satellite fault detection and exclusion technique was also developed to obtain a positioning system with higher integrity, i.e. a system that would better reject faulty satellite signals. In highly dense areas, satellite signals commonly be-
come faulty when deflected or obstructed by foliage and high rise buildings. The new FDE, called extended FDE, consists of an additional test threshold, the inclusion of a speed sensor, and the application of the non-holonomic constraint. The results showed that extended FDE managed to significantly outperform conventional FDE by at least tens of metres, even when slow growth errors were present. Furthermore, due to the presence of additional sensors, the extended FDE can continue to operate even when the number of visible satellites is below four, making it highly suitable for use in dense urban environments.

- Chapter 5 tested a non-radio CP technique based on GNSS signals coupled with MSS. In this method, the inter-vehicle range was formed using the code-based GNSS signals, exchanged by the vehicles via DSRC. In particular, the double differencing technique was used to calculate the inter-vehicle range. This was chosen due to its practicality and performance value compared to other differencing techniques, such as single and triple differencing. The non-radio-based derived range was tightly coupled with MSS and tested against a navigational-grade MSS. The results showed that this system outperformed the standalone MSS by at least 25% during partial GNSS outages. Furthermore, the inclusion of MSS as part of CP allows for the CP to be fully functional even with limited satellite visibility. However, it should be noted that this technique is not suitable for use in high-multipath environments, due to the inter-vehicle range being formed using the code-based double differencing technique.

- Another outcome of this study was the testing of a CP system based on radio ranging and in an interconnected (V2X) environment. This is particularly useful as non-radio-ranging applications in real conditions are severely limited. In this test, two vehicles and two RSUs were deployed, all interconnected using UWB transceivers. The UWB data was first calibrated to ensure that accurate ranging in outdoor environments was achieved. In this test, five sensors were utilised: GNSS, IMU, speed sensor, DSRC, and UWB transceivers. The data collected from each sensor was time synchronised before being analysed using the various algorithms mentioned in previous chapters, such as EKF, MDPC, SPAWN, and MDSPAWN. Like in the
previous chapters, the system was tested in several scenarios where both full and partial GNSS availability was simulated. One of the highlights from the results was that the introduction of MSS in the CP network enabled each vehicle to become less reliant on V2X ranging. Furthermore, novel algorithms such as MDSPAWN and FSPAWN consistently outperformed conventional ones.

- The last chapter addressed the shortcomings faced in chapter 5, where the analysis was only conducted on small-scale datasets. The results using simulated large datasets mostly concurred with the findings from the smaller datasets: the proposed CP algorithms could perform better than the conventional ones. The addition of MSS also heightened the CP positioning accuracy, particularly in GNSS-challenging environments. Furthermore, its inclusion allows for the reduction of V2X connections while retaining the accuracy achieved when all V2X connections are utilised. This leads to a better VANET communication performance as less connections are needed, i.e. there are fewer congestions in the wireless network.

7.3 Conclusion

The thesis concludes with a reflection on the aim of this study: to improve CP for VANETs in challenging environments by incorporating MSS. This thesis has clearly demonstrated that the inclusion of MSS in the CP significantly improves performance in terms of availability and positioning accuracy of VANETs. In addition, the thesis has proposed an extended satellite FDE method, a new dynamic stochastic IMU modelling technique, and new centralised, decentralised, and federated algorithms that outperform conventional techniques. Through careful analysis using both real and simulated datasets, the proposed approaches were tested, satisfying the aim of this thesis. The objectives of this thesis were also fully satisfied. The first objective was met in chapters 2 and 3, which discussed all existing technologies and investigated the effects of network design, distribution, and geometry on the achievable VANET positioning accuracy. The second objective was to address the limitations of conventional CP algorithms and propose improvements suitable for implementation in VANET. These were heavily discussed in chapters
3, 5, and 6, where the proposed algorithms were developed and compared to conventional ones. The third objective was to improve existing MSS implementation. In this vein, this thesis has proposed two improvements: an extended FDE method and dynamic stochastic IMU modelling, both of which were proven to outperform conventional methods, as discussed in chapter 4. Finally, the last objective was to test the proposed improvements under varying conditions: in open sky, semi-dense, and dense urban environments. This was achieved by collecting real datasets in open sky and semi-dense urban environments. However, real datasets of large-scale VANETs were not available due to logistical limitations faced during the study; therefore, they were simulated using realistic signal propagation techniques, as discussed in chapter 6. It is hoped that the findings of this thesis will motivate further research in realising more accurate positioning solutions for vehicular networks to meet the requirements of ITS, particularly at the where-in-lane level.

