



UCGE Reports
Number 20209

Department of Geomatics Engineering

**INS/GPS Integration Using Neural Networks for Land
Vehicular Navigation Applications**

(URL: <http://www.geomatics.ucalgary.ca/links/GradTheses.html>)

by

Kai-Wei Chiang

November 2004



UNIVERSITY OF CALGARY

INS/GPS Integration Using Neural Networks for Land Vehicular
Navigation Applications

by

Kai-Wei Chiang

A DISSERTATION
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF GEOMATICS ENGINEERING

CALGARY, ALBERTA, CANADA

NOVEMBER, 2004

© Kai-Wei Chiang 2004

ABSTRACT

Most of the positioning technologies for modern land vehicular navigation systems have been available for 25 years. Virtually all of the systems augment two or more of these technologies. Typical candidates for an integrated navigation system are the Global Position System (GPS) and Inertial Navigation Systems (INS). The Kalman filter has been widely adopted as an optimal estimation tool for the INS/GPS integration, however, several limitations of such multi-sensor integration methodology have been reported; such as the impact of INS short term errors, model dependency, prior knowledge dependency, sensor dependency, and linearization dependency.

To reduce the impact of short term INS sensor errors, the bandwidth of true motion dynamics were identified by spectrum analysis and the first generation denoising algorithm that used the Discrete Wavelet Transform (DWT) was applied to identify the limitations of the existing denoising algorithm. Consequently, this research proposed the cascade denoising algorithm to overcome the limitations of existing denoising algorithms. It was then evaluated using several INS/GPS integrated land vehicular systems and the results demonstrated superior performance to existing denoising algorithms in both the positioning and spectrum domains. In addition, the impact of proposed algorithms on different integrated systems was investigated extensively.

Furthermore, an alternative INS/GPS integration methodology, the conceptual intelligent navigator incorporating artificial intelligence techniques, was proposed to reduce the remaining limitations of traditional navigators that use the Kalman filter approach. The proposed conceptual intelligent navigator consisted of several different INS/GPS integration architectures that were developed using artificial neural networks to acquire the navigation knowledge. In addition, the “brain”, a navigation information database, and a window based weight updating scheme were implemented to store and accumulate navigation knowledge. The conceptual intelligent navigator was evaluated using several INS/GPS integrated land vehicular systems and the results demonstrated superior

performance to traditional navigator in the position domain. Finally, a low cost INS/GPS integrated system was considered to verify the advantages gained by incorporating the conceptual intelligent navigator as an alternative method toward developing next generation land vehicular navigation systems.

ACKNOWLEDGEMENTS

I would like to express my sincerest gratitude to my supervisor Prof. Naser El-Sheimy for his continuous support, encouragement, vision, guidance, advice, trust, invaluable contributions, proposed ideas and constructive suggestions during my graduate studies. Throughout the last three and half years he has been a generous supervisor and a good friend. Any academic accomplishments I have made during this period are as much as a reflection of his efforts to motivate me as they are of anything else.

Also, I would like to express my deepest appreciation to Dr. Aboelmaged Noureldin for his last long cooperation, fruitful discussions and creative ideas. Dr. King-Chong Lo and Dr. Jaan-Rong Tsay are thanked for encouraging and supporting my idea of overseas study four years ago.

Many thanks go to all my friends and colleagues at The University of Calgary, especially members of the Mobile Multi-sensor Research Group and my officemates at F319A: Dr. Xiaoji Niu, Dr. Sameh Nassar, Eun-Hwan Shin, Walid Abd El-Hamid, Haiying Hu and Chris Goodall. My work and discussions with them were always a pleasure that created an ideal working environment for my research. Among them, Eun-Hwan is specially acknowledged for his kindness to provide the INS toolbox which was applied in certain part of this research to acquire the navigation solutions from the INS mechanization or extended Kalman filter.

This research was founded in part by the Natural Sciences and Engineering Research council of Canada (NSERC), and The Geomatics for Informed Decisions (GEOIDE) Canadian Centres of Excellences Project grants of my supervisor, The Department of Geomatics Engineering Graduate Research Scholarships, Special awards and Travel Grants, The University of Calgary Travel Grants and The Innovation in Mobile Mapping Award. Non-finical support was provided through the US Institute of Navigation Student

Paper Award and the International Chinese Professionals in Global Positioning Systems Student Paper Award.

Special thanks go to the most important friend of my life, Hsiu-Wei, for her everlasting love, patience, understanding, cooperation, encouragement, sacrifice, pleasant and calm living and working atmosphere, which have contributed a lot to the successful achievement of this thesis.

Finally, the deepest gratitude of all goes to my family members. I owe all kinds of obligation to my lovely parents and my brother for their unlimited and unconditional love, inspiration, sacrifice, guidance, care, encouragement and financial support. This research, and indeed, my entire education, is the results of a thirst for knowledge that was inspired and encouraged by them.

DEDICATION

To

My parents, my brother and Hsiu-Wei

“Without your support and sacrifice, I could not have done this work”

TABLE OF CONTENTS

	Page
APPROVAL PAGE.....	ii
ABSTRACT.....	iii
ACKNOWLEDGEMENTS.....	v
DEDICATION.....	vii
TABLE OF CONTENTS.....	viii
LIST OF TABLES.....	xiii
LIST OF FIGURES.....	xvi
NOTATION.....	xxi
Convention.....	xxi
Acronyms.....	xxii
Symbols.....	xxv
CHAPTER 1: INTRODUCTION.....	1
1.1 Background.....	2
1.2 Research Objectives and Contributions	6
1.3 Thesis Outline	9
CHAPTER 2: LAND VEHICULAR NAVIGATION SYSTEMS.....	12
2.1 Coordinate Frames.....	12

2.2 Historical perspective of Land Vehicular Positioning Technologies	16
2.3 The Role of Land Vehicular Navigation System.....	21
2.4 Positioning Technologies for Land Vehicular Navigation Systems.....	23
2.4.1 Modern Positioning Technologies.....	23
2.4.2 Multi-sensors Augmented Positioning Technologies.....	26
 CHAPTER 3: FUNDAMENTALS OF INS/GPS INTEGRATION	32
3.1 Fundamentals of GPS	34
3.1.1 GPS Observables and Positioning Principles.....	35
3.1.2 Future Development of GNSS Positioning Technologies.....	41
3.2 Fundamentals of INS	43
3.2.1 Physical Implementations of IMU	45
3.2.2 Initialization of Strapdown INS	47
3.2.3 INS Mechanization Equations	49
3.2.4 INS Error Equations	57
3.2.5 INS Error Characteristics	62
3.2.6 Future Development of INS Based Positioning Technology-MEMS IMU.....	65
3.3 INS/GPS Integration and Kalman Filter	66
3.3.1 Fundamentals of Kalman Filter	66
3.3.2 Limitations of INS/GPS Integration Using Kalman Filter	70
 CHAPTER 4: CASCADE DENOISING OF IMU SIGNALS.....	77
4.1 Continue Wavelet Transform	78
4.1.1 Spectrum Perspective of CWT	80
4.1.2 The Linkage between FFT and CWT	83
4.2 Discrete Wavelet Transform	85
4.2.1 Spectrum Perspective of DWT	87
4.3 Spectrum Analysis of IMU Raw Measurements.....	89
4.4 Existing Denoising Algorithms.....	93
4.4.1 First Generation Denoising Algorithm	93

4.4.2 Limitations of First Generation Denoising Algorithm	97
4.4.3 Second Generation Denoising Algorithm	99
4.4.4 Spectrum Perspective of Existing Denoising Algorithms.....	101
4.5 Cascade Denoising Algorithms.....	106
4.6 Performance Analysis of Cascade Denoising Algorithm.....	110
 CHAPTER 5: ARTIFICIAL NEURAL NETWORKS METHODOLOGY.....	114
5.1 Fundamentals of Artificial Neural Networks	116
5.1.1 Artificial Neurons and Neural Networks	117
5.1.2 ANNs Architecture	121
5.1.3 Learning Procedures	125
5.2 Multi-Layered Feed-Forward Neural Networks.....	126
5.2.1 Nonlinear Mapping and MFNNs	127
5.2.2 Standard Backpropagation Learning Algorithm	132
5.3 Recurrent Neural Networks	144
5.3.1 Elman Networks.....	146
5.3.2 Standard Backpropagation Learning algorithm.	147
5.4 Performance Analysis of a MFNN and SRN	150
 CHAPTER 6: DEVELOPMENT OF THE CONCEPTUAL INTELLIGENT NAVIGATOR.....	156
6.1 ANNs Based INS/GPS Integration Architecture	159
6.1.1 Position Update Architecture	159
6.1.2 Position and Velocity Update Architecture	161
6.1.3 Position, Velocity, and Azimuth Update Architecture	163
6.2 Navigation Information Database	169
6.3 Window Based Weights Updating Strategy.....	172
6.3.1 Limitations of Traditional Weights Updating Methods.....	173
6.3.2 Development of Window Based Weights Updating Strategy.	174
6.4 Performance Analysis of the Conceptual Intelligent Navigator.....	179

CHAPTER 7: RESULTS AND DISCUSSIONS.....	185
7.1 Introduction	185
7.2 Performance Analysis of the Cascade Denoising Algorithm	188
7.2.1 Performance Summary of Different Integrated Systems	189
7.2.2 Performance Analysis Index (PAI).....	193
7.3 Performance Analysis of the Conceptual Intelligent Navigator.....	197
7.3.1 CIMU/GPS integrated system.....	197
7.3.2 LN200/GPS integrated system.....	203
7.3.3 XBOW/GPS integrated system.....	207
7.4 Performance Analysis of a Low Cost MEMS/GPS Integrated System.....	211
7.4.1 Frequent GPS Outages Test.....	213
7.5 Summary.....	221
 CHAPTER 8: CONCLUSIONS AND RECOMMENDATIONS.....	225
8.1 Conclusions	225
8.2 Recommendations	229
 REFERENCES.....	232
 APPENDIX A:	245
A.1 Examples of Associated Frequency	245
A.2 Multiresolution Analysis	246
A.3 Performance Analysis of Different Thresholding Algorithm.....	252
A.4 Performance Analysis of TIW Lowpass Filter Denoising	253
A.5 Performance Analysis of Cascade Denoising Algorithm (LN200)	254
A.6 Performance Analysis of Cascade Denoising Algorithm (XBOW).....	257
 APPENDIX B:	260
B.1 Biological Neurons and Neural Networks	260
B.2 Second Order Optimization Learning Algorithms	261
B.3 Linearized Recursive Estimation Learning Algorithm.....	270

B.4 Performance Analysis of Conceptual Intelligent Navigator (LN200).....	273
B.5 Performance Analysis of Conceptual Intelligent Navigator (XBOW).....	274
 APPENDIX C:	276
C.1 Specifications of Navigation Grade IMUs	276
C.2 Specifications of Tactical Grade IMUs	278
C.3 Specifications of MEMS IMUs	279

LISTS OF TABLES

1.1	Benefits of Land vehicular navigation system	2
1.2	The steps toward developing cascade denoising algorithm.....	7
1.3	The investigation for proper ANNs' architecture.....	7
1.4	The core functional schemes of conceptual intelligent navigator.....	8
1.5	The major contributions of this research work.....	9
2.1	Land vehicular navigation systems user requirements in urban areas.....	23
2.2	The requirements of an integrated navigation system.....	27
3.1	Comparison between INS,GPS and INS/GPS	33
3.2	A Summery of GPS errors source.....	40
3.3	GPS positioning capability in autonomous mode.....	41
3.4	The GPS-Galileo GNSS characteristics.....	43
3.5	Comparison between gimballed and strapdown Implementation.....	47
3.6	Constant coefficients for normal gravity.....	52
3.7	Error sources and parameters of nav-grade IMU.....	64
3.8	A brief comparison of commonly used INS/GPS integration architectures.....	71
4.1	Associated coefficients of Daubechies wavelet family.....	85
4.2	Relationship between stop bands and decomposition level.....	89
4.3	The impact of different factors.....	92
4.4	Bandwidth of true motion dynamic.....	92
4.5	Optimal decomposition level for kinematic IMU measurements.....	95
4.6	Performance of DWT denoising.....	112
4.7	Performance of cascade TIW denoising.....	113
5.1	Comparison between KF and NN for INS/GPS data fusion and navigation.....	115
5.2	The function scheme of a neuron.....	118
5.3	Mathematical operations of an artificial neuron.....	118
5.4	Function scheme of each layer.....	122
5.5	Types of connection.....	124
5.6	Performance summary.....	144

5.7	RMS errors.....	154
5.8	Training time required for 500 seconds data.....	155
6.1	Performance Summary (RMSE).....	167
6.2	RMSE (2 nd field test).....	182
6.3	RMSE (2 nd field test).....	183
7.1	The Summary of different field tests.....	186
7.2	Background information of simulated GPS outages.....	189
7.3	Performance summary of positional errors (CIMU/DGPS)	189
7.4	Performance summary of positional errors (CIMU/SPP).....	191
7.5	Performance summary of positional errors (LN200/GPS).....	193
7.6	Performance summary of positional errors (XBOW/GPS).....	193
7.7	PAIs (CIMU/GPS).....	194
7.8	PAIs (LN200/GPS).....	195
7.9	PAIs (XBOW/GPS).....	195
7.10	RMSE of different azimuth measurements (CIMU/DGPS).....	198
7.11	Performance summary of VUA and AUA (test trajectory, CIMU/DGPS).....	200
7.12	Performance summary of different navigators (test trajectory, CIMU/GPS)....	201
7.13	Performance summary of different navigator (test trajectory, LN200/GPS)...	204
7.14	Performance summary of different navigator (test trajectory, BOW/GPS).....	208
7.15	Background information of GPS outages.....	213
7.16	Performance summary (MST/GPS).....	214
7.17	Performance analysis index.....	219
A.1	Digested associated frequency of DB3 ($\omega=0.8$).....	245
A.2	Digested associated frequency of DB5 ($\omega=0.96$).....	246
A.3	Frequencies and decomposition level.....	251
A.4	Performance Comparison (DWT).....	252
A.5	Performance Comparison (TIW).....	253
A.6	Performance of wavelet low pass filter (TIW).....	254
A.7	Performance summary of positional errors (LN200/DGPS).....	255
A.8	Performance summary of positional errors (LN200/SPP).....	256
A.9	Performance summary of positional errors (XBOW/DGPS).....	257

A.10	Performance summary of positional errors (XBOW/SPP).....	258
B.1	Performance summary.....	270
B.2	RMSE of different azimuth measurements (LN200/DGPS).....	274
B.3	Performance summary of VUA and AUA (test trajectory, LN200/DGPS).....	274
B.4	RMSE of different azimuth measurements (XBOW/DGPS).....	275
B.5	Performance summary of VUA and AUA (test trajectory, XBOW/DGPS).....	275
C.1	The category of IMUs.....	276

LISTS OF FIGURES

1.1	Current and Future POS/NAV Market Shares by Applications	3
2.1	Inertial frame (i-frame).....	13
2.2	Earth-Fixed frame (e-frame).....	14
2.3	Local Level frame (l-frame).....	15
2.4	Body frame (b-frame).....	16
2.5	Ancient odometers (computer graphic).....	17
2.6	Ancient differential odometers: (computer graphic).....	18
2.7	Ancient magnetic compass.....	19
2.8	Autonomous land vehicular navigation system.....	22
2.9	Performance comparisons of various augmented positioning technologies.....	27
2.10	Core components of a modern land vehicular navigation system.....	31
3.1	Gimbaled platform and local level frame implementation	45
3.2	Strapdown implementation	46
3.3	Local level frame mechanization	57
3.4	Simulation results of a navigation grade IMU	65
3.5	Conceptual plot of the spectrum of INS sensor errors	73
3.6	Conceptual plot of the spectrum of INS sensor errors (After INS/DGPS).....	73
3.7	The impact of X-Gyro noise	74
4.1	Comparisons between FT, STFT and CWT.....	81
4.2	Wavelets with different scales	82
4.3	Spectrum of CWT	83
4.4	Examples of the center frequencies of Daubechies wavelet family	85
4.5	Three levels wavelet decomposition and their conceptual spectrums	87
4.6	Three levels wavelet reconstruction	87
4.7	Spectrum of DWT (Approximation/Detail signals).....	88
4.8	Spectrum of CIMU.....	90
4.9	Spectrum of LN200.....	91
4.10	Spectrum of Crossbow AHRS400-CC.....	91

4.11	IMU signals denoising.....	95
4.12	Comparison between thresholding algorithms.....	96
4.13	Example of non-translation invariance.....	98
4.14	Examples of Pseudo Gibbs Phenomena.....	99
4.15	Gibbs Phenomena.....	99
4.16	UDWT decomposition.....	100
4.17	UDWT reconstruction.....	100
4.18	Simulated signals.....	102
4.19	Spectrum of DWT denoising.....	103
4.20	Spectrum of TIW denoising.....	104
4.21	Conceptual plot of the spectrum of INS sensor errors (INS/DGPS+ perfect denoising).....	104
4.22	Conceptual plot of the spectrum of INS sensor errors (INS/DGPS+ lowpass filter).....	105
4.23	Conceptual plot of the spectrum of INS sensor errors (INS/DGPS+ existing denoising).....	106
4.24	Cascade denoising algorithm.....	107
4.25	Spectrum of cascade DWT denoising.....	107
4.26	Spectrum of Cascade TIW denoising.....	108
4.27	Spectrum of CIMU X-Gyro after DWT denoising (>stop band).....	108
4.28	Spectrum of CIMU X-Gyro after cascade TIW denoising.....	109
4.29	Conceptual plot of the spectrum of INS sensor errors (INS/DGPS+ cascade denoising).....	110
4.30	The reference trajectory and simulated GPS blockage.....	111
4.31	Position errors during GPS blockages.....	112
4.32	Position errors during GPS blockages.....	113
5.1	A simplified neuron model and an artificial neuron model	117
5.2	Shapes of activation functions	121
5.3	Layer organization	122
5.4	A single layered and a multi-layered feed forward architecture.....	123
5.5	Recurrent network architectures	124