### 7.4 Recommendations for Future Studies

Based on the findings of this study, this section proposes several recommendations to further improve the developed CP system. The following outlines some of the recommendations concerning both local and network-based measurements, the inclusion of non-distance-based measurements, devices of V2X communication and ranging, and the architecture of the CP itself.

- The MSS model used in this study employed the error-state KF, in which the errors of the navigational output are placed as the state vector instead of the actual states themselves, also known as total-state KF. The main benefit of error-state KF over the latter is that it is computationally less demanding, hence reducing the need for a faster processing unit and subsequently lowering the power requirement. On the other hand, studies such as [93] and [121] have shown that employment of the total-state KF could provide a slightly superior result over error-state KF, particularly when GNSS observations are limited. However, this approach is computationally more demanding: whereas a typical error-state KF would require 1,500
FLOPS per second, the total-state KF would require around 43,000 FLOPS per second, assuming the MSS is running at 10 HZ. Furthermore, as pointed out in [34], this approach requires the noise parameters to be tuned to suit the characteristics of the trajectory. Nevertheless, although there are challenges in implementing the total-state KF for MSS, modern processing units are typically fast enough to handle relatively demanding computations while consuming a small amount of power. With regard to the free tuning parameters, the characteristics of vehicle trajectories in a typical urban environment should not vary much. Thus, it is possible to make certain assumptions for parameters to be tuned to suit typical urban navigational characteristics.

- In this study, the UWB noise characteristics were assumed to be randomly distributed centred around a certain mean. However, chapter 5 showed that this was not the case: the measurement bias varied across the kinematic test, which could be caused by various factors such as ambient and device temperatures, small fluctuations of voltage, and quantisation of signals. Such characteristic needs to be investigated using techniques such as AV, DAV, and PSD analyses. Using these techniques would allow for the varying types of errors to be properly quantified, which in turn would allow for a more accurate modelling of the measurements to ultimately produce better navigation solutions.

- One of the major points of focus in this study was to address ITSs current technological limitation, which is that a high number of V2X connections or measurements are needed to implement a fully operational CP. This translates to more wireless traffic, which is problematic in large VANETs. This study used MSS to reduce the number of required connections, and allowed CP to operate at an acceptable level of positioning accuracy even when the number of V2X connections was not sufficient. In recent times, however, emerging sensors have become more readily available and more economical due to batch production as demand from automotive industry increases. For example, locally based measurements are obtained from radio and light detection, and ranging systems are emerging sensors integrated in most prototype vehicles, as automotive manufacturers aim to roll out autonomous vehi-
cles in the near future [18, 66, 92]. In addition, network-based measurements can also be integrated into the CP network. Traffic light vehicle detectors and counter and video scanners are such systems that are already in place, particularly in major city intersections, where they are typically used to gather data to better manage traffic congestion. Integration of these various types of data will pose new challenges, which will require improved understanding through further research.

- The communication protocols of DSRC and UWB transceivers have been well established by IEEE standards (IEEE 802.11 and 802.15, respectively). However, ongoing studies such as [63] and [25] have demonstrated that there is still room for improvement. These studies have proposed that it is possible to further optimise communication between vehicles by altering the architecture of the transceivers protocols, particularly for larger wireless networks. In addition, sharing data structure and compression [98] is another ongoing topic of interest. Future designs of wireless communication need to take into consideration what types of data are to be shared. Is there a better compression method that could minimise our digital footprint? To ensure that CP can fully maximise the number of available network-level measurements.

- This study tested newly proposed algorithms such as MDPC, MDSPAWN, and FS-PAWN and showed superior results for CP over conventional algorithms such as KF and SPAWN. For example, besides showing increased accuracy, MDSPAWN was also able to reduce the sampling size of the vehicles, hence increasing performance via reduction of computational cost. However, the usage of this approach for real implementation might still be a challenge as it requires a relatively large amount of data to be shared across the network, even when it is structured in a decentralised manner. Hence, there is a need to investigate how these samples could possibly be further reduced to achieve better computational and communication traffic performance while maintaining current positioning accuracy.

- The generation of simulated data described in subsection 6.5.2 does not take into account the geometry of the V2X connections when reducing its measurements.
In future studies, it would be worth investigating how V2X measurements can be rejected based on their geometry, i.e., directions of the incoming measurements. This would enable a more optimal HDOP, which would then increase the accuracy of the positioning solution.