5.6	Learning procedures	126
5.7	A two layered feed-forward network	129
5.8	Nonlinear mapping	132
5.9	Learning flow of standard backpropagation algorithm.....	138
5.10	The impact of momentum learning.....	140
5.11	Regression problem solved by MFNN with different learning algorithms.....	143
5.12	Auto-associator and Hopfield network.....	145
5.13	Jordan and Elman network.....	146
5.14	Elman network.....	147
5.15	Topology of SPUA_MFNN.....	150
5.16	Topology of SPUA_SRN.....	151
5.17	Learning strategy of SPUAs.....	151
5.18	Trajectory of the first field test.....	152
5.19	Trajectory of the second field test.....	152
5.20	Position errors for the SPUA and KF during different GPS outages.....	154
5.21	Position errors for the SPUA and KF during different GPS outages.....	155
6.1	Comparison between conventional and conceptual intelligent navigator.....	157
6.2	Core components of the conceptual intelligent navigator.....	159
6.3	Topology of PUA.....	160
6.4	System configuration and learning strategy of PUA	161
6.5	Topologies of PVUA	162
6.6	System configuration and learning strategy of PVUA.....	163
6.7	Instability of $\phi_{DGPS-V}(t)$	165
6.8	Azimuth constrain algorithm.....	166
6.9	Comparison between different azimuth outputs.....	166
6.10	Topologies of PVAUA.....	167
6.11	System configuration and learning strategy of PAVUA.....	168
6.12	NAVi of PUA.....	170
6.13	Distributed navigation knowledge storage.....	170
6.14	Locate redundant navigation knowledge.....	172
6.15	Window based weights updating strategy.....	176

6.16	One step training procedure.....	177
6.17	Two steps training procedure.....	178
6.18	Trajectory of 1 st field test.....	181
6.19	Trajectory of 2 nd field test.....	181
6.20	Position Error of 2 nd field test.....	182
6.21	Window-based weights updating strategy.....	183
6.22	Training time for each window (PUA).....	184
6.23	Training time for each window (PVUA).....	184
7.1	The picture of test van	185
7.2	The trajectories of different field tests	187
7.3	Simulated GPS outages	188
7.4	Position errors (CIMU/DGPS).....	190
7.5	Position errors (CIMU/SPP).....	191
7.6	The impact of denoising on the integrated systems (CIMU/GPS)	192
7.7	The impact of cascade denoising algorithm	196
7.8	The quantized impact of cascade denoising algorithm	196
7.9	Performance of CP, VUA and AUA (CIMU/DGPS)	198
7.10	Position errors (CIMU/GPS).....	200
7.11	Trajectories generated by different navigators (CIMU/GPS).....	203
7.12	Position error (LN200/GPS).....	205
7.13	Trajectories generated by different navigators (LN200/GPS).....	206
7.14	Position errors (XBOW/GPS)	208
7.15	Trajectories generated by different navigators (XBOW/GPS).....	210
7.16	System setup and field test trajectories	212
7.17	Position errors (MST/GPS).....	215
7.18	Trajectories generated by different navigator (MST/GPS).....	216
7.19	Performance analysis index(Outage No.1 and No.2).....	220
7.20	Performance analysis index(Outage No.5 and No.6).....	220
A.1	Examples of wavelet and scaling functions.....	248
A.2	Schemes of FWT.....	249
A.3	Distinctions between (A/D) and (cA/cD).....	250

A.4	Example of the distinction between (A/D) and a (cA/cD).....	251
A.5	DWT Denoising Examples	253
A.6	TIW Denoising Examples	253
A.7	Position errors during GPS blockages	254
A.8	Position errors (LN200/DGPS).....	255
A.9	Position errors (LN200/SPP).....	256
A.10	The impact of denoising on the integrated systems (LN200/GPS).....	257
A.11	Position errors (XBOW/DGPS).....	258
A.12	Position errors (XBOW/SPP).....	259
A.13	The impact of denoising on the integrated systems (XBOW/GPS).....	259
B.1	Three stages of biological neural networks.....	260
B.2	A biological neuron.....	261
B.3	Shape and contour plot of Rosenbrock's function	269
B.4	Search path of the minimum.....	269
B.5	Performance of CP, VUA and AUA (LN200/DGPS).....	274
B.6	Performance of CP, VUA and AUA (XBOW/DGPS).....	275
C.1	The specifications of CIMU.....	276
C.2	The specifications of iNAV-RQH.....	277
C.3	The specifications of LN 90-100.....	277
C.4	The specifications of HG-1700.....	278
C.5	The specifications of LN-200.....	278
C.6	The specifications of MST	279
C.7	The specifications of IMU400CC-100.....	279
C.8	The specifications of ISI IMU	280

NOTATION

1. Conventions

- 1.1 Vectors are represented by lower-case letters
- 1.2 Matrices are represented by upper-case letters
- 1.3 Functions are represented by either upper-case or lower-case letters.
- 1.4 A dot above a vector; a matrix or a quantity indicates a time differentiation.
- 1.5 A “vector” is always considered as three-dimensional. A superscript indicates the particular coordinate frame in which the vector is represented. For example: $\mathbf{a}^i = (a_x^i, a_y^i, a_z^i)$ represents the components of the vector \mathbf{a} in the i-frame.
- 1.6 Rotation (transformation) matrices between two coordinate frames are denoted by R . The two coordinate frames are indicated by a superscript and a subscript. For example: R_i^j represents a transformation matrix from the i-frame to the j-frame.
- 1.7 Angular velocity between two coordinate frames represented in a specific coordinate frame can be expressed either by a vector $\boldsymbol{\omega}$ or by the corresponding skew-symmetric matrix Ω . A superscript and two subscripts will be used to indicate the corresponding coordinate frames. For example: $w_{ij}^k = (w_x, w_y, w_z)^T$ or

$$\Omega_{ij}^k = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ w_y & w_z & 0 \end{bmatrix}$$

describes the angular velocity between the i-frame and

the j-frame represented in the k-frame.

2. Acronyms

2-D	Two dimensional
3-D	Three dimensional
ADAS	Advanced Driver Assistance Systems
AIAs	Artificial Intelligent Algorithms
AITS	Artificial Intelligent Transport System
AIVHS	Artificial Intelligent Vehicle Highway System
ANNs	Artificial Neural Networks
A-S	Anti-Spoofing
AUA	Azimuth Update Architecture.
CDMA	Code Division Multiple Access
CS	Commercial Service
CWT	Continuous Wavelet Transform
DCM	Direct Cosine Matrix
DEKF	Decoupled Extended Kalman Filter
DGPS	Differential GPS
DL	Decomposition level
DWT	Discrete Wavelet Transform
DR	Dead Reckoning System
EKF	Extended Kalman Filter
ENU	East-North-Up
FT	Fourier Transform
GIS	Geographic Information System
GLONASS	Russian GNSS
GNSS	Global Navigation Satellite Systems
GPS	Global Positioning System
IMU	Inertial Measurements Units
INS	Inertial Navigation System
ITS	Intelligent Transport System

IVHS	Intelligent Vehicle Highway System
KF	Kalman filter
LRLSL	Linearized Recursive Least Square Learning
LS	Least Square
MCS	Master Control Station
MEMS	Micro electromechanical system
MEO	Mean Earth Orbiting
MFNNs	Multi-Layered Feed-Forward Neural Networks
MM	Map Matching
MRA	Multiresolution Analysis
NASTAR	Navigation System with Timing and Ranging
NDEKF	Neuron-decoupled EKF
NIMA	National Imagery and Mapping Agency
NNs	Neural Networks
NED	North-East-Down
OCS	Operational Control Segment
PPS	Precise Positioning Services
PRS	Public Regulated Service
PUA	Position Update Architecture
PVUA	Position and Velocity Update Architecture
PVAUA	Position, Velocity and Azimuth Update Architecture.
RHCP	Right Hand Circular Polarization
RLG	Ring Laser Gyro
RMS	Root Mean Square
RMSE	Root Mean Square Error
RNNs	Recurrent Neural Networks
RNSS	Radio Navigation Satellite Service
SA	Selective Availability
SFNNs	Single-Layer Feed-Forward Neural Networks
SNR	Signal to Noise Ratio
SRN	Simple Recurrent Network

SPP	Single Point Positioning
SPS	Sign Post Systems(Chapter 2)
SPS	Standard Positioning Service (Chapter3)
SPUA	Simple Position Update Architecture
STFT	Short Time Fourier Transform
SWT	Stationary Wavelet Transform
TIW	Translation Invariant Wavelet Transform
TRNS	Terrestrial Radio- Navigation Systems
UDWT	Undecimated Wavelet Transform
UKF	Unscented Kalman Filter
VISAT	Video-Inertial-SATellite
VUA	Velocity Update Architecture
WDEKF	Weight-decoupled EKF
ZUPT	Zero Velocity Update

3. Symbols

α	Momentum constant
b	Receiver clock bias
b_f	Accelerometer bias
b_w	Gyro drifts
c	The speed of light
γ	Normal gravity
d_{ion}	Ionospheric error
d_{trop}	Tropospheric error
dt	Satellite clock error
dT	Receiver clock error
$d\rho$	Orbital error
∇	Single difference operator
$\Delta\nabla$	Double difference operator
δ	Threshold value
$\delta^{(h)}$	Delta variable of hidden neuron
$\delta^{(o)}$	Delta variable of output neuron
$\varepsilon_{M\phi}$	Carrier phase multipath error
ε_ϕ	Receiver phase noise error
ε_{Mp}	Code range multipath error
ε_p	Receiver code noise error
F_A	Associated frequency
F_c	Center frequency
$F_{k,k-1}$	State transition matrix
F_p	Pseudo frequency
F_s	Sampling frequency

$F(w)$	Fourier transform
f	Vector of accelerometer specific force measurements
ϕ	Carrier phase measurement
G_{k-1}	System noise coefficient matrix
g	Gravity vector
η	Learning rate parameter
H_k	Design matrix
K_k	Kalman gain matrix
λ	Carrier wavelength
N	Integer cycle ambiguity
P	Pseudorange measurement
ρ	Geometric range
Q	System noise covariance matrix
R	Measurement noise covariance matrix
S_f	Scale factor of accelerometer
T^{STFT}	Short time Fourier transform
v_k	Measurement noise
$W_a^{(c)}$	Synaptic weights matrix of context layer
$W_a^{(h)}$	Synaptic weights matrix of hidden layer
$W_a^{(o)}$	Synaptic weights matrix of output layer
$\Delta w_a^{(c)}$	Synaptic weight correction of context neuron
$\Delta w_a^{(h)}$	Synaptic weight correction of hidden neuron
$\Delta w_a^{(o)}$	Synaptic weight correction of output neuron
w	Wavelet transform
w_k	System noise
ψ	Mother wavelet function
$\psi_{a,b}$	Scaled and Shifted version of mother wavelet function

CHAPTER 1

INTRODUCTION

Navigation comprises the methods and technologies to determine the time varying position and attitude of a moving object by measurement. Position, velocity and attitude, when presented as time variable functions, are called navigation states because they contain all necessary navigation information to georeference a moving object at any moment of time. In those cases where only the position of the moving object is required, the term ‘kinematic positioning’, instead of navigation, is usually used [Schwarz and El-Sheimy, 1999].

Land vehicular navigation has a surprisingly long history. Some of its principles follow the analogous precepts of animal survival mechanisms. For instance, dead reckoning is the same approach used by honey bees to report the exact position of nectar as a bearing and distance through a complicated dance pattern. Some experiments have revealed that pigeons are capable of detecting the Earth's magnetic field and can use it to orient and possibly to navigate [French, 1986]. The earliest vehicular navigation systems, the range finder chariot and Chinese south-pointing chariot, date back to at least two thousands years ago (see chapter 2 for details).

During the last two decades, the rapid growth in the use of land vehicles and the dependence on roads for a significant proportion of freight movement has led to increasingly high levels of congestion, and with all the associated problems of economic loss, environment damage and safety concern [Drane and Rizos, 1998].. In this area, an enhanced land vehicular navigation system will offer great potential for improvements. According to Zhao [1997], the benefits of land vehicular navigation systems can be categorized as Table 1.1:

In fact, Navigation systems are becoming standard in-vehicle equipments. Based on the positioning capability of the vehicle, location-based information for the driver could be provided. On the other hand, with the advanced development of computer technologies

in hardware and software, the incorporation of Artificial Intelligent Algorithms (AIAs) with the next generation navigation system is receiving more attention. AIAs include artificial neural networks (ANNs), fuzzy logic, evolutionary computing, probabilistic computing, expert systems, and genetic algorithms. The trend toward the incorporation of AIAs and navigation algorithm is fueled by the need for intelligent systems and the limitations with the current navigation algorithms, such as the extended Kalman filter, which have been reported by several researchers [Gelb, 1974; Brown and Hwang, 1992 Vanicek and Omerbasic, 1999]. As the extended Kalman filters usually serve as the multi-sensors integration scheme for the current land vehicular navigation; therefore, investigating new integration techniques using AIAs for general land vehicular navigation become the motivation of this research.

Table 1.1: Benefits of Land vehicular navigation system

Society	Commercial applications:	Single user
<ul style="list-style-type: none"> ○ Economic efficiency: It makes the best use of vehicle fleets and in minimizing journey time or distance ○ Road safety: It can significantly enhance road safety by presenting the route guidance information 	<ul style="list-style-type: none"> ○ Road and load optimization: It reduces costs and improves efficiencies. ○ Crisis alarm and response: It addresses the safety issues affecting the vehicle. ○ “Just in time” operation: It requires accurate prediction of arrivals and precise timing. ○ Schedule adherence: it is more applicable to transit vehicles, such as buses and trains. 	<ul style="list-style-type: none"> ○ En route driving information: It assists avoidance of incidents. ○ Safety readiness: It alerts driver regarding safety issues. ○ Automatic vehicle operation: It facilitates driverless operation. ○ Traveler services: It provides information on destinations for travelers en route.

1.1 Background

The last two decades have shown an increasing trend in the use of positioning and navigation (POS/NAV) technologies in land vehicle applications; primarily in land vehicular navigation systems. Figure (1.1) shows the major sections of the current POS/NAV market in terms of a dollar value. Currently, the land vehicle market section has a 32% share (about \$4 billion US) of the total market. By 2005, this will grow to over \$12 billion US reaching almost 50% of the total POS/NAV market [Schwarz and El-Sheimy, 1999].

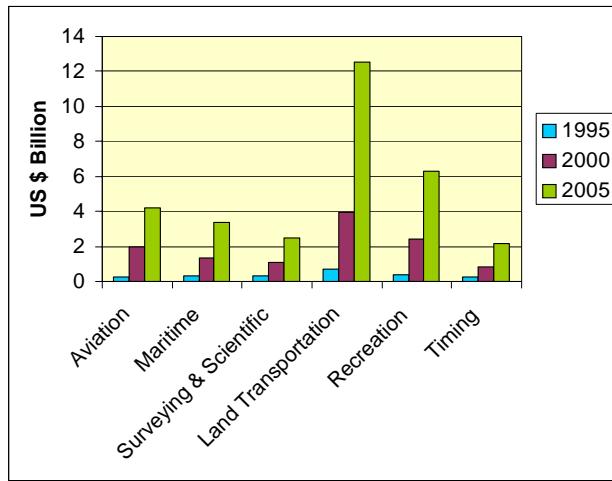


Figure 1.1: Current and Future POS/NAV Market Shares by Applications
(After Schwarz and El-Sheimy, 1999)

Application areas of POS/NAV technologies in land transportation are numerous including automated car navigation, emergency assistance, fleet management, person finding, asset tracking, collision avoidance, environment monitoring, and automotive assistance [Czerniak et al, 1998]. More important, the convergence of location, information management and communication technologies have created a rapidly emerging market known as location-based service (LBS). LBS is a critical enabling technology using location as a filter or magnet to extract relevant information to provide value-added service such as location-aware billing, automated advertising services and other location-based information sought by the user based on his or her location. Because of the importance of location information, the market in turn has pushed hard for the

development of next generation low-cost/small-size in-vehicle navigation and guidance systems, providing not only location information but also route-guidance and location-sensitive services [El-Sheimy, 2000a]

Modern land vehicular navigation systems have been thought of as a very recent development, however, most of their positioning technologies have been available for 25 years. Virtually all modern land vehicle navigation systems integrate two or more of these technologies. Typical candidates for such an integrated navigation system are the Global Position System (GPS) and Inertial Navigation Systems (INS); more details will be given in chapters 2 and 3.

GPS has become the primary positioning technology for land vehicular navigation applications. However, while an increasing number of user communities accept that GPS can easily achieve the level of performance required under ideal conditions, it also widely acknowledges that the system becomes highly unreliable in certain environments. Thus, although GPS has been recognized as an all-weather available positioning technology, the line of sight signal requirements prevent its use as an all-environment positioning technology.

On the contrary, an INS is a self-contained positioning and attitude device. In other words, it meets the all-environment requirement. The primary advantage of using an INS for land vehicle navigation applications is that velocity and position of the vehicle can be provided with abundant dynamic information and excellent short term performance. However, an INS if used as stand-alone system (i.e. without external aiding) is only accurate for a limited time as its errors grow with time. Thus, an integrated system provides an enhanced navigation system that has superior performance in comparison with either a stand-alone GPS or INS as it can overcome each of their limitations. Although the Kalman filter has been widely adopted as a standard optimal estimation tool for the integration of INS and GPS, it does have limitations. These limitations will be briefly discussed below with further details given in section 3.3.2.

According to Skaloud [1999], the benefits of INS/GPS integration are band-limited as the lower boarder of the INS/DGPS error spectrum is mainly determined by the biases in GPS observations while the upper boarder is mainly determined by short term inertial sensor errors. Since the long term errors usually include accelerometer bias, gyro drifts that are usually modeled as error states are limited with the external aiding for long period of time, thus, the remaining short term errors are responsible for a certain amount of the error accumulation during the GPS outage period.

Skaloud [1999], Burton et al., [1999], and Nassar [2004] successfully used the wavelet transform as a denoising tool to reduce the impact of INS short term errors and to improve the positioning accuracy during GPS outages for land vehicular and airborne applications. The main concern in denoising the INS data is how to remove short term errors without jeopardizing the true motion dynamic component of the vehicle. It requires the prior knowledge of the bandwidth of the true motion dynamic of typical land vehicles and the spectrum characteristic of the wavelet denoising algorithm. However, the research activity mentioned above did not provide such information. In addition, they did not suggest any proper criteria that can be applied to decide upon the optimal decomposition level required by the wavelet denoising algorithm.

On the other hand, the Kalman filter depends on a set of measurements and a proper dynamics model for the navigation parameters error states and stochastic model of sensor errors to provide optimally estimated states [Skaloud, 1999]. Thus, besides the quality of the measurements, the final quality of the filter states relies on the quality of the dynamic model. If the filter is exposed to input data that does not fit the model, it will not result in reliable estimates. Obviously, the model presentation has to depend on the initial knowledge and on the real process taking place in the system.

As mentioned in Vanicek and Omerbasic [1999], the Kalman filter is system or sensor dependent as the parameters of the Kalman filter vary from one system to another (i.e., IMUs with different accuracy level), or even between similar sensors (IMUs with similar accuracy level). In fact, it requires an expert to spend enormous amounts of time and

effort to come up with a set of optimal parameters. Such a tuning process can be regarded as the learning process of the Kalman filter. In contrast, the AIAs have been verified as successful and effective for providing solutions to certain engineering and scientific problems that could not be solved using conventional estimation techniques.

ANNs have been extensively studied with the aim of achieving human-like performance, especially in the field of pattern recognition and robot control and navigation [Mandic and Chambers, 2001]. These networks are composed of a number of nonlinear computation elements which operate in parallel and are arranged in a manner reminiscent of biological neural interconnections. ANNs are designed to mimic the human brain and duplicate its intelligence by utilizing adaptive models that can learn from the existing data and then generalize what has been learnt [Ham and Kostanic, 2001]. This research work attempts to investigate the incorporation of ANNs methodologies toward developing alternative INS/GPS integration methodology for general land vehicular navigation applications.

1.2 Research Objectives and Contributions

The main objective of this research is to develop a novel wavelet denoising algorithm, cascade denoising algorithm, and develop a conceptual intelligent navigator that consists of ANNs based INS/GPS integration architectures for next generation land vehicular navigation systems. Ultimately, the conceptual intelligent navigator is expected to overcome or, at least, reduce the limitations of the conventional Kalman filter based INS/GPS integration algorithms. As each of these limitations contributes to certain amount of positional error accumulation during GPS outage, therefore, the proposed new algorithms including cascade denoising algorithm and the conceptual intelligent navigator are expected to reduce the impact of these limitations by reducing the positional error accumulation during GPS outage.

The first objective is to develop a cascade denoising algorithm to remove short term INS sensor errors including high frequency noises, vibrations and other disturbances to

improve the positioning accuracy during GPS outages. In addition, it is implemented to overcome the limitations of existing denoising algorithms. Thus the steps toward developing the cascade denoising algorithm can be summarized in Table 1.2.

Table 1.2: The steps toward developing cascade denoising algorithm

Spectrum analysis of Continuous and Discrete Wavelet Transform (CWT and DWT)	
Task	<i>The spectrum characteristics can be applied as an index to decide the optimal decomposition level required for the denoising algorithm.</i>
Spectrum analysis of raw measurements of Inertial Measurements Units (IMU)	
Task	<i>The bandwidths of true motion dynamics that are monitored by each sensor (i.e., X-Gyro/Accel, Y-Gyro/Accel and Z-Gyro/Accel) can be applied to determine the optimal decomposition level without jeopardizing the true motion dynamic components.</i>
Spectrum analysis of existing denoising algorithm	
Task	<i>The limitations of the existing denoising algorithm in the spectrum domain can be identified by investigating its spectrum characteristics.</i>

After removing the short term errors, the next objective is to review and investigate the proper ANNs' architectures for developing the INS/GPS integration algorithm and to build the foundation toward investigating of a conceptual intelligent navigator. The investigation for the proper ANNs' architectures is given in Table 1.3.

Table 1.3: The investigation for proper ANNs' architecture

Performance analysis of different types of ANNs' architectures	
Task	<i>1. A supervised static neural network (i.e., Multi-layered feed-forward neural networks, MFNNs) and a supervised dynamic neural network (i.e., recurrent neural networks), are investigated. 2. The performance analysis of the two algorithms in terms of positioning accuracy and learning speed 3. The investigation will recommend the ANN architecture that will be the core algorithm toward the development of an ANNs based INS/GPS integration architecture.</i>
Performance analysis of different learning algorithms	
Task	<i>1. Standard backpropagation algorithm, second order learning algorithms and extended Kalman filter based learning algorithms will be compared in terms of their learning speed and prediction accuracy.</i>

After the proper ANNs' architecture and associated learning algorithms are decided, the final objective is to develop the core components of the conceptual intelligent navigator for INS/GPS integration at the software level. Three functional schemes are required for developing the conceptual intelligent navigator, as shown in Table 1.4.

Table 1.4: The core functional schemes of conceptual intelligent navigator

Acquisition of the navigation knowledge	
Task	<i>1. The navigator is expected to acquire necessary knowledge during the learning process 2. Several INS/GPS integration architectures using Multi-layer Feedforward Neural Networks (MFNNs) are developed to provide the navigation knowledge to the navigator.</i>
Storage of the navigation knowledge	
Task	<i>1. The navigator will include a navigation knowledge database for storing the acquired and learned navigation knowledge. The database will act as the "brain" of the navigator.</i>
Accumulation of the navigation knowledge	
Task	<i>1. The navigator is expected to have the ability to accumulate the navigation knowledge and store it in the "brain" to provide up-to-date navigation knowledge if required. 2. Consequently, a window based weight updating strategy will be developed to accumulate acquired navigation knowledge.</i>

Finally, several INS/GPS integrated land vehicular navigation systems are applied to evaluate the performance of the developed conceptual intelligent navigator in terms of positioning accuracy during GPS outage. The major contributions of this research work are given as Table 1.5. In addition, some of the material presented in Chapters 4, 5, and 6 has been previously published or submitted for publication in papers (see Table 1.5 for details). In those cases where the candidate has been the author or a co-author of these papers, quotations are not indicated as such, but are simply referenced.

Table 1.5: The major contributions of this research work

Development of the cascade denoising algorithm	
Contributions	<ol style="list-style-type: none"> 1. <i>The spectrum analysis of the Continuous Wavelet Transform (CWT), Discrete Wavelet Transform (DWT) and existing denoising algorithms;</i> 2. <i>The spectrum analysis of raw IMU dynamics signals and</i> 3. <i>The comparison and performance analysis between the existing denoising algorithm and the developed cascade denoising algorithm in the spectrum and position domains.</i>
Publications	<i>Chiang et al., [2004b]</i>
Development of the conceptual intelligent navigator	
Contributions	<ol style="list-style-type: none"> 1. <i>The performance analysis of a static and a dynamic neural network using an INS/GPS integrated land vehicular navigation system;</i> 2. <i>The development of several ANNs based INS/GPS integration architectures to generate navigation knowledge;</i> 3. <i>The development of the window based weights updating strategy to accumulate navigation knowledge;</i> 4. <i>The incorporation of the navigation information database as the “brain” of the navigator to store navigation knowledge.</i>
Publications	<i>Chiang and El-Sheimy [2002], Chiang et al., [2003], El-Sheimy et al., [2003], Chiang [2003], Chiang [2004], Chiang and El-Sheimy [2004a], Chiang et al., [2004a] ,Chiang and El-Sheimy [2004b] ,Chiang and El-Sheimy [2004c], El-Sheimy et al., [2004]</i>

1.3 Thesis Outline

The thesis contains eight chapters and three appendices that are organized as described below:

Chapter 1 presents the motivation, objectives, and contributions of this research work.

Chapter 2 reviews several aspects of land vehicular navigation systems, such as various coordinate frames, the historical perspective of land vehicular positioning technologies, the role of land vehicular navigation systems, and the modern positioning technologies of land vehicular navigation systems.

Chapter 3 presents the fundamentals of GPS and INS to review the advantages and the disadvantages of each system and the benefits of an INS/GPS integrated system. The chapter also includes the future development of the satellite based and INS based positioning technologies. Finally, the fundamentals of Kalman filtering and associated INS/GPS integration architectures are reviewed. Before ending, the limitations of the Kalman filter based INS/GPS integration algorithm are identified.

Chapter 4 presents the Continuous Wavelet Transform (CWT) to gain some appreciation regarding the benefits of the technique. Following that, the Discrete Wavelet Transform (DWT) and Multiresolution Analysis (MRA) are given to provide the spectrum perspective of MRA. The bandwidth of true motion dynamic signal is investigated through the spectrum perspective of kinematic IMU signals. In addition, the limitations of existing wavelet denoising algorithms are given through a spectrum analysis. Finally, the idea of the cascade denoising algorithm is implemented and its performance is analyzed by using kinematic IMU measurements.

Chapter 5 demonstrates the fundamentals of ANNs. Background information is given, followed by several aspects associated with static neural networks and dynamic neural networks, such as topologies of different neural network architectures, standard backpropagation learning algorithms, second order learning algorithms and linearized recursive estimation learning algorithms. The chapter addresses the performance evaluation of a dynamic and static neural network using real data obtained through INS/GPS integrated land vehicular systems. The conclusions are then applied to decide whether dynamic neural networks or static neural networks are to be used as the core algorithm for developing the conceptual intelligent navigator.

Chapter 6 introduces several ANNs based INS/GPS integration architectures to generate navigation knowledge for the conceptual intelligent navigator. After that, the concept of a navigation information database is discussed to provide storage space for the navigation knowledge. A window based weights updating strategy is then given to accumulate

navigation knowledge. The chapter concludes with a performance analysis of the proposed intelligent navigator.

Chapter 7 presents the detailed performance analysis of the cascade denoising algorithm and the conceptual intelligent navigator using several INS/GPS integrated land vehicular navigation systems that include navigation grade, tactical grade, and micro electromechanical system (MEMS) IMUs. In addition, several issues regarding the cascade denoising algorithm on these different IMU systems as well as the performance of the conceptual intelligent navigator on different INS/GPS integrated systems (i.e., different IMU systems combined with Differential GPS (DGPS) mode or Single Point Positioning (SPP) mode) is investigated.

Chapter 8 draws the major conclusions from the research work and provides recommendations for future investigation.

CHAPTER 2

LAND VEHICULAR NAVIGATION SYSTEMS

Land vehicular navigation has a surprisingly long history. In fact, most of the basic principles of modern land vehicular navigation date back approximately 2000 years [French, 1986]. The earliest land vehicular navigation systems are thought to be the Chinese range finder chariot (ancient odometer); a distance measuring device and Chinese south-pointing chariot which acted as an automatic direction-keeping system. With the recent development of computer technologies during the past two decades, the evolution of positioning technologies of land vehicular navigation systems have been accelerated with the fastest speed ever in human history. As a result, the role of land vehicular navigation systems has become more significant and critical in several aspects of daily life, such as economic efficiency and road safety issues.

In this chapter, several aspects of land vehicular navigation systems are reviewed. This includes the various coordinate frames that are usually used, the historical perspective, the role of these systems and the modern positioning technologies adopted.

2.1 Coordinate Frames

The navigation result is expressed relative to a known reference, which is usually defined by a specific coordinate system. The measurements of different navigation sensors are also resolved relative to a particular coordinate system. When the sensor coordinate system and navigation frame of reference do not match, the sensor measurement must be transformed from the sensor coordinate system to the navigation frame. According to El-Sheimy [2004a], several different reference frames that are applied for navigation systems are:

- Inertial Frame (i-frame)

The origin of the i-frame coincides with the Earth's centre of mass. The axes are non-rotating with respect to the fixed stars, its z-axis parallel to the spin axis of the Earth, its

x-axis pointing towards the mean vernal equinox, and its y-axis completing a right-handed orthogonal frame, as shown in Figure (2.1). The vernal equinox is the ascending node between the celestial equator and the ecliptic. Thus, the right ascension system is used as the inertial frame in practice, since it closely approximates an inertial frame.

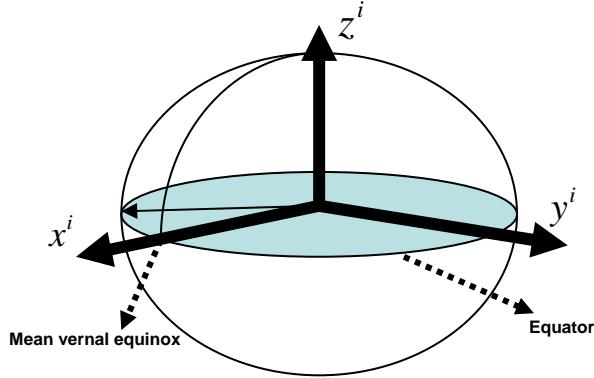


Figure 2.1: Inertial frame (i-frame)

■ Earth-Fixed Frame (e-frame)

The origin of the e-frame coincides with the Earth's centre of mass and the axes are fixed with respect to the Earth. Its x-axis points towards the mean meridian of Greenwich, its z-axis is parallel to the mean spin axis of the Earth, and its y-axis completes the right-handed orthogonal frame, as shown in the Figure (2.2). In this definition, the i-frame and e-frame differ by a constant angular rotation which equals the mean rotation rate of the Earth, $w_{ie}^e = (0, 0, w^e)$, about the common z-axis. The small variations in orientation and the speed between the actual rotation axis of the earth and the mean rotation axis can be neglected for practical purposes. The relationship going from the e-frame to the i-frame is:

$$C_e^i = \begin{bmatrix} \cos w_e t & -\sin w_e t & 0 \\ \sin w_e t & \cos w_e t & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

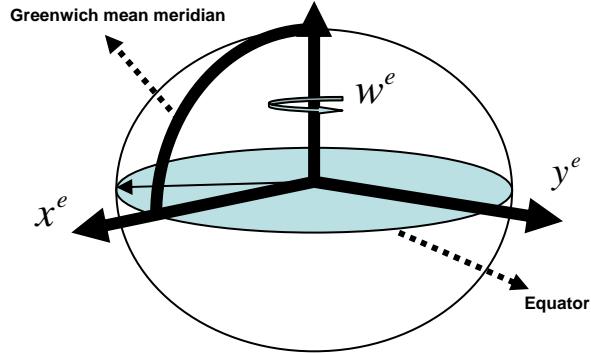


Figure 2.2: Earth-Fixed frame (e-frame)

■ Navigation Frame(n-frame)

The navigation frame is usually defined as the local level frame (l-frame) which is a local geodetic frame whose origin coincides with the origin of the sensor frame. Its x-axis points towards geodetic north, its z-axis is orthogonal to the reference ellipsoid pointing down, and its y-axis completes a right-handed orthogonal frame. This is the north-east-down (NED) system which is shown in Figure (2.3). The rotation between the e-frame and the l-frame is described by the following Direct Cosine Matrix (DCM):

$$C_e^l = R_y(-\varphi - \frac{\pi}{2})R_z(\lambda) = \begin{bmatrix} -\sin \varphi \cos \lambda & -\sin \varphi \sin \lambda & \cos \varphi \\ -\sin \lambda & \cos \lambda & 0 \\ -\cos \varphi \cos \lambda & -\cos \varphi \sin \lambda & -\sin \varphi \end{bmatrix} \quad (2.2)$$