- The simulation set-up discussed in chapters 4, 5 and 6 dictates that GNSS satellite visibility and multipath effects for each vehicle are dependent on where the vehicles are at any particular time. When they are within a certain area, the vehicles are then assigned outage and multipath effects with randomised magnitudes. More accurate approaches such as ray tracing might be a more realistic method, and should therefore be used when generating GNSS pseudoranges in future studies.

- The analyses in this study were limited to a small sample of real datasets and large sample from a simulated VANET. Further studies are needed to test the proposed CP system with large-scale VANETs using real datasets. However, this requires a tremendous effort and logistical coordination. Inevitably, such tests are still needed as real world conditions are often different than simulated environments. Obstructions such as foliage, other types of vehicles such as trams or trains, and surrounding materials can affect the signal propagation of both GNSS and terrestrial ranging measurements.

- Due to the nature of CP, its successful implementation requires coordination and planning from various parties, including private corporations, government agencies, and the public sector. A framework is needed to address governments policy, technological standards, and user privacy regarding the implementations of CP. Further studies are needed to specifically look into these issues.

- With the exception of the tests discussed in chapters 4 and 5, the evaluation of CP in this thesis was limited to open sky environments with simulated GNSS outages and multipath effects. This was due to the limited availability of proper devices to provide truth data, such as a navigational-grade IMU/GNSS integrated system. Future studies should employ a more independent positioning system that could provide highly reliable positioning solutions for the CP system to be tested in real
dense urban environments.

- Interoperability of devices plays a highly important role in maintaining the integrity of a VANETs wireless signals. This in turns allows for the system to have a continuous operating system without any disruption. Current ranging technologies such as UWB and DSRC(WIFI) have overlapping bandwidths [60]; this poses serious threat in maintaining continuous operation, particularly in dense urban environments, due to signal interference. Hence, further studies and certification standards of the signals need to be established.
Appendix A

GLONASS Positioning

A.1 GLONASS coordinates

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_e$</td>
<td>Ephemerides reference epoch</td>
</tr>
<tr>
<td>$r_{xyz}(t_e)$</td>
<td>Coordinates at $t_e$ in PZ-90</td>
</tr>
<tr>
<td>$v_{xyz}(t_e)$</td>
<td>Velocity at $t_e$ in PZ-90</td>
</tr>
<tr>
<td>$a_{xyz}(t_e)$</td>
<td>Lunar solar gravitational acceleration at $t_e$</td>
</tr>
<tr>
<td>$\tau_n(t_e)$</td>
<td>Satellite clock bias</td>
</tr>
<tr>
<td>$\gamma_n(t_e)$</td>
<td>Satellite relative frequency offset</td>
</tr>
</tbody>
</table>

Table A.1: GLONASS broadcast ephemeris and clock message parameters

A.1.1 Coordinates transformation to an inertial reference frame

The initial conditions provided in the broadcast ephemeris is in the PZ-90 coordinates system. There, they need to be transformed into an absolute (inertial) coordinates system.

\[
\begin{align*}
    r_{xa}(t_e) &= r_x(t_e) \cos(\theta_G) - r_y(t_e) \sin(\theta_G) \\
    r_{ya}(t_e) &= r_x(t_e) \sin(\theta_G) + r_y(t_e) \cos(\theta_G) \\
    r_{za}(t_e) &= r_z(t_e) \\
    v_{xa}(t_e) &= v_x(t_e) \cos(\theta_G) - v_y(t_e) \sin(\theta_G) - \omega_E r_y(t_e)
\end{align*}
\]
\[
\begin{align*}
  v_{xa}(t_e) &= v_x(t_e) \sin(\theta_{Ge}) + v_y(t_e) \cos(\theta_{Ge}) + \omega_{Er}x(t_e) \quad (A.5) \\
  v_{za}(t_e) &= v_z(t_e) \quad (A.6)
\end{align*}
\]