Where λ is the longitude, φ is the latitude, and R_y, R_z describe the rotation of the coordinate systems about the y and z-axis respectively. Using the orthogonality characteristics of the DCM, the DCM from the l-frame (NED) to the e-frame is obtained as follows:

$$C_l^e = (C_e^l)^{-1} = (C_e^l)^T = \begin{bmatrix} -\sin \varphi \cos \lambda & -\sin \lambda & -\cos \varphi \cos \lambda \\ -\sin \varphi \sin \lambda & \cos \lambda & -\cos \varphi \sin \lambda \\ \cos \varphi & 0 & -\sin \varphi \end{bmatrix} \quad (2.3)$$

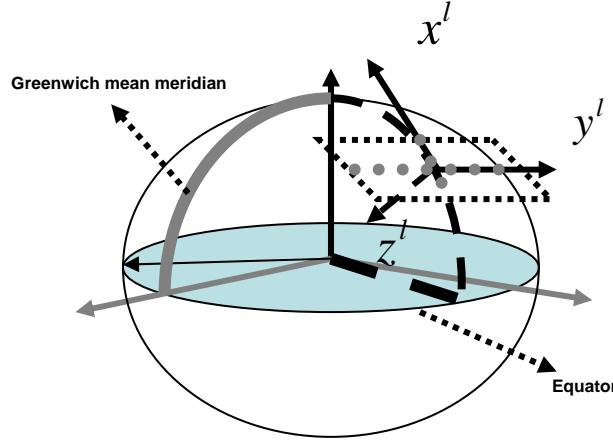


Figure 2.3: Local Level frame (l-frame)

■ Body Frame (b-frame)

The body frame is an orthogonal frame attached to the vehicle. Its axes coincide with the input axes of the sensor block; thus, the raw outputs of the IMU are the components of the rotation rate and the acceleration experienced by the sensor block along the body axes. This forward-across-down body frame is shown in the Figure (2.4). The DCM between the l-frame and b-frame is:

$$C_l^b = R_x(\phi)R_y(\theta)R_z(\psi) \quad (2.4)$$

Where $\phi, \theta, \text{and } \psi$ are the three components of the Euler rotation angles roll, pitch, and azimuth, respectively, between the l-frame and the b-frame. Similarly, the DCM from the b-frame to the l-frame can be obtained via the orthogonality criteria of DCM:

$$\begin{aligned}
C_b^l &= (C_l^b)^{-1} = (C_l^b)^T = R_z(-\psi)R_y(-\theta)R_z(-\phi) \\
&= \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \quad (2.5) \\
&= \begin{bmatrix} \cos\theta\cos\psi & -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi \\ \cos\theta\sin\psi & \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix}
\end{aligned}$$

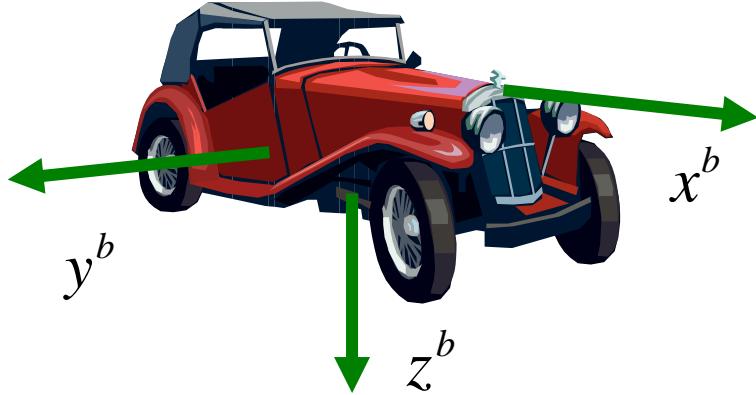


Figure 2.4: Body frame (b-frame)

2.2 Historical perspective of Land Vehicular Positioning Technologies

Before reviewing the modern positioning technologies, which are given in section 2.4, the historical perspective about the evolution of those positioning technologies is given as follows:

- Odometer

In general, almost all land vehicular navigation systems include an odometer as part of a dead-reckoning system [French, 1987]. The odometer is a device that measures distance traveled and its name comes from the Greek word hodos (way) and metron (measure). The first odometer was used in China during the Later Han Dynasty (25-220 AD). Chang Heng, the inventor of the known seismograph, also invented an odometer, also known as

the range finder chariot, as shown in the Figure (2.5). A set of gears connected the wheels of the carriage with the wooden drummer, on the same principle as that of the modern automobile odometer. The gear mechanism was driven by the wheels of a cart operated by two automatons. One struck a drum as each li (0.5 km) went by to measure distance, and the second rang a bell at the end of every 10 li (5km) [French, 1986].

Modern mechanical odometers for land vehicles use similar principles to rotate numerical-faced cylinders to indicate distance traveled. However, mechanical odometers are now being replaced by electronic odometers which use a computer to add an increment of distance to the displayed value for each electronic pulse received from a sensor monitoring the rotation of a wheel. Signals from electronic odometers are used as dead reckoning inputs for all modern land vehicular navigation systems.

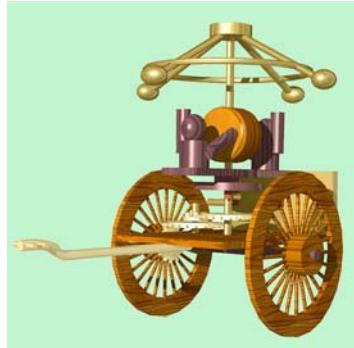


Figure 2.5: Ancient odometers (computer graphic)
(Courtesy of the National Science & Technology Museum, Taiwan]

■ Differential Odometer

The differential odometer is essentially a pair of odometers; one for a wheel on each side of the vehicle, as illustrated in Figure (2.6). As the vehicle turns, the outer wheel travels farther than the inner wheel by a distance that is equal to the product of the change in heading and the width of the vehicle. Thus continuous comparison of the difference in travel by the two wheels indicates the occurrence and degree of turns.

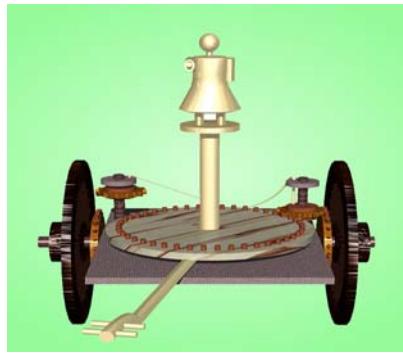


Figure 2.6 Ancient differential odometres: (computer graphic)
(Courtesy of the Ancient Chinese Machinery Research Center, Taiwan)

According to a legend, the chariot was invented by the first emperor, Huang Di, in ancient China around 2600 B.C. However, the differential odometer was invented in China about 2000 years ago according to true historic literatures. It was the technological basis for the “south-pointing chariot”. An ancient Chinese mechanical engineer, Ma Chuin, was credited for the invention of the south pointing chariot. When changing heading, a gear train driven by the chariot’s outer wheel engaged and rotated a horizontal turntable to exactly offset the change in heading. Thus, a figure with an outstretched arm mounted on the table always pointed in its original direction regardless of which way the chariot turned, as indicated in Figure (2.6) [French, 1986].

■ Magnetic Compass

A compass is an instrument containing a freely suspended magnetic element which displays the direction of the horizontal component of the Earth's magnetic field at the point of observation. The magnetic compass is an old Chinese invention, probably first made in China during the Qin dynasty (221-206 B.C.). The Chinese designed the compass on a square slab which had markings for the cardinal points and the constellations. The pointing needle was a lodestone spoon-shaped device, with a handle that would always point south, as shown in Figure (2-7). Magnetized needles used as direction pointers instead of the spoon-shaped lodestones appeared in the 8th century AD, again in China, and between 850 and 1050 they seem to have become common

navigational devices on ships. The first person recorded to have used the compass as a navigational aid was Zheng He (1371-1435), who made seven ocean voyages between 1405 and 1433 [Zhao,1997].



Figure 2.7: Ancient magnetic compass
(Courtesy of the National Science & Technology Museum, Taiwan)

■ Gyroscope

In the mid-19th century, the spinning top acquired the name, "gyroscope," though it was not invented as a navigation tool at that time. The French scientist Leon Foucault had experimented with a long, heavy pendulum in an attempt to observe the rotation of the Earth. The pendulum was set swinging back and forth along the north-south plane, while the Earth turned beneath it.

Foucault corroborated the observation by using a spinning top in a similar manner. He placed a wheel, rotating at high-speed, in a supporting ring in such a way that the axis of the spinning wheel could move independently of the ring. In fact, the supporting ring moved over the course of a day, as it was connected to the surface of the rotating Earth. The axis of the wheel remained pointing in its original direction, confirming that the Earth was rotating in a twenty-four hour period. Foucault named his spinning wheel a "gyroscope," from the Greek words "gyros" (revolution) and "skopein" (to see) [Titterton, and Weston, 1997].

■ Early Electronic Land Vehicular Navigation System

One of the earliest land vehicular navigation systems to incorporate electronic components was the vehicular odograph developed for jeeps and other military vehicles during World War II by the U.S Army Corps of Engineers [French, 1986]. The main components included a magnetic compass whose needle position was read by a photocell. The compass output drove a servomechanism to rotate a mechanical shaft corresponding to vehicle heading. The compass shaft was coupled to a mechanical computer in a plotting unit that resolved travel distance from an odometer shaft into x, y components, and drove a stylus to plot the vehicle's course automatically on a map of corresponding scale.

■ Radio-based Navigation System

The first radio-based navigation technique amounted to determining the direction of a known transmitter by rotating a direction-sensitive antenna. Much higher precision was offered by a series of systems known as OMEGA, DECCA, and Long Range Navigation (LORAN-C), see Kayton and Fried [1996] for details. These were developed around the time of World War II. By timing the difference in arrivals of radio signals from a 'master' and a 'slave' transmitter (which re-transmitted the master signal the moment it received it), a ship could locate itself along a specific curve (in the 2-D plane case, a hyperbola) [French, 1995].

The Global Positioning System (GPS) is the only system that is able to provide the user's position on the Earth anytime, in any weather, anywhere at present time. 28 (24+4spares) GPS satellites orbit the Earth at an altitude of 20,000 km. They are continuously monitored by numerous worldwide ground stations. The satellites transmit signals that can be detected by anyone with a GPS receiver. Section 3.1 contains more detailed information about the fundamentals of GPS.

■ Micro-Electro-Mechanical Systems Inertial Measurement Units

Inertial Navigation Systems (INS) have become an important component of aircraft, ship and submarine navigation and guidance since World War II. Although the latest

development of INS and computer technologies have reduced the size and improved the accuracy and stability of INS, the high cost and government regulations limit the wider inclusion of high quality IMUs as part of commercialized land vehicular navigation systems. The recent development of Micro-Electro-Mechanical Systems (MEMS) technology have shown promising light towards the development of IMUs. MEMS are integrated micro devices or systems combining electrical and mechanical components whose size ranges from micrometres to millimetres. MEMS, an enabling technology combined with the miniaturization of electronics, have made it possible to produce chip-based inertial sensors for use in measuring angular velocity and acceleration. These chips are small, lightweight, consume very little power, and are extremely reliable. They have therefore found a wide spectrum of applications in the automotive and other industrial applications [Schwarz and El-Sheimy, 1999]. The detailed content of the INS fundamentals will be described in section 3.2.

2.3 The Role of Land Vehicular Navigation Systems

According to Krakiwsky [1993] and Zhao [1997], the roles of land vehicular navigation systems can be grouped into four types: autonomous, fleet management, advisory and inventory. Among these systems, the autonomous systems are usually implemented as in-vehicle navigation system, therefore, the configuration of such system is given in brief below; see Krakiwsky [1993] and Zhao [1997] for more details about the system configuration of fleet management, advisory and inventory systems.

Autonomous systems usually operate in stand-alone vehicles, require continuous route guidance and navigation information and include on-board navigation devices. Figure (2.8) illustrates an example of an on-board navigation computer, a GPS receiver with a communication link for positioning correction from nearby base station (i.e., Differential GPS, DGPS) and auxiliary sensors such as Inertial Measurement Units (IMU). These systems do not require a dispatch or control center. Autonomous systems may or may not include communication links if the DGPS mode is replaced by the kinematic single point positioning (SPP) mode, depending upon the positioning accuracy required. In fact, with the GPS modernization plan, the positioning accuracy of SPP can be improved to 1~5

metres level [Chiang, 2003]. Meanwhile, the latest development of MEMS IMU makes itself an appropriate candidate as part of an autonomous system in the near future. As a result, SPP and MEMS IMUs can be used to develop for next generation land vehicular navigation systems that are inexpensive, small, and consume low power.

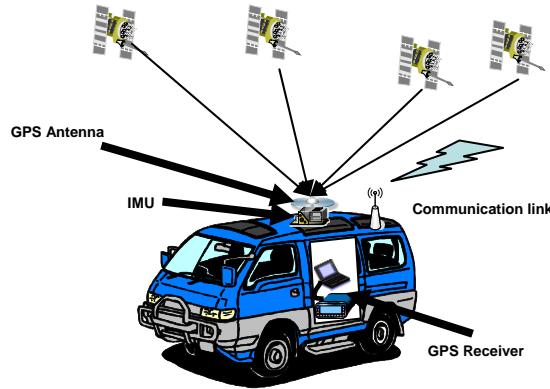


Figure 2.8: Autonomous land vehicular navigation system

The land vehicular navigation systems have different requirements in terms of positioning accuracy and continuity. A fundamental factor affecting the positioning accuracy is the environment that the system is operating in, for example, rural or urban areas. French [1990] reports the diversity of the applications and the different accuracies required for different user requirements. Table 2.1 illustrates the examples of land vehicular navigation user requirements in urban areas, see Krakiwsky [1994] and Hofmann-Wellenhof et al.,[2003] for details.

According to Hofmann-Wellenhof et al.,[2003], Autonomous systems in urban areas require continuous positioning information accurate to within 2~5 metres (2D-95%) in order to always know exactly which street and street lane is being traveled and which intersection is being approached. In rural areas, the required positioning accuracy for the same type of system deteriorates to within 5~10 metres as it is adequate to simply know the highway that is being traveled on.

Table 2.1: Land vehicular navigation systems user requirements in urban areas
 (After Krakiwsky, 1994 and Hofmann-Wellenhof et al., 2003)

Application	Vehicle Type	Accuracy(m) 2D-95%	Purpose and benefits
Autonomous	<i>Car</i>	2-5	<i>Continuous positioning</i>
Taxi	<i>Taxi Car</i>	10-200	<i>Efficient dispatching</i>
Urban Transit	<i>Bus</i>	20-50	<i>Maintain schedule</i>
Inter-city buses	<i>Bus</i>	50-200	<i>Scheduling user connection</i>
Ambulances	<i>Ambulance</i>	10-20	<i>Fast response to emergencies</i>
Police	<i>Car</i>	10-20	<i>Improved fleet management, response time, safety</i>
Utilities	<i>Truck</i>	20-50	<i>Fast response to repairs</i>

2.4 Positioning Technologies for Land Vehicular Navigation Systems

2.4.1 Modern Positioning Technologies

The core component of any land vehicular navigation system is the positioning technology. Positioning technologies, which include the development of positioning sensors and navigation algorithms, have undergone a major evolution over the past few decades to meet a wide range of applications. According to El-Sheimy [2004a], the information that is required in every land vehicular navigation system on a continuous basis is:

- Position and velocity of the vehicle with respect to a reference frame,
- Distance traveled and orientation of the vehicle, comprising the vehicle azimuth (heading) and tilt angle (pitch and roll).

The principle of modern positioning technologies that are applied in land vehicular navigation systems are:

- Dead Reckoning (DR) Systems

The principle of a DR system is based on the relative position fixing method, which requires knowledge of the starting point of the vehicle and its subsequent speed and direction in order to calculate its present position. A DR system consists of odometres which may be set up singly or in pairs onto either the wheel or the transmission of the vehicle, a compass or single axis gyroscope that measures the azimuth (heading), and quite often a tilt sensor that gives information about the pitch or roll angles of the vehicle. DR systems are relatively cheap but errors in the system increase with distance traveled, mainly in the azimuth (heading) component, as a result, the augmentation of DR systems with absolute sensors, for example GPS sensors, have become the practical solution to avoiding accumulation of errors over long periods of operation.

■ Inertial Navigation Systems (INS)

An inertial navigation system (INS) contains two core components: an inertial measurement unit (IMU) and a navigation computer. The IMU consists of three accelerometers and three gyros, whose respective input axes form an orthogonal triad, plus digitization and digital interface electronics. The accelerometers measure the specific forces that the IMU experiences, comprising accelerations and gravity with respect to the inertial reference frame. The gyros measure the angular rate that the IMU experiences comprising its angular rate with respect to the earth, along with the Earth's angular rate with respect to the inertial reference frame. The navigation computer receives the inertial data and performs two functions. First it performs an *alignment*, during which it establishes an initial orientation using the local gravity vector as the vertical reference and North component of the earth rate vector as the heading reference. After establishing a navigation frame of reference that is locally leveled and having a known azimuth (heading) with respect to North, the navigation computer then switches to its *free-inertial navigation* mode. The key advantage of an INS is that, once aligned, it navigates as a self-contained navigation without the need for external signals or communications. As a special type of DR system its position error grows with time due to alignment errors and inertial sensor errors. A detailed description of INS fundamentals is given in section 3.2.

■ Map Matching (MM)

MM allows the determination of the vehicle's position information by correlating information about features in its immediate vicinity with stored knowledge of the area, in the form of digital maps. MM uses the navigation solutions generated by various positioning systems, such as terrestrial or satellite systems. MM functions serve to increase or maintain the positioning accuracy as the vehicle already knows its position by the utilization of other positioning sensors [Zhao, 1997]. A 'local' map is formed and correlated with the stored digital map database of the area to provide the best possible estimate of the vehicle's position. Thus, such technology is usually incorporated with a Geographic Information System (GIS) and other positioning technologies, such as GPS and INS.

■ Sign Post Systems (SPS)

These systems usually utilize radio and induction coil infrastructure mounted on the sides of the streets, usually on traffic signal sites or bus stations. These sign posts or beacons are capable of communicating with vehicles, when the transceiver-equipped vehicle approaches to the proximity. The transmitted data can include various types of information, such as traffic information, segments of a map database required for route guidance, or coordinates for illustration. The main disadvantage is that it can only provide positioning information to the vehicle driving along those routes that include such an infrastructure. The density of the signal posts and the high cost maintenance are the main factors that define the limitation of such a system. For these reasons, it is generally only applied for bus service.

■ Terrestrial Radio- Navigation Systems (TRNS)

As the name implies, this system utilizes the radio-frequency (RF) signals from a number of beacons scattered around the operation area. The exact position of the vehicle is determined from the intersection of the incoming signals from several TRF beacons.

Solving the trilateration problems, the positioning information is then obtained. Examples of such systems include Omega and Loran-C. Omega provides an absolute accuracy of 4 to 8 km which makes it inapplicable for land vehicular navigation applications. Loran-C is a pulsed system but those pulses are subject to interference resulting in a low accuracy of 500 metres, especially in urban areas [Enge et al., 1995].

■ Global Navigation Satellite Systems (GNSS)

Satellite based positioning and navigation systems have been used successfully over the last two decades. The first system was the U.S. navy navigation satellite system known as TRANSIT which utilized the Doppler Effect to locate a moving vehicle. The two most sophisticated satellite systems, developed by the U.S.A and U.S.S.R., are GPS and GLONASS, respectively, which are both based on ranging from a number of satellites at known positions in orbits. In addition, a new European GNSS system, known as Galileo is being developed as well. In fact, GPS is the principal positioning system used in most of the land vehicular navigation systems at the present time. GPS is capable of providing reliable position and velocity information as well as keeping its accuracy levels for the entire mission. However, satellite signal blockage in urban areas (or by buildings, bridges or even trees) may deteriorate the overall performance of GPS. Moreover, it usually takes several minutes to evaluate the position following signal blockages. Detailed descriptions of GPS fundamentals are given in section 3.1.

2.4.2 Multi-sensor Augmented Positioning Technologies

It is very common that an individual navigation system may not meet the requirements of the positioning accuracy of a specific application. In such a situation, a multi-sensor augmented system that meets the requirements is preferred. One of the significant benefits of an integrated system is that it can augment relatively low cost sensors, which can alleviate their individual performance limitations. Furthermore, an augmented system can be implemented in a single high precision sophisticated system that incorporates a high degree of redundancy and a lower cost than a stand-alone system. A

good integrated system needs to fulfill the following characteristics that are shown in Table 2.2. Figure (2.9) illustrates the accuracy comparisons of various augmented positioning technologies previously mentioned.

Table 2.2: The requirements of an integrated navigation system

Requirement	Functional description
Continuous positioning	<i>The ability of the system to provide continuous and accurate positioning information (i.e., all environments).</i>
World-wide function	<i>It is essential for a navigation system to provide wide area applicability in order to operate everywhere.</i>
Robust-resist sensor failure	<i>The ability of the system to be reliable all the time.</i>
Cost effective	<i>The in-vehicular navigation components comprising the integrated system have to be the best compromise between cost and performance.</i>

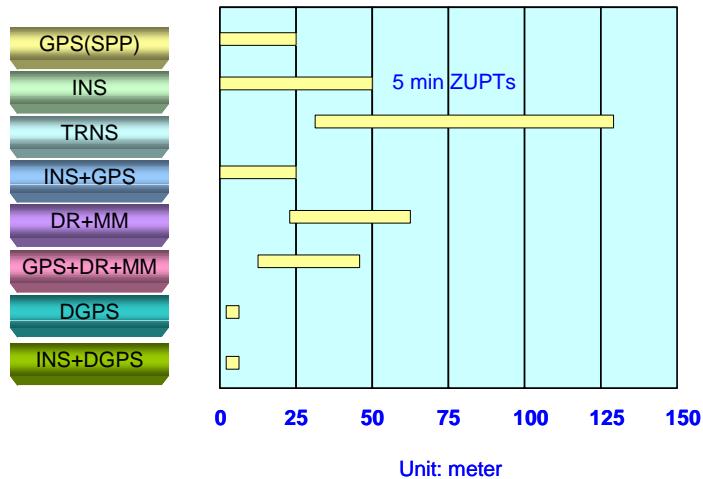


Figure 2.9: Performance comparisons of various augmented positioning technologies
(Adapted from El-Sheimy, 2000a)

As indicated in Figure (2.9), good candidates for such an integrated land vehicular navigation system are INS and GPS. Each system has its own limitations, thus the

integration of these two systems provides a system that has superior performance in comparison with either GPS, or INS stand-alone systems. For instance, GPS derived positional errors have approximately white noise characteristics over the whole frequency range. The GPS derived position and velocity information are therefore excellent external measurements for updating the INS, thus improving its long term accuracy. Similarly, the INS can provide precise position and velocity data for GPS signal acquisition and reacquisition after outages. The detailed benefits of INS/GPS integration are given in Chapter 3.

According to El-Sheimy [2000a], the trend towards integrated systems in positioning and navigation is fuelled by the demand for high accuracy, lightweight, low cost, and by technological developments which satisfy this demand. Three developments are especially important in this context: future enhancement to GPS with its modernization plan, the development of the new European GNSS system, known as Galileo and the progress in MEMS based INS systems. These issues are discussed in sections 3.1 and 3.2, respectively. Furthermore, the core components of any integrated system are the multi-sensor data fusion techniques, which are briefly discussed below:

■ Kalman filter

INS/GPS integrated systems have been common in civilian and military aeronautics applications where GPS position fixes update the accumulated error of inertial measurement sensors. When two measurements of a state, for instance position and velocity, are available where the random noise in each measurement has known distinct frequency content, a filtering technique called the complementary filter allows an optimal filter to be designed to minimize the effect of the noise on the signal estimates. The Kalman filter estimates the instantaneous state of a linear system perturbed by Gaussian white noise and provides a means of inferring information by the uses of direct and indirect measurements [Gelb, 1974; Brown and Hwang, 1992]. The Kalman filter does not have to read the required states; it can read an indirect measurement including associated noise and then estimates the required states. For example, in the INS/GPS

integrated system, the Kalman filter, using GPS measurements and INS states model, has been used to determine the error in the INS states (e.g. positions, velocities, attitudes, ...etc).

In fact, it has become a standard integration scheme of INS and GPS systems. Most of the navigation methods found in literatures use the Kalman filter approach to integrate GPS and INS. A brief description of INS/GPS integration utilizing the KF is given in section 3.3. However, as mentioned previously, several limitations of Kalman filter have been reported, and are given in section 3.3.2. Therefore, several alternative filtering based algorithms for INS/GPS integration have been investigated.

■ Alternative filters

Xu [1995] suggested a new self-learning navigation filter associated with probability space and non-Newtonian dynamics. This new filter relied basically on the information contained in measurements on the vehicle: position fixes, velocities and their error statistics. Salychev [1998] suggested scalar adaptive estimation and wave estimation algorithms. Mohamed [1999] suggested adaptive Kalman filter based INS/GPS integration architecture. Fredrik et al., [2002] proposed a framework for positioning, navigation and tracking problems using particle filters (sequential Monte Carlo methods). It consisted of a class of motion models and a general non-linear measurement equation in position. How and Deyst [2004] demonstrated the possibility utilizing the Unscented Kalman filter (UKF) as an estimation tool for solving navigation problems. Frykman [2003] suggested particle filters based aircraft integrated navigation with the utilization of INS and GPS. Shin and El-Sheimy [2004] suggested an UKF based INS/GPS integration scheme.

■ AIAs

Recently, with the advances in AIAs, such as ANNs, fuzzy logic, evolutionary computing, probabilistic computing, expert systems, and genetic algorithms, such techniques are expected to have great potential for next generation navigation algorithms.

AIA algorithms have been successfully applied in solving many engineering problems and have demonstrated their superiority over traditional methods.

Meng and Kok [1993] suggested a neural network-based navigation algorithm for a mobile robot. Cerepakhin and Greenwood [1994] proposed a neural network based method to integrate GPS and Loran-C for marine use. Townsend et al., [1994] proposed a Radial Basis Function (RBF) Networks approach for mobile robot positioning. Dumville and Tsakiri [1994], Ashkenazi et al. [1995] and Tsakiri [1996] utilized a neural network to integrate DR and GPS for land vehicle navigation. Kim and Mohan [1998] suggested a neural-fuzzy controller for mobile robot navigation. Leyden et al., [1999] proposed a fuzzy logic based navigation algorithm for a mobile robot. Chansarkar [1999] utilized RBF networks for GPS positioning and navigation. Forrest et al., [2000] suggested an inertial navigation data fusion system employing an unsupervised neural network as the data integrator to estimate INS errors. Ojeda and Borenstein [2002] and Ojeda et al., [2004] proposed a fuzzy logic rule-based position estimation algorithm for mobile robots as one of the prototypes of marsian rovers. Aguilar et al., [2003] utilized Elman recurrent neural networks to predict the yaw and pitch head movement using MEMS IMUs.

As for INS/GPS integration, Chiang and El-Sheimy[2002] and Chiang et al., [2003] first suggested an INS/GPS integration architecture utilizing Multi-Layer Feed-Forward Neural Networks (MFNNs) for fusing data from DGPS and either navigation grade IMUs or tactical grade IMUs. Chiang [2003] proposed an MFNN based INS/GPS architecture for integrating IMUs with Single Point Positioning (SPP). Chiang [2004] proposed an optimal GPS/MEMS integration architecture for land vehicle navigation utilizing neural network. In addition; Chiang and El-Sheimy [2004s] proposed the idea of developing the conceptual intelligent navigator that used ANNs approach. The preliminary results demonstrated the possibility and the potential of utilizing such algorithms as the multi-sensors data fusion algorithm for next generation land vehicular navigation. Figure (2.10) illustrates the components of a modern land vehicular navigation system.

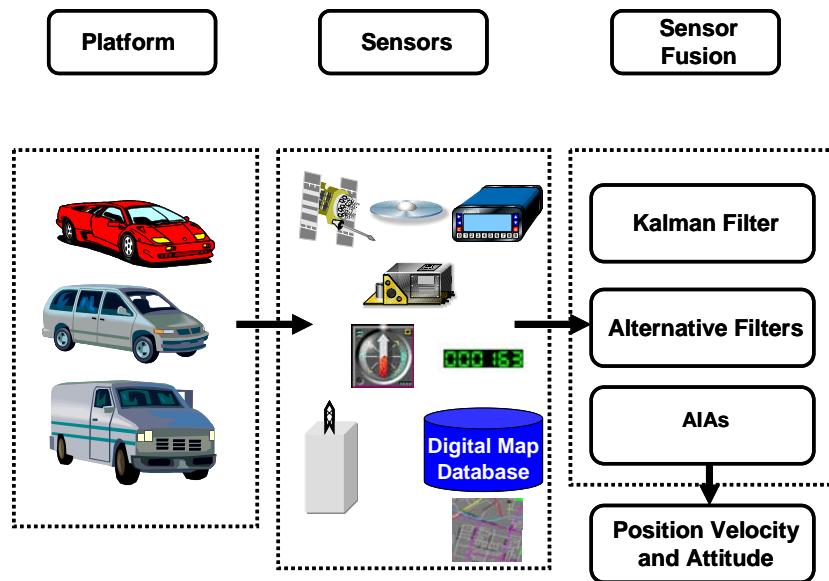


Figure 2.10: Core components of a modern land vehicular navigation system

CHAPTER 3

FUNDAMENTALS OF INS/GPS INTEGRATION

Today, most vehicular navigation systems rely mainly on the Global Positioning System (GPS) receivers as their primary source of information to provide the position of the vehicle. The system is able to provide precise positioning information to an unlimited number of users anywhere on the planet. As a result, the number of applications using GPS has increased dramatically beyond most people's imagination and the applications include tracking of where people, a fleet of trucks, trains, ships or planes are and how fast they are moving. Other applications include directing emergency vehicles to the scene of an accident, mapping where a city's assets are located, and providing precise timing for endeavors that require large-scale co-ordination. GPS, however, can provide these types of information only under ideal conditions which require an open environment (i.e. open space areas). In other words, the system doesn't work very well in urban, canopy areas due to signal blockage and attenuation deteriorating the obtainable positioning accuracy [El-Sheimy, 2000b]. Therefore, GPS has to be integrated with other sensors to bridge periods of GPS signal blockage in order to provide continuous navigation solutions.

An INS is a self-contained positioning and attitude device that continuously measures three orthogonal linear accelerations and three angular rates to calculate the required position [Jekeli, 2001]. The primary advantage of using a stand-alone INS for navigation applications is that velocity and position of the vehicle can be provided with abundant dynamic information and excellent short term performance since acceleration, angular rotation and attitude data are provided at high update rates. There are, however, disadvantages in using an INS. The position errors of an INS accumulate with time as the errors of accelerometers are double integrated; In addition, the errors of gyros result in attitude errors (i.e. the horizontal platform misalignments), which project the impact of gravity into the horizontal axes and disturb the acceleration measurement to the vehicle. Both errors grow as a function of time; therefore, an INS is only accurate for a limited time without external aiding.

An integrated system provides an enhanced navigation system that has superior performance in comparison with either a stand-alone GPS or INS as it can overcome each of their limitations. For example, The GPS derived position and velocity is excellent external aid for updating the INS, thus improving its long-term accuracy. In contrast, INS can provide precise position and velocity data to bridge the gap during GPS signal blockages. Table 3.1 illustrates the comparison of the performance between the INS, GPS and integrated system.

Table 3.1: Comparison between INS, GPS and INS/GPS

	INS	GPS	INS/GPS
Advantages	<ul style="list-style-type: none"> ✓ <i>High position, velocity accuracy over short term</i> ✓ <i>Accurate attitude information</i> ✓ <i>High measurement output rate</i> ✓ <i>Autonomous</i> ✓ <i>No signal outage</i> 	<ul style="list-style-type: none"> ✓ <i>High position, velocity accuracy over long term</i> ✓ <i>Uniform accuracy, independent of time</i> ✓ <i>Not sensitive to gravity</i> 	<ul style="list-style-type: none"> ✓ <i>High position and velocity accuracy</i> ✓ <i>Precise attitude determination</i> ✓ <i>High data rate</i> ✓ <i>Navigation output during GPS outage</i>
Disadvantages	<ul style="list-style-type: none"> ✓ <i>Accuracy deteriorates with time</i> ✓ <i>Affected by gravity</i> ✓ <i>Required Initial and In-flight calibration and alignment</i> 	<ul style="list-style-type: none"> ✓ <i>Noisy attitude information</i> ✓ <i>Low measurement output rate</i> ✓ <i>Non autonomous</i> ✓ <i>Cycle slip and loss of lock</i> 	<ul style="list-style-type: none"> ✓ <i>Cycle slip detection</i> ✓ <i>Reduction of GPS signal search time</i> ✓ <i>Gravity vector determination</i> ✓ <i>Jamming resistance</i>

In this chapter, the fundamentals of GPS and INS are presented to review the advantages and disadvantages of each system and the benefits of an INS/GPS integrated system. The future development of the satellite and INS based positioning technologies are discussed. After that, the fundamentals of Kalman filtering and associated INS/GPS integration architectures are reviewed. Finally, the limitations of the Kalman filter based INS/GPS integration algorithm are identified.

3.1 Fundamentals of GPS

The Navigation System with Timing and Ranging (NASTAR) GPS is a satellite based radio-navigation system developed and maintained by the Joint Program Office (JPO), which is directed by the U.S Department of Defense (DoD). The system consists of the space segment consisting of satellites that broadcasts signals, the control segment steering the world wide system, and the user segment including the many types of GPS receivers available [Hofmann-Wellenhof et al., 2001]. The current space segment consists of 28 Block II/IIA/IIR satellites (www.navcen.uscg.gov/Ftp/gps/status.txt) occupying six orbital planes inclined at a 55-degree angle with respect to the equator. GPS satellites broadcast navigation messages and provide a global, 24-hour and all-weather navigation service [Misra and Enge, 2001].

The orbit of GPS satellites is 20200 km about the Earth's surface with a period of about 12 hours. The transmitted signals contain two frequencies; L1 at 1575.42 MHz and L2 at 1227.6 MHz. At the present time, these signals are bi-phase modulated by one or two PRN codes; the Coarse/Acquisition, C/A code, and the precise code, P code. The L1 carrier is modulated by both C/A and P codes while the L2 carrier is only modulated by P code. The C/A code is transmitted at 1/10 of the fundamental GPS frequency (10.23 MHz) and is repeated every one millisecond. On the contrary, the P code is transmitted at the fundamental frequency and is only repeated every 267 days. The navigation message that contains the broadcast ephemeris and health information is modulated on both frequencies at 50 bits per second [Parkinson et al., 1996].