\(a_{xyz}\) are the acceleration components broadcast in the navigation message are the projections of lunar solar accelerations to axes of the ECEF Greenwich coordinate system. They can be transformed to the inertial system using

\[
\begin{align*}
  Jr_{x}x + Jr_{s}x &= a_x(t_e) \cos(\theta_{Ge}) - a_y(t_e) \sin(\theta_{Ge}) \quad (A.7) \\
  Jr_{x}y + Jr_{s}y &= a_x(t_e) \sin(\theta_{Ge}) + a_y(t_e) \cos(\theta_{Ge}) \quad (A.8) \\
  Jr_{x}z + Jr_{s}z &= a_z(t_e) \quad (A.9)
\end{align*}
\]

where \(\theta_{Ge}\) is the sidereal time at epoch \(t_e\)

\[
\theta_{Ge} = \theta_{G0} + \omega_{E}(t_e - 3 Hrs) \quad (A.10)
\]

where \(\omega_{E}\) is the earth’s rotation rate \((0.729211510^{-4} \text{ rad/s})\) and \(\theta_{G0}\) is the sidereal time in Greenwich at midnight GMT of a date at which epoch \(t_e\) is specified.

\[
\theta_{G0} = 6^h41^{m}50^{s}.54841 + 8640184.812866T_u + 0.093104T_u^2 - 6.210^{-6}T_u^3 \quad (A.11)
\]

\[
T_u = \frac{\text{Julian UT1 date} - 2451545.0}{36525} \quad (A.12)
\]

**A.1.2 Numerical integration**

\[
\begin{align*}
  \frac{dr_{xa}}{dt} &= v_{xa} \quad (A.13) \\
  \frac{dr_{ya}}{dt} &= v_{ya} \quad (A.14) \\
  \frac{dr_{za}}{dt} &= v_{za} \quad (A.15)
\end{align*}
\]
A.1 GLONASS coordinates

\[
\frac{dv_{x_a}}{dt} = -\bar{\mu} r_{x_a} + \frac{3}{2} C_{20} \bar{r}_{x_a} \rho^2 (1 - 5 \bar{r}_{z_a}^2) + Jr_{x_a} m + Jr_{x_a} s
\]  
(A.16)

\[
\frac{dv_{y_a}}{dt} = -\bar{\mu} r_{y_a} + \frac{3}{2} C_{20} \bar{r}_{y_a} \rho^2 (1 - 5 \bar{r}_{z_a}^2) + Jr_{y_a} m + Jr_{y_a} s
\]  
(A.17)

\[
\frac{dv_{z_a}}{dt} = -\bar{\mu} r_{z_a} + \frac{3}{2} C_{20} \bar{r}_{z_a} \rho^2 (3 - 5 \bar{r}_{z_a}^2) + Jr_{z_a} m + Jr_{z_a} s
\]  
(A.18)

\[
\frac{dv_{x_b}}{dt} = -\bar{\mu} r_{x_b} + \frac{3}{2} C_{20} \bar{r}_{x_b} \rho^2 (1 - 5 \bar{r}_{z_b}^2) + Jr_{x_b} m + Jr_{x_b} s
\]  
(A.19)

where

\[
\bar{\mu} = \mu / r^2, \quad \bar{r}_{xyz} = r_{xyz} / r
\]  
(A.20)

\[
\bar{\rho} = a_E / r
\]  
(A.21)

\[
r = \sqrt{r_{x_a}^2 + r_{y_a}^2 + r_{z_a}^2}
\]  
(A.22)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a_E = 6378.136\text{km})</td>
<td>Equatorial radius of Earth (PZ-90)</td>
</tr>
<tr>
<td>(\mu = 398600.44\text{km}^3/\text{s}^2)</td>
<td>Gravitational constant (PZ-90)</td>
</tr>
<tr>
<td>(C_{20} = -1082.63 \times 10^{-6})</td>
<td>Second zonal coefficient of spherical harmonic expression</td>
</tr>
</tbody>
</table>

Table A.2: Definition

A.1.3 Coordinate transformation

The coordinates obtained from the numerical integration of the equations of motion need to be transformed back to Earth-fixed frame PZ-90 using:

\[
r_x(t) = r_{x_a}(t) \cos(\theta_G) + r_{y_a}(t) \sin(\theta_G)
\]  
(A.23)

\[
r_y(t) = -r_{x_a}(t) \sin(\theta_G) + r_{y_a}(t) \cos(\theta_G)
\]  
(A.24)

\[
r_z(t) = r_{z_a}
\]  
(A.25)
PZ-90 to WGS-84 transformation

To metre level accuracy for PZ-90 to ITRF97, prior to PZ-90.02 (20.9.2007)