The C/A code is unrestricted and is designated for the Standard Positioning Service (SPS), where Single Point Positioning (SPP) accuracies of 13 metres horizontally and 22 metres vertically can be achieved at a 95% confidence interval [NSTB/WASS T&E Team, 2003]. The accurate P code provides Precise Positioning Services (PPS) to authorized user. The restriction is accomplished utilizing Anti-Spoofing (A-S), where the P code is translated to obtain the encrypted Y code except for authorized users [Parkinson et al., 1996].

The control segment comprises the operational control system that consists of a Master Control Station (MCS), five monitor stations and four ground antenna upload stations. The main operational task of this segment is to maintain the satellite in its orbit through commanded maneuvers, and generate satellite clock and orbit corrections. It uploads the corrections to the satellites thus they can be broadcasted to user through the navigation data [Hofmann-Wellenhof et al., 2001].

The user segment is basically different types of GPS receivers, which use direct line of sight GPS satellite signals to determine the user's position, velocity and time with accuracies specified by various receivers. A GPS receiver measures the apparent transit time of the satellite signal from the satellite to the user, known as pseudorange; it consists of the propagation delay and receiver clock bias. By using at least four such measurements and knowing the satellite position from the ephemeris data, the problem is reduced to determining four unknown parameters, the receiver's three position components and the clock bias, from known measurements(i.e., pseudoranges) which are expressed as follows [Misra and Enge, 2001]:

$$P_i = \sqrt{(x_i - x_u)^2 + (y_i - y_u)^2 + (z_i - z_u)^2} - b \quad (3.1)$$

where,

- i is the satellite index ,
- P is the pseudorange (m),
- x_i, y_i, z_i are the coordinates of ith satellite (m, m, m),
- x_u, y_u, z_u are the coordinates of user (m, m, m),
- b is the receiver clock bias (s)

The user position vectors and clock bias can be determined by linearizing the above equation with respect to some initial values and using least squares or Kalman filtering algorithms. In general, the solution becomes over-determined and the redundant measurements improve the accuracy and integrity as measurements are available from more than four satellites.

3.1.1 GPS Observables and Positioning Principles

■ Single point positioning and observables

The essential GPS observables are pseudorange, carrier phase and the instantaneous Doppler frequency. As mentioned briefly in the previous section, the pseudorange is measured by comparing the replica of C/A code generated in the receiver with the code transmitted from the satellite to determine the time shift through an autocorrelation process. The carrier phase measurement is the accumulated phase offset between the receiver reference signal and the received satellite signal. As a result, the initial number of integer cycles in the carrier phase is unknown [Hofmann-Wellenhof et al., 2001]. The unknown integers are known as ambiguities. The basic pseudorange measurement that contains various error components is given as follows [Lachapelle, 2001]:

$$P = \rho + d\rho + c(dt - dT) + d_{ion} + d_{trop} + \varepsilon_{Mp} + \varepsilon_p \quad (3.2)$$

where,

P	is the pseudorange measurement (m),
ρ	is the geometric range between the satellite and receiver antennas (m),
$d\rho$	is the orbital error (m),
c	is the speed of light (m/s),
dt	is the satellite clock error (s),
dT	is the receiver clock error (s),
d_{ion}	is the ionospheric error (m),
d_{trop}	is the tropospheric error (m),
ε_{Mp}	is the code range multipath error (m) , and
ε_p	is the receiver code noise error (m).

Similar to the code measurement, the carrier phase measurement contains many error components and is given as follows [Lachapelle, 2001]:

$$\phi = \rho + d\rho + c(dt - dT) + \lambda N - d_{ion} + d_{trop} + \varepsilon_{M\phi} + \varepsilon_\phi \quad (3.3)$$

where,

- ϕ is the measured carrier phase (m),
- λ is the carrier wavelength(m),
- N is the integer cycle ambiguity (cycles),
- $\varepsilon_{M\phi}$ is the carrier phase multipath error (m) , and
- ε_ϕ is the receiver carrier noise error (m).

The difference between code range and carrier phase measurements can be seen from Equation 3.2 and 3.3 and are described as follows:

- The ionospheric delay error has an opposite sign in the two expressions, due to the impact of group delay and phase advance on the code and phase measurement, respectively
 - The code multipath error and code noise out of Equation 3.2 are replaced by carrier multipath error and carrier noise error in Equation 3.3, and
 - Equation 3.3 contains one additional term corresponding to the integer ambiguity.
- Differenced Positioning and observables

Single Point Positioning (SPP) is inaccurate due to various error sources, as indicated in Equation (3.2) and (3.3). However, some of these errors are spatially correlated. The degree of correlation between errors at two receivers is a function of the baseline length. As a result, correlated error sources can be reduced by differencing the observation equations (Equation (3.2) or (3.3)) of the two receivers. One of the differencing techniques is single differencing, known as *between-receiver single difference*, which reduces the orbit and atmospheric error terms. The positioning equation for both pseudorange and carrier phase with the between-receiver single differenced observation can be formed as follows [Lachapelle, 2001]:

$$\Delta P = \Delta \rho + \Delta d \rho - c \Delta d T + \Delta d_{ion} + \Delta d_{trop} + \Delta \varepsilon_{Mp} + \Delta \varepsilon_p \quad (3.4)$$

$$\Delta\phi = \Delta\rho + \Delta d\rho - c\Delta dT + \lambda\Delta N - \Delta d_{ion} + \Delta d_{trop} + \Delta\varepsilon_{M\phi} + \Delta\varepsilon_\phi \quad (3.5)$$

where Δ represents a between-receiver single difference

Satellite clock error terms have vanished in Equation (3.4) and (3.5), as it is the same for the two receivers at a particular time epoch. Other error terms have now become the difference of errors between the two receivers. As a result, a high degree of correlation of errors between receivers results in the cancellation of the error in the differenced equation. For a short baseline, the orbital error, ionospheric delay error and the tropospheric delay error are highly correlated, thus the residual error can be assumed to be very small. The receiver clock bias, integer ambiguity, multipath and receiver noise still remain. Consequently, multipath error is perhaps the most dominant source of error in the single differenced measurements in the short baseline. On the contrary, the residual orbital, ionospheric and tropospheric error terms become significant compared to multipath error for a long baseline.

The difference in the true range term ($\Delta\rho$) now refers to the difference in distances, where the first distance is between receiver A and the satellite, and the second distance is between receiver B and the satellite. If the satellite position and one of the receiver's positions are known (i.e., reference station), the position of the other receiver can be determined. This is the concept of differential positioning. By further differencing the between-receiver single difference across two different satellites, a between-receiver-between-satellite double difference can be obtained as follows [Lachapelle, 2001]:

$$\Delta\nabla P = \Delta\nabla\rho + \Delta\nabla d\rho + \Delta\nabla d_{ion} + \Delta\nabla d_{trop} + \Delta\nabla\varepsilon_{Mp} + \Delta\nabla\varepsilon_p \quad (3.6)$$

$$\Delta\nabla\phi = \Delta\nabla\rho + \Delta\nabla d\rho + \lambda\Delta\nabla N - \Delta\nabla d_{ion} + \Delta\nabla d_{trop} + \Delta\nabla\varepsilon_{M\phi} + \Delta\nabla\varepsilon_\phi \quad (3.7)$$

where ∇ represents a between-satellite single difference thus $\Delta\nabla$ is the double difference operator

The receiver clock error term, $c\Delta dT$ has vanished in both Equation (3.6) and (3.7), as it is the same for the two satellites observed at the same time. The residual orbital,

ionospheric and tropospheric errors are very small, causing multipath to be the dominant source of error in the double differenced measurements for a short baseline. However, for a long baseline, the residual orbital, ionospheric and tropospheric error terms become significant over the multipath error term. There is statistically twice as much receiver noise in the case of a single measurement [Raquet, 1998].

The benefit of the double differenced observable is that the receiver clock error term is eliminated. Consequently, if the residual errors are small, the double differenced ambiguity term can be resolved to an integer value. While forming double differenced measurements between satellite pairs, one of the satellites is kept common in all the pairs. That satellite is called the base satellite, which is generally the satellite with highest elevation, as the highest elevation satellite is likely to have least amount of multipath, atmospheric delay errors and phase noise.

In general, pseudorange measurements are used in single differenced form, while carrier phase measurements are used in double differenced form where a high positioning accuracy is required [Hofmann-Wellenhof et al., 2001]. However, double differenced ambiguity term $\Delta \nabla N$ needs to be solved. According to the characteristics of differential positioning, the main errors that affect the ambiguity resolution for a short base line (<10 km) are carrier phase multipath and receiver noise.

However, ionospheric and tropospheric errors become more significant for long baselines. Positioning accuracy with the fixed ambiguities has positioning accuracy at the centimeter level [Hofmann-Wellenhof et al., 2001]. Table 3.2 gives a brief summary of these errors mentioned above and their mitigation in differential mode. Where the estimates for DGPS are based on the premise that the user's distance from the base station is tens of kilometers and signal latency is tens of seconds [Misra and Enge, 2001]. See Raquet [1998], Misra and Enge[2001], and Lachapelle [2001] for details about GPS errors and their mitigation.

Table 3.2: A Summary of GPS errors source (After Misra and Enge, 2001)

Source	Potential Error Size	Residual Error and Mitigation
Satellite clock error	<i>Clock modeling error: 2m (rms)</i>	<i>DGPS: 0.0m, completely removed via DGPS</i>
Satellite ephemeris prediction	<i>Component of the ephemeris prediction error along the line: 2m (rms)</i>	<i>DGPS: 0.5ppm (rms), The post-DGPS-corrected residual error is spatial correlated, thus can only be reduced via DGPS.</i>
Ionospheric delay	<p><i>Effect upon the code and the carrier is equal but opposite sign: The code is delay while the carrier is advanced by the same amount.</i></p> <p><i>Delay in zenith direction: 2~10 m, depending on user latitude, time of delay, and solar activity.</i></p> <p><i>Delay for a satellite at elevation angle E=zenith delay * oblique factor (E), oblique factor: 1 at zenith, 1.8 at 30° elevation angle; and 3 at 5°</i></p>	<p><i>Single-frequency receiver using broadcast model: < 10m</i></p> <p><i>Dual-frequency receiver (compensate for the ionospheric delay but magnifies noise): 1m (rms)</i></p> <p><i>DGPS: 0.2-20 ppm (rms), The post- DGPS-corrected residual error is spatial correlated, thus can only be reduced via DGPS.</i></p>
Tropospheric delay	<p><i>Code and the carrier are both delayed by the same amount.</i></p> <p><i>Delay in zenith direction at sea level: 2.3~2.5 m, lower at higher altitude</i></p> <p><i>Delay for a satellite at elevation angle E=zenith delay * oblique factor (E), oblique factor: 1 at zenith, 2 at 30° elevation angle; 4 at 15°; and 10 at 5°</i></p>	<p><i>Model based on average meteorological conditions: 0.1~1m</i></p> <p><i>DGPS: 0.2-0.4ppm (rms) plus altitude effect; The post- DGPS-corrected residual error is spatial correlated, thus can only be reduced via DGPS.</i></p>
Multipath	<p><i>In a ‘clean’ environment:</i></p> <p><i>Code: 0.5-1m</i></p> <p><i>Phase: 0.5~1cm</i></p>	<p><i>Uncorrelated between antennas.</i></p> <p><i>Mitigation through antenna design and siting, receiver design, and carrier-smoothing of code measurements</i></p>
Reviver noise	<p><i>Code: 0.25-0.5m(rms)</i></p> <p><i>Phase: 1~2mm(rms)</i></p>	<p><i>Uncorrelated between receivers.</i></p> <p><i>Mitigation through antenna design</i></p>

3.1.2 Future Development of GNSS Positioning Technologies

As mentioned in chapter 2, the future development of GNSS based positioning technologies include 1) the modernization of GPS and 2) the deployment of the European GNSS system, known as Galileo. Indeed, the modernization of GPS and the advance of Galileo will lead to a truly multi-frequency civil GNSS. Undoubtedly, both systems will be highly beneficial to land vehicular navigation applications. A brief introduction of GPS modernization and Galileo and their impacts on land vehicular navigation are given in this and the next section, respectively.

According to Misra and Enge [2001], current constellation of GPS Block II and IIA satellites has some limitations. First of all, for stand-alone operations in real time civilian users (i.e., land vehicles) have only access to the C/A code on L1. As a result, a dual-frequency ionospheric delay correction can not be applied for these users. Further, L1 and L2 signals are very weak; therefore, they can not penetrate into concrete and steel buildings or underground and are susceptible to interference and jamming. GPS modernization plans were first announced in 1998. The detail content of the plans is beyond the scope of this research and can be found in McDonald [2002].. The evolution of GPS positioning capability in autonomous mode after GPS modernization is given in Table 3.3.

Table 3.3: GPS positioning capability in autonomous mode (After Misra and Enge, 2001)

Year	Signal Status	Horizontal Position Accuracy (95%)
1999	C/A-coded signal on L1, SA active	20-100 m
2000	C/A-coded signal on L1, SA off	10-20 m
2010	C/A-coded signal on L1 and L2C	5-10 m
2015	C/A-coded signal on L1, L2C on L2, new civil signal on L5	1-5 m

The impacts of GPS modernization on land vehicular navigation are given as follows:

- Improving positioning accuracy in autonomous mode

Since the SA (Selective Availability) turn off on May 2000, the remaining dominate error source is the ionospheric delay error. However, with two modernized civilian signals, civilian users are able to estimate the ionospheric delay correction to improve the positioning accuracy of better than ten meters autonomously. According to Table 3.4 and Table 2.1, the positioning accuracy of SPP can be improved to 1 metre level which is accurate enough for most of the land vehicle navigation applications.

- Replacing DGPS with SPP for land vehicular navigation applications

DGPS is the only option for providing sub-meter level positioning accuracy for land vehicular navigation at the present time as it can reduce the impact of spatially correlated error sources (i.e., ionospheric delay, tropospheric delay and orbital error). However, such a technique is not suitable for general users as additional communication links needs to be established between the rover vehicle and the base station for receiving and transmitting differential correction data.

As mentioned previously, the distribution density of such a base station is a critical factor for providing consistent positioning accuracy. For making such a system available, users might have to purchase the service. Fortunately, with modernized GPS signals, providing meter level positioning accuracy in SPP mode (autonomous mode) becomes possible, as indicated in Table 3.3. Consequently, the additional cost of purchasing telematics services can be eliminated, as cost is always the major concern for the general user. In other words, SPP can provide a cost effective solution for land vehicles with modernized GPS signals.

On the other hand, Galileo is the European global navigation satellite system, providing a highly accurate, guaranteed global positioning service under civilian control. It will be compatible with GPS. Galileo will deliver real-time positioning accuracy down to the meters range by offering dual frequencies as standard service for civilian users, which is unprecedented for current GPS. However, the modernized GPS will eventually provide three frequencies for civilian users. The space segment of Galileo is intended to consist of a

total 30 Mean Earth Orbiting (MEO) satellites configured as a Walker constellation, distributed over three orbital planes. The altitude is 23616 km, and the inclination is 56°, see Hein et al [2002] for the details about Galileo.

Table 3.4 illustrates the significant advantages of having modernized GPS and Galileo available to users worldwide. According to O’Keefe [2001] and McDonald [2002], the benefits of the future multi-frequency GNSS include improvements in system accuracy and reliability, satellite availability, autonomous integrity, interference mitigation, urban canyon operation and in kinematic precision measurements.

Table 3.4: The GPS-Galileo GNSS characteristics (After McDonald, 2002)

Characteristic	GPS + WAAS	Galileo + EGNOS	Combined Capability
Spacecraft in Orbit	$28 + 3$	$30 + 3$	$58 + 6$
Spacecraft Availability (aver.)	$8 - 9$	$8 - 9+$	$16 - 18$
Integrity (autonomous)	<i>Fair</i>	<i>Fair</i>	<i>Excellent</i>
Coverage (worldwide) (nominal HDOP-VDOP)	<i>Good</i>	<i>Good</i>	<i>Excellent</i>
Dilution of Precision	$1 - 3$	$1 - 3$	$0.7 - 2$
Interference Susceptibility	<i>Low</i>	<i>Low</i>	<i>Very Low</i>
Safety Services Protection	<i>2 signals</i>	<i>4+ signals</i>	<i>6+ signals</i>
Frequencies Available (civil)	$1 - 3$	$1 - 5$	$2 - 8$
E911, Related Capabilities	<i>Fair</i>	<i>Fair</i>	<i>Very Good</i>
Accuracy (Autonomous, code)*	$1\text{-}2 m.$	$1\text{-}2 m.$	$0.6\text{-}1.3 m.$

3.2 Fundamentals of INS

As mentioned in the previous chapter, most of the current INS technologies were developed within the last three decades [Greenspan, 1995]. The basic principle of inertial navigation is based on Newton’s first and second laws of motion. The basic principle is based on double integration of the observed acceleration obtained by accelerometers that are mounted on the body of the vehicle. The first integration of the vehicle acceleration generates velocity and the second integration yields distances (which leads to position).