\[
\begin{bmatrix}
    r_x \\
    r_y \\
    r_z
\end{bmatrix}_{\text{ITRF97}} = \begin{bmatrix}
    r_x \\
    r_y \\
    r_z
\end{bmatrix}_{\text{PZ-90}} + \begin{bmatrix}
    -3\text{ppb} & -353\text{mas} & -4\text{mas} \\
    353\text{mas} & -3\text{ppb} & 19\text{mas} \\
    4\text{mas} & -19\text{mas} & -3\text{ppb}
\end{bmatrix} \begin{bmatrix}
    r_x \\
    r_y \\
    r_z
\end{bmatrix}_{\text{PZ-90}} + \begin{bmatrix}
    0.07 \\
    -0.00 \\
    -0.77
\end{bmatrix} \tag{A.26}
\]

For PZ-90 to ITRF2000, post (20.09.2007):

\[
\begin{bmatrix}
    r_x \\
    r_y \\
    r_z
\end{bmatrix}_{\text{ITRF2000}} = \begin{bmatrix}
    r_x \\
    r_y \\
    r_z
\end{bmatrix}_{\text{PZ-90}} + \begin{bmatrix}
    -0.36 \\
    0.08 \\
    0.18
\end{bmatrix} \tag{A.27}
\]
Appendix B

WGS84 Parameters

This section lists the WGS84 parameters, which are used in developing INS mechanizations and calculation of GNSS solutions.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>WGS 84</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semi Major Axis $a$</td>
<td>m</td>
<td>6,378,137</td>
</tr>
<tr>
<td>Earth’s Angular Velocity $\omega_{ie}$</td>
<td>$10^{-11} rad/s$</td>
<td>7,292,115</td>
</tr>
<tr>
<td>Geocentric Gravitational Const. GM</td>
<td>$10^{-8} m^3/s^2$</td>
<td>3,986,005</td>
</tr>
<tr>
<td>Reciprocal Flattening $1/f$</td>
<td>-</td>
<td>298.257 233 563</td>
</tr>
<tr>
<td>Semi Minor Axis $b$</td>
<td>m</td>
<td>6,356,752.3142</td>
</tr>
<tr>
<td>Normal Gravity at Equator $\gamma_e$</td>
<td>$m/s^2$</td>
<td>9.780 325 3359</td>
</tr>
<tr>
<td>Normal Gravity at Pole $\gamma_p$</td>
<td>$m/s^2$</td>
<td>9.832 184 9378</td>
</tr>
<tr>
<td>First Eccentricity $e^2$</td>
<td>-</td>
<td>0.006 694 379 990 14</td>
</tr>
</tbody>
</table>

Table B.1: WGS84 Parameters [14]
Appendix C

FSPAWN Clique Determination

The architecture of Federated SPAWN (FSPAWN) algorithm used in this thesis relies on the segregation of participating vehicles within the VANET. The pseudo-algorithm codes below briefly lists the steps taken when grouping neighbouring vehicles to create cliques within a network.

The main function is as listed in Algorithm 5. Here the number of cliques is determined by the size of the network divided by the predetermined size of each clique. The global connection information are then copied into each clique and one of the vehicles within each clique will be nominated as a reference vehicle (refN). Then the program will group neighbouring vehicles into cliques f(Grouping nodes into cliques). The final step is to check if any vehicles were left out, which will then be added to the their closest clique f(Check remaining nodes).

Algorithm 5 Main

1: Determine number of clique = CL, based on sC
2: Copy connections from Sim to clique
3: Set refN = 1
4: f(Grouping nodes into cliques)
5: f(Check remaining nodes)
Algorithm 6 \( f \) (Grouping nodes into cliques)

1: for cl = 1 to CL do

2:    if cl > 1 then

3:       for i = 1:cl-1 do

4:          if ispresent(ref_N) then

5:             refN = refN + 1

6:          end if

7:       end for

8:    end if

9:    if cl = 1 then

10:       if s_C > s_Cn then

11:          s_C = s_Cn

12:       end if

13:       clq_cur = [ref_N, conn_N(1:s_C-1)];

14:    else

15:       for i = 1:cl-1 do

16:          if ispresent(Nodes, Clq(i)) then

17:             conn_N(flg) = []

18:          end if

19:       end for

20:       if s_C > s_Cn then

21:          s_C = s_Cn

22:       end if

23:       clq_cur = [ref_N, conn_N(1:s_C-1)];

24:    end if

25:    ref_n = ref_n + 1;

26:    Clique.clique(i) = clq_cur;