However, the initial navigation conditions of the system (position, velocity, and orientation) have to be available for the integration process. Therefore, this requirement makes INS a relative positioning system (i.e. DR system). On the contrary, GNSS are absolute positioning systems. To determine the position and velocity in the desired navigation frame (i.e., l-frame), the accelerations have to be projected from the body frame (b-frame) to the navigation frame of interest before integration. A cluster of gyroscopes that are orthogonally mounted and placed along the same axes with the accelerometers are applied to measure the angular velocity of the accelerometers and to update its orientation with respect to the navigation frame. An INS consists of the following components:

- An IMU that contains two orthogonal sensor triads:
 - The accelerometers which measure the vehicle's acceleration and its linear motion in three mutually orthogonal directions.
 - The gyroscopes which measure the vehicle's angular velocity and its angular motion in three mutually orthogonal directions
 - The sensors are fixed in the body of the IMU and are therefore called the body axes
- Navigation computers :
 - For the implementation of the alignment process (Section 3.2.2) and the mechanization equation (Section 3.2.3)

It should be noted here that the raw accelerometer output must be corrected for the gravitational acceleration, g , before the actual acceleration of the vehicle, a , can be obtained. Therefore, the output of an accelerometer is called a specific force, f , and is given as follows [Savant Jr. et al., 1961];

$$f = a - g \quad (3.8)$$

There are many different designs of the inertial sensors with different performance characteristics and there are two ways to implement the IMU: 1) gimbaled (stable

platform) implementation 2) strapdown implementation. These two implementations are described in the next section.

3.2.1 Physical Implementations of IMU

1. Gimbaled (stable platform) implementation: The accelerometers are mounted on a gimbaled platform which is kept aligned to the navigation frame (i.e., local level frame) by gyro feedback. This is done by torquing the gimbals based on the changes of the navigation frame [Savant Jr. et al., 1961]. As a result, the accelerometer outputs after applying the gravity corrections can be integrated to provide velocity and position in the navigation frame. The advantage of a gimbaled system is that no coordinate transformation is required, thus the navigation computation is made simple. However, such systems encounter problems when the local level frame (l-frame) is applied as the navigation frame, and the system works in the polar region as the control torque becomes very large. Consequently, the wandering mode is applied instead of local level mode in a high latitude area [Salchev, 1998]. Figure (3.1) illustrates the operation of a gimbaled system utilizing the local level frame as the navigation frame. When the gimbaled system moves from point A to point B, the local level frame rotates an angle with respect to the inertial space, and the gimbal platform tracks the rotation of the local level frame. Consequently, the gimbal platform axes still coincide with the local level frame.

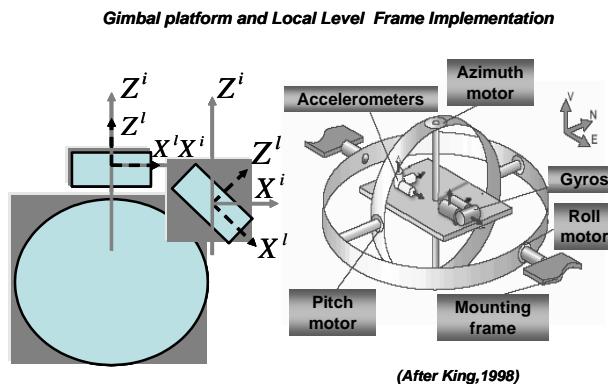


Figure 3.1: Gimbaled platform and local level frame implementation

2. *Strapdown implementation*: The accelerometer and gyros are mounted directly on to the body of the vehicle to form a body frame, as shown in the Figure (3.2). The rotation rates measured by the gyros are used to constantly update the transformation matrix between the body frame (b-frame) and the navigation frame (i.e., l-frame). The observed acceleration is passed through the transformation to obtain the acceleration in the navigation frame after applying the gravity correction.

The reasons for the rapid advance of strapdown systems are their advantages in power consumption, weight, cost, flexibility and stability. Undoubtedly, the recent development of the computer-related technologies and optical gyros has been accelerating the rapid development of strapdown systems [Titterton and Weston, 1997]. The essential difference between the strapdown and the gimbaled system concerns the transformation from the body frame (b-frame) to navigation frame (l-frame). With the gimbaled system, the transformation is done mechanically with torque commands to the servos of the platform gimbals and to the gyros. These torque-induced rotations stabilize the platform so that it is always parallel to the desired frame. On the other hand, the transformation is accomplished computationally in the strapdown implementation. Table 3.5 illustrates the comparison between the two implementations.

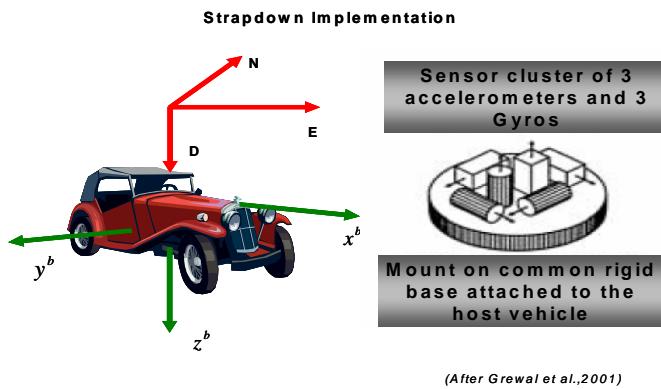


Figure 3.2: Strapdown frame implementation

Table 3.5: Comparison between gimbaled/strapdown implementation (After El-Sheimy, 2004a)

Characteristics	Strapdown Implementation	Gimbaled Implementation
Weight	<i>Light</i>	<i>Heavy</i>
Size	<i>Small/compact</i>	<i>Relatively big</i>
Performance	<i>Navigation-grade accuracy</i>	<i>Navigation-grade accuracy or better</i>
Environment	<i>High reliability with immunity to shock and vibrations forces</i>	<i>High reliability with less immunity to shock and vibrations forces</i>

3.2.2 Initialization of Strapdown INS

Certain initialization procedures are required prior to computing the navigation parameters. The strapdown INS initialization process contains the following two stages:

1. *Calibration*: In most of the high-end or navigation grade INS, the gyroscope and accelerometer bias and scale factors are usually calibrated by the manufacturer and therefore no separate calibration is required every time the sensor is used. However, periodical calibration every few months is recommended. For low cost sensors, however, the bias and the drift stability of the gyros are much poorer and frequent calibration is very important. For instance, the calibration of a gyro is performed by averaging the raw angular velocities measured from the gyros over 10~20 minutes in static mode and comparing with the true value to estimate the drift. For more details, see Chatfield [1997] and Jekeli [2001].

To start the INS calculations, the initial parameters of the transformation matrix, C_b^l , (rotation matrix from the body frame to the local level frame, roll, pitch and azimuth) are required. The process of computing the initial parameters of the transformation matrix is called the INS alignment procedure. There are two alignment modes, namely the *Accelerometer Leveling* (roll and pitch) and the *Gyro Compassing* (azimuth).

I) Alignment_ Accelerometer Leveling: Aligns the Z^b to Z^l by forcing the two planimetric accelerometers, X^b and Y^b , measurements to zero. Consequently, the true vertical is established and these two accelerometers can be considered being located in a level plane. In strapdown systems, the accelerometer leveling is conducted mathematically using the following equations which assume the measurements f_x^b and f_y^b are tilted with respect to the vertical direction defined by the gravity vector [Rogers, 2003]:

$$\text{Roll} = \phi = \sin^{-1}\left(\frac{f_x^b}{g}\right) \approx \frac{f_x^b}{g} \quad (3.9)$$

$$\text{Pitch} = \theta = \sin^{-1}\left(\frac{f_y^b}{g}\right) \approx \frac{f_y^b}{g} \quad (3.10)$$

Where ϕ and θ are assumed to be very small angles.

The above equations illustrate the accelerometer leveling errors. Hence, the accuracy of the accelerometer leveling procedure is limited by the accelerometer biases, and as such the errors in the computed roll and pitch angles can be computed as follows:

$$\delta\phi = \frac{b_{fx}}{g} \quad (3.11)$$

$$\delta\theta = \frac{b_{fy}}{g} \quad (3.12)$$

II) Alignment_ Gyro Compassing: The gyro compassing is realized by using the gyro measurements. It assumes that the accelerometer leveling has been accomplished. However, the X^b and Y^b accelerometers can be arbitrary rotated with respect to the l-frame with an angle, called the azimuth angle (ψ). According to the definition of the body-frame which has been given in section 2.1 and Figure (3.2), the X-gyro and Y-gyro measurements in this situation are given by the following equations [Rogers, 2003]:

$$w_x^b = w_e \cos \varphi \cos \psi \quad (3.13)$$

$$w_y^b = -w_e \cos \varphi \sin \psi \quad (3.14)$$

Consequently, the azimuth angle is obtained as follows:

$$\psi = \tan^{-1} \left(-\frac{w_y^b}{w_x^b} \right) \quad (3.15)$$

$$\delta\psi = \frac{b_{w_y^b}}{w_e \cos \varphi}, (\psi \leq 0) \quad (3.16)$$

The gyro compassing is also dependent on the gyro drift $b_{w_y^b}$ and can be on the order of a few arcminutes for a good quality IMU to a few degrees for a low cost IMU. The Earth rotation rate is $15^\circ/\text{hour}$ ($0.004167^\circ/\text{second}$) whereas the gyro bias variations can be as large as $1.0^\circ/\text{second}$ in the case of low cost IMUs. This means that the gyros can not be used to perform a self contained azimuth alignment. Consequently, Gyro Compassing cannot be accomplished as the gyro drift exceeds the Earth rotation rate. Hence for gyros with drift rates larger than the Earth rotation rate, Gyro Compassing is usually accomplished via external aid (e.g. a magnetic compass). For more details about other azimuth aiding techniques, see Titterton and Weston [1997] and Farrell and Barth [1998].

3.2.3 INS Mechanization Equations

The raw measurements from accelerometers and gyros are specific forces and angular velocities which are measured along the body frame (b-frame). The navigation frame is where the data integration is performed. The choice of the navigation is application dependent. The local level frame is often selected as the navigation frame for the following reasons [El-Sheemy, 2004a]:

- The definition of the local level frame is based on the normal to the reference ellipsoid; as a result, the geodetic coordinate difference $\{\Delta\varphi, \Delta\lambda, \Delta h\}$ can be applied as the output of the system.
- The axes of the local level frame (NED) are aligned with the local north, east and down directions. Therefore, the attitude angles (roll, pitch and azimuth) can be obtained directly as an output of the mechanization equations.
- Due to the Schuller's effect, the computational errors in the navigation parameters on the horizontal plane are bounded.
- Computation efficiency

■ Equation of Motions

According to Newton's second law of motion, the fundamental equation for the motion of a particle in the field of the earth, expressed in an inertial frame, is of the form

$$\ddot{r}^i = f^i + \bar{g}^i \quad (3.17)$$

where,

- | | |
|--------------|------------------------------|
| \ddot{r}^i | is the acceleration vector |
| f^i | is the specific force vector |
| \bar{g}^i | is the gravitational vector |

The above equation of motion can be transformed into the local level frame (l-frame, NED) and can be expressed as a set of first order differential equations. For more details, see Shin [2001].

$$\begin{bmatrix} \dot{r}^l \\ \dot{v}^l \\ \dot{C}_b^l \end{bmatrix} = \begin{bmatrix} D^{-1}v^l \\ C_b^l f^b - (2w_{ie}^l + w_{el}^l) \times v^l + g^l \\ C_b^l (\Omega_{ib}^b - \Omega_{il}^b) \end{bmatrix} \quad (3.18)$$

$$D^{-1} = \begin{bmatrix} \frac{1}{M+h} & 0 & 0 \\ 0 & \frac{1}{(N+h)\cos\varphi} & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (3.19)$$

The specific force f^b is the raw output measured by the accelerometer and is defined as the difference between the true acceleration in space and the gravitational acceleration. The transformation matrix from the b-frame to l-frame, C_b^l , is given previously in Equation 2.4.

M and N are radii of curvature in the meridian and prime vertical, respectively and can be expressed as follows [Schwarz and Wei, 2000].

$$N = \frac{a}{(1-e^2 \sin^2 \varphi)^{\frac{1}{2}}} \quad (3.20)$$

$$M = \frac{a(1-e^2)}{(1-e^2 \sin^2 \varphi)^{\frac{3}{2}}} \quad (3.21)$$

Where a and e are the semi-major axis and linear eccentricity of the reference ellipsoid, respectively.

The position vector in the l-frame is given by curvilinear coordinates that contain latitude, φ , longitude, λ , and ellipsoidal height, h :

$$\mathbf{r}^l = [\varphi \ \lambda \ h]^T \quad (3.22)$$

The velocity vector in the l-frame is given as follows:

$$\mathbf{v}^n = \begin{bmatrix} v_N \\ v_E \\ v_D \end{bmatrix} = \begin{bmatrix} (M+h) & 0 & 0 \\ 0 & (N+h)\cos\varphi & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \dot{\varphi} \\ \dot{\lambda} \\ \dot{h} \end{bmatrix} \quad (3.23)$$

Where v_N, v_E and v_D are north, east and downward velocity components. The gravity vector in the local level frame, \mathbf{g}^l , is expressed as the normal gravity at the geodetic latitude φ and ellipsoidal height h [Schwarz and Wei, 2000]

$$\mathbf{g}^l = [0 \quad 0 \quad \gamma]^T, \gamma = a_1(1 + a_2 \sin^2 \varphi + a_3 \sin^4 \varphi) + (a_4 + a_5 \sin^2 \varphi)h + a_6 h^2 \quad (3.24)$$

Where a_1 to a_6 are constant values and are listed in the Table 3.6

The rotation rate vector of the e-frame with respect to the i-frame projected into the e-frame is expressed as follows:

$$\mathbf{w}_{ie}^e = [0, 0, w_e]^T \quad (3.25)$$

Table 3.6: Constant coefficients for normal gravity

$a_1(m/\text{sec}^2)$	9.7803267715	$a_4(m/\text{sec}^2)$	-0.0000030876910891
$a_2(m/\text{sec}^2)$	0.0052790414	$a_5(m/\text{sec}^2)$	0.0000000043977311
$a_3(m/\text{sec}^2)$	0.0000232718	$a_6(m/\text{sec}^2)$	0.0000000000007211

Projecting the vector to the l-frame utilizing Equation 2.2 gives

$$\mathbf{w}_{ie}^l = C_e^l \mathbf{w}_{ie}^e = [w_e \cos \varphi \quad 0 \quad -w_e \sin \varphi]^T \quad (3.26)$$

The transport rate represents the turn rate of the l-frame with respect to the e-frame and is given using the rate of change of latitude and longitude which are given as follows:

$$\mathbf{w}_{el}^l = [\dot{\lambda} \cos \varphi \quad -\dot{\varphi} \quad -\dot{\lambda} \sin \varphi]^T = \left[\frac{v_E}{(N+h)} \quad \frac{-v_N}{(M+h)} \quad \frac{-v_E \tan \varphi}{(N+h)} \right]^T \quad (3.27)$$

Ω_{ie}^l and Ω_{el}^l are the skew symmetric matrices corresponding to w_{ie}^l and w_{el}^l , respectively.

The angular velocity w_{ib}^b , is the raw output measured by the gyros and Ω_{ib}^b is its skew-symmetric matrix.

$$w_{ib}^b = \begin{bmatrix} w_x & w_y & w_z \end{bmatrix}^T \quad (3.28)$$

The angular velocity Ω_{il}^b is subtracted from Ω_{ib}^b to remove 1) earth rotation rate and 2) orientation change of the local level frame. As a result, Ω_{il}^b is expressed as follows:

$$\Omega_{il}^b = \Omega_{ie}^b + \Omega_{el}^b \quad (3.29)$$

Thus, w_{il}^b can be obtained as follows

$$w_{il}^b = C_l^b (w_{ie}^l + w_{el}^l) = C_l^b w_{il}^l = C_l^b \begin{bmatrix} w_e \cos \varphi + \frac{v_E}{(N+h)} & -v_N & -w_e \sin \varphi - \frac{v_E \tan \varphi}{(N+h)} \end{bmatrix}^T \quad (3.30)$$

Consequently, the Ω_{il}^b can be obtained through w_{il}^b .

■ Mechanization Equations

The mechanization equations are applied to solve the equations of motion in order to obtain the position, velocity and attitude increment. In reality, strapdown IMUs work in discrete form and provide velocity and angle increments (Δv_f and $\Delta \theta_{ib}^b$, respectively) over the time interval t_k to t_{k+1} in the body frame. Combining these increments with the initial condition of the system, the navigation information can be obtained. According to Schwarz and Wei [2000], the mechanization equations consist of four basic steps:

■ Error Compensation

The accelerometer and gyros outputs are corrected utilizing the following equations

$$\Delta v_f = \begin{bmatrix} \frac{1}{(1+S_{fx})} & 0 & 0 \\ 0 & \frac{1}{(1+S_{fy})} & 0 \\ 0 & 0 & \frac{1}{(1+S_{ fz})} \end{bmatrix} (\Delta \tilde{v}_f - b_f \Delta t) \quad (3.31)$$

$$\Delta \theta_{ib}^b = \Delta \tilde{\theta}_{ib}^b - b_w \Delta t \quad (3.32)$$

Where S_{fx} , S_{fy} and $S_{ fz}$ are the scale factors of the accelerometers. b_f and b_w are the biases of the accelerometer and gyro, respectively. $\Delta \tilde{v}_f$ and $\Delta \tilde{\theta}_{ib}^b$ are raw outputs of accelerometers and gyros, respectively. Δv_f and $\Delta \theta_{ib}^b$ are compensated outputs of accelerometers and gyros, respectively. $\Delta t = t_{k+1} - t_k$.