27: end for
Algorithm 7 $f$(Check remaining nodes)

1: $N = 1:n N$ \hfill $\triangleright$ N - all nodes
2: $\textbf{for } cl = 1 \text{ to } CL \textbf{ do}$
3: \hspace{1em} $\textbf{if } \text{ispresent}(N, \text{Clique}(cl)) \textbf{ then}$ \hfill $\triangleright$ Identify used nodes
4: \hspace{2em} $N(idx) = []$ \hfill $\triangleright$ Delete used nodes
5: \hspace{1em} $\textbf{end if}$
6: $\textbf{end for}$
7: $\textbf{for } i = 1 \text{ to } \text{size}(N) \textbf{ do}$ \hfill $\triangleright$ Remaining N
8: \hspace{1em} $\text{conn}_N = \text{Nodes.conn}_N(N(i));$ \hfill $\triangleright$ Declare connection node
9: $\textbf{for } j = 1 \text{ to } n, \text{CL \ do}$
10: \hspace{2em} $\textbf{if } \text{ispresent}(N, \text{Clique}(cl)) \textbf{ then}$ \hfill $\triangleright$ Identify used nodes
11: \hspace{3em} $N(idx) = []$ \hfill $\triangleright$ Delete used nodes
12: $\textbf{end if}$
13: $\textbf{end for}$
14: $\textbf{end for}$
Appendix D

Logarithmic Likelihood

When computing the measurement likelihood, it is more convenient to work with the natural logarithm of the likelihood function when dealing with large set of measurements. This is to avoid getting a product value of 0, which happens when the likelihood are too small or when there are a large number of measurements. To begin, equation 3.50 can be represented in logarithmic form as,

$$\ln l(z|x) = \sum_{t=1}^{T} \ln(N(z_t; h_t(x, y), \sigma^2))$$  \hspace{1cm} (D.1)$$

Using

$$\ln(xy) = \ln(x) + \ln(y)$$  \hspace{1cm} (D.2)$$

$$\ln(x/y) = \ln(x) - \ln(y)$$  \hspace{1cm} (D.3)$$

and, recall that the PDF of normal distribution is

$$l(d|x) = (2\pi)^{-T/2} |\Sigma|^{-1/2} \exp \left( -\frac{1}{2} (d_t - h_t(x))^T \Sigma^{-1} (d_t - h_t(x)) \right)$$  \hspace{1cm} (D.4)$$

then

$$\ln l(d|x) = -\frac{T}{2} \ln(2\pi) - \frac{1}{2} \ln(|\Sigma|) - \frac{1}{2} (d_t - h_t(x))^T \Sigma^{-1} (d_t - h_t(x))$$  \hspace{1cm} (D.5)$$

thus,

$$\ln(l(d|x)/\hat{l}(d|x)) = -\frac{1}{2} (d_t - h_t(x))^T \Sigma^{-1} (d_t - h_t(x)) + \frac{1}{2} (d_t - \hat{h}_t(x))^T \Sigma^{-1} (d_t - \hat{h}_t(x))$$  \hspace{1cm} (D.6)$$
or for independent variables,

\[ \ln \left( \frac{I(d|x)}{\hat{I}(d|x)} \right) = \frac{1}{2\sigma^2} \left( -\sum_{t=1}^{T} (d_t - h_t(x))^2 + \sum_{t=1}^{T} (d_t - \hat{h}_t(x))^2 \right) \]  

Let \( u^i = \ln(I(d|x)/\hat{I}(d|x)) \), the weights can then be computed as,

\[ \bar{\omega}^i = \exp(u^i - \max(u^i, \ldots, u^n)) \]  

\[ w^i = \frac{\bar{\omega}^i}{\sum \bar{\omega}} \]
References


REFERENCES


Minerva Access is the Institutional Repository of The University of Melbourne

Author/s:
Hasnur Rabiain, Azmir

Title:
Robust cooperative positioning for VANETs based on multi sensor system and enhanced estimation algorithms

Date:
2018

Persistent Link:
http://hdl.handle.net/11343/212124

File Description:
Thesis

Terms and Conditions:
Terms and Conditions: Copyright in works deposited in Minerva Access is retained by the copyright owner. The work may not be altered without permission from the copyright owner. Readers may only download, print and save electronic copies of whole works for their own personal non-commercial use. Any use that exceeds these limits requires permission from the copyright owner. Attribution is essential when quoting or paraphrasing from these works.