■ Attitude Integration

The body frame angular increment with respect to the navigation frame (l-frame) is given by the following equation:

$$\begin{aligned} \Delta \theta_{lb}^b &= [\Delta \theta_x \quad \Delta \theta_y \quad \Delta \theta_z] \\ &= \Delta \theta_{ib}^b - C_l^b (w_{ie}^l + w_{el}^l) \Delta t \end{aligned} \quad (3.33)$$

The magnitude of the angular increment is given by the following equation:

$$\Delta \theta = \sqrt{\Delta \theta_x^2 + \Delta \theta_y^2 + \Delta \theta_z^2} \quad (3.34)$$

The above two equations are applied to update the quaternion, see Rogers [2003] for the detailed definition and properties of the quaternion for attitude computation.

$$\begin{bmatrix} q_1(t_{k+1}) \\ q_2(t_{k+1}) \\ q_3(t_{k+1}) \\ q_4(t_{k+1}) \end{bmatrix} = \begin{bmatrix} q_1(t_k) \\ q_2(t_k) \\ q_3(t_k) \\ q_4(t_k) \end{bmatrix} + 0.5 \begin{bmatrix} c & s\Delta\theta_z & -s\Delta\theta_y & s\Delta\theta_x \\ -s\Delta\theta_z & c & s\Delta\theta_x & s\Delta\theta_y \\ s\Delta\theta_y & -s\Delta\theta_x & c & s\Delta\theta_z \\ -s\Delta\theta_x & -s\Delta\theta_y & -s\Delta\theta_z & c \end{bmatrix} \begin{bmatrix} q_1(t_k) \\ q_2(t_k) \\ q_3(t_k) \\ q_4(t_k) \end{bmatrix} \quad (3.35)$$

The parameters s and c are given as follows:

$$s = 1 - \frac{\Delta\theta^2}{24} + \frac{\Delta\theta^4}{1920} + \dots \quad (3.36)$$

$$c = -\frac{\Delta\theta^2}{4} + \frac{\Delta\theta^4}{192} + \dots$$

The initial value of the quaternion is obtained after determining the initial DCM using Equation 2.4 with the computed initial attitudes during alignment process.

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} 0.25(C_{32} - C_{23}) / 0.5\sqrt{1 + C_{11} + C_{22} + C_{33}} \\ 0.25(C_{13} - C_{31}) / 0.5\sqrt{1 + C_{11} + C_{22} + C_{33}} \\ 0.25(C_{21} - C_{12}) / 0.5\sqrt{1 + C_{11} + C_{22} + C_{33}} \\ 0.5\sqrt{1 + C_{11} + C_{22} + C_{33}} \end{bmatrix} \quad (3.37)$$

The DCM is updated as follows:

$$C_b^l = \begin{bmatrix} (q_1^2 - q_2^2 - q_3^2 + q_4^2) & 2(q_1q_2 - q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 + q_3q_4) & (q_2^2 - q_1^2 - q_3^2 + q_4^2) & 2(q_2q_3 - q_1q_4) \\ 2(q_1q_3 - q_2q_4) & 2(q_2q_3 + q_1q_4) & (q_3^2 - q_1^2 - q_2^2 + q_4^2) \end{bmatrix} \quad (3.38)$$

The Euler angle of the attitudes, roll, pitch, and azimuth, are then given as follows

$$\theta = -\tan^{-1}\left(\frac{C_{31}}{\sqrt{1 - C_{31}^2}}\right) \quad (3.39)$$

$$\phi = a \tan 2(C_{32}, C_{33}) \quad (3.40)$$

$$\psi = a \tan 2(C_{21}, C_{11}) \quad (3.41)$$

Where $C_{i,j}$'s, $1 \leq i, j \leq 3$ are the $(i, j)^{th}$ elements of the DCM matrix and atan2 is a four quadrant inverse tangent function.

■ Velocity and Position Integration

The body frame velocity increment due to the specific force is transformed to the navigation frame using the following equation [Schwarz and Wei, 2000]:

$$\Delta v_f^l = C_b^l \begin{bmatrix} 1 & 0.5\Delta\theta_z & -0.5\Delta\theta_y \\ -0.5\Delta\theta_z & 1 & 0.5\Delta\theta_x \\ 0.5\Delta\theta_y & -0.5\Delta\theta_x & 1 \end{bmatrix} \Delta v_f^b \quad (3.42)$$

The first order sculling correction is applied utilizing Equation 3.42. The velocity increment is obtained by applying the gravity and the Coriolis correction:

$$\Delta v^l = \Delta v_f^l - (2w_{ie}^l + w_{el}^l) \times v^l \Delta t + g^l \Delta t \quad (3.43)$$

The velocity integration is then given as

$$v_{k+1}^l = v_k^l + \Delta v_{k+1}^l \quad (3.44)$$

The position integration is obtained using second order Runge-Kutta method:

$$r_{k+1}^l = r_k^l + 0.5 \begin{bmatrix} \frac{1}{(M+h)} & 0 & 0 \\ 0 & \frac{1}{(N+h)\cos\varphi} & 0 \\ 0 & 0 & -1 \end{bmatrix} (v_k^l + v_{k+1}^l) \Delta t \quad (3.45)$$

Figure (3.3) shows a schematic diagram for the implementation of the Local Level Frame (l-frame) INS mechanization.

3.2.4 INS Error Equations

The INS mechanization equations provide no information about the error of the system as they process the raw data from the IMU to estimate the navigation parameters. In order to improve the performance of INS, it is necessary to incorporate the sensor errors in the estimation process using one of the optimal estimation methods. This is commonly done through the utilization of the Kalman filter. However, given the non-linear nature of the INS mechanization, the perturbation analysis is applied to linearize the system differential equation for the error analysis. Given the complexity of the system at hand, the full derivation of the perturbed system is not given here. However the reader is advised to consult Savage [2000], Shin [2001], and El-Sheimy [2004a] for complete derivation of these equations.

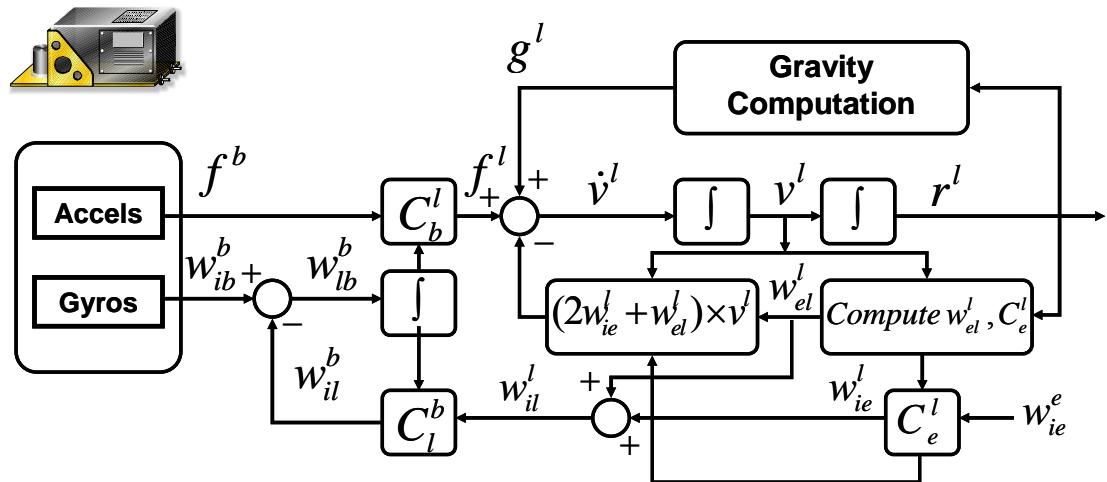


Figure 3.3: Local level frame mechanization

- Position error equations

The position error equations are function of position and velocity; therefore, the position error equations are obtained using the partial derivatives:

$$\delta \dot{r}^l = \begin{bmatrix} \delta \dot{\phi} \\ \delta \dot{\lambda} \\ \delta \dot{h} \end{bmatrix} = F_{rr} \begin{bmatrix} \delta \varphi \\ \delta \lambda \\ \delta h \end{bmatrix} + F_{rv} \begin{bmatrix} \delta v_N \\ \delta v_E \\ \delta v_D \end{bmatrix} \quad (3.46)$$

$$F_{rr} = \begin{bmatrix} 0 & 0 & \frac{-v_N}{(M+h)^2} \\ \frac{v_E \sin \varphi}{(N+h) \cos^2 \varphi} & 0 & \frac{-v_E}{(N+h)^2 \cos \varphi} \\ 0 & 0 & 0 \end{bmatrix} \quad (3.47)$$

$$F_{rv} = \begin{bmatrix} \frac{1}{(M+h)} & 0 & 0 \\ 0 & \frac{1}{(N+h) \cos \varphi} & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (3.48)$$

For analysis purposes, the first term of Equation 3.46 can be neglected as the velocity components are divided by the earth radius which results in small quantities. A simplified equation is given as follows [El-Sheimy, 2004a]:

$$\delta \dot{r}^l = \begin{bmatrix} \delta \dot{\phi} \\ \delta \dot{\lambda} \\ \delta \dot{h} \end{bmatrix} = \begin{bmatrix} \frac{1}{(M+h)} & 0 & 0 \\ 0 & \frac{1}{(N+h) \cos \varphi} & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \delta v_N \\ \delta v_E \\ \delta v_D \end{bmatrix} \quad (3.49)$$

■ Velocity error equations

The velocity error equations are given as follows:

$$\delta\dot{v}^l = \begin{bmatrix} \delta\dot{v}_N \\ \delta\dot{v}_E \\ \delta\dot{v}_D \end{bmatrix} = \begin{bmatrix} 0 & f_D & -f_E \\ -f_D & 0 & f_N \\ f_E & -f_N & 0 \end{bmatrix} \begin{bmatrix} \delta\phi \\ \delta\theta \\ \delta\psi \end{bmatrix} + C_b^l \begin{bmatrix} \delta f_x \\ \delta f_y \\ \delta f_z \end{bmatrix} + F_{vr} \begin{bmatrix} \delta\varphi \\ \delta\lambda \\ \delta h \end{bmatrix} + F_{vv} \begin{bmatrix} \delta v_N \\ \delta v_E \\ \delta v_D \end{bmatrix} \quad (3.50)$$

The gravity vector in the local level frame, g^l , is approximated by the normal gravity vector, $g^l = [0 \ 0 \ \gamma]^T$, and γ varies with altitude. A spherical Earth model is assumed and the simplified inverse square gravity model is given as [Rogers, 2003],

$$\gamma = \gamma_0 \left(\frac{R}{R+h} \right)^2 \quad (3.51)$$

Where γ_0 is the normal gravity at $h=0$, and $R = \sqrt{(MN)}$. Perturbing Equation 3.51 yields

$$\delta\gamma = 2 \left(\frac{\gamma}{R+h} \right) \delta h \quad (3.52)$$

$$F_{vr} = \begin{bmatrix} -2v_E w_e \cos\varphi - \frac{v_E^2}{(N+h)\cos^2\varphi} & 0 & \frac{-v_N v_D}{(M+h)^2} + \frac{v_E^2 \tan\varphi}{(N+h)^2} \\ 2w_e(v_N \cos\varphi - v_D \sin\varphi) + \frac{v_E v_N}{(N+h)\cos^2\varphi} & 0 & \frac{-v_E v_D}{(N+h)^2} - \frac{v_N v_E \tan\varphi}{(N+h)^2} \\ 2v_E w_e \sin\varphi & 0 & \frac{v_E^2}{(N+h)^2} + \frac{v_N^2}{(M+h)^2} - \frac{2\gamma}{(R+h)} \end{bmatrix} \quad (3.53)$$

$$F_{vv} = \begin{bmatrix} \frac{v_D}{(M+h)} & -2w_e \sin\varphi - 2\frac{v_E \tan\varphi}{(N+h)} & \frac{v_N}{(M+h)} \\ 2w_e \sin\varphi + \frac{v_E \tan\varphi}{(N+h)} & \frac{v_D + v_N \tan\varphi}{(N+h)} & 2w_e \cos\varphi + \frac{v_E}{(N+h)} \\ -2\frac{v_N}{(M+h)} & -2w_e \cos\varphi - 2\frac{v_E}{(N+h)} & 0 \end{bmatrix} \quad (3.54)$$

Similarly, the first two terms of the velocity error equations are the major components of the velocity errors as the velocity components are divided by the earth radius which result

in small quantities. The simplified velocity error equation is given as follows [El-Sheimy, 2004a]:

$$\delta\dot{v}^l = \begin{bmatrix} \delta\dot{v}_N \\ \delta\dot{v}_E \\ \delta\dot{v}_D \end{bmatrix} = \begin{bmatrix} 0 & f_D & -f_E \\ -f_D & 0 & f_N \\ f_E & -f_N & 0 \end{bmatrix} \begin{bmatrix} \delta\phi \\ \delta\theta \\ \delta\psi \end{bmatrix} + C_b^l \begin{bmatrix} \delta f_x \\ \delta f_y \\ \delta f_z \end{bmatrix} \quad (3.55)$$

In general, f_D is usually larger than f_N and f_E as it is close to the gravity value (9.8 m/sec^2) thus the coupling relationships between δV_N and $\delta\theta$ and between δV_E and $\delta\phi$ are strong. In contrast, the coupling relationship between either δV_N or δV_E and $\delta\psi$ is weak. The advantage of the strong coupling relationship between two errors is that if one of them is accurately estimated, then the other can be estimated accurately as well [El-Sheimy, 2004a].

■ Attitude error equations

$$\dot{\varepsilon}^l = \begin{bmatrix} \dot{\delta\phi} \\ \dot{\delta\theta} \\ \dot{\delta\psi} \end{bmatrix} = F_{ev} \begin{bmatrix} \delta v_N \\ \delta v_E \\ \delta v_D \end{bmatrix} - C_b \begin{bmatrix} \delta w_x \\ \delta w_y \\ \delta w_z \end{bmatrix} + F_\sigma \begin{bmatrix} \dot{\delta\phi} \\ \dot{\delta\theta} \\ \dot{\delta\psi} \end{bmatrix} - \begin{bmatrix} 0 & w_e \sin\varphi + \frac{v_e \tan\varphi}{N+h} & \frac{v_N}{M+h} \\ -w_e \sin\varphi - \frac{v_e \tan\varphi}{N+h} & 0 & -w_e \cos\varphi - \frac{v_e}{N+h} \\ \frac{v_N}{M+h} & w_e \cos\varphi + \frac{v_e}{N+h} & 0 \end{bmatrix} \begin{bmatrix} \delta\phi \\ \delta\theta \\ \delta\psi \end{bmatrix} \quad (3.56)$$

$$F_{ev} = \begin{bmatrix} 0 & \frac{1}{N+h} & 0 \\ \frac{-1}{M+h} & 0 & 0 \\ 0 & -\tan\varphi & 0 \end{bmatrix} \quad (3.57)$$

$$F_{er} = \begin{bmatrix} -w_e \sin \varphi & 0 & \frac{-v_E}{(N+h)^2} \\ 0 & 0 & \frac{v_N}{(M+h)^2} \\ -w_e \cos \varphi - \frac{v_E}{(N+h)\cos^2 \varphi} & 0 & \frac{v_E \tan \varphi}{(N+h)^2} \end{bmatrix} \quad (3.58)$$

Similarly, the first two terms of the attitude error equations are the major components, as a result, the simplified attitude error equations are given as follows:

$$\dot{\varepsilon}' = \begin{bmatrix} \dot{\delta\phi} \\ \dot{\delta\theta} \\ \dot{\delta\psi} \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{N+h} & 0 \\ \frac{-1}{M+h} & 0 & 0 \\ 0 & \frac{-\tan \varphi}{N+h} & 0 \end{bmatrix} \begin{bmatrix} \delta v_N \\ \delta v_E \\ \delta v_D \end{bmatrix} - C_b \begin{bmatrix} \delta w_x \\ \delta w_y \\ \delta w_z \end{bmatrix} \quad (3.59)$$

As a result, δv_N is coupled with $\delta\theta$ and δv_E is coupled with $\delta\phi$.

■ Accelerometer bias and gyro drift error equations

After removing the deterministic parts of the accelerometer bias (b) and gyro drift (d) before sending them as the inputs for INS mechanization, the remaining stochastic parts of the sensor errors are usually modeled as first order Gauss-Markov processes [Rogers,2003]:

$$\dot{b} = -\alpha b + \sqrt{2\alpha\sigma^2}$$

$$\delta f^b = \begin{bmatrix} \dot{\delta f}_x \\ \dot{\delta f}_y \\ \dot{\delta f}_z \end{bmatrix} = \begin{bmatrix} -\alpha_x & 0 & 0 \\ 0 & -\alpha_y & 0 \\ 0 & 0 & -\alpha_z \end{bmatrix} \begin{bmatrix} \delta f_x \\ \delta f_y \\ \delta f_z \end{bmatrix} + \begin{bmatrix} \sqrt{2\alpha_x\sigma_x^2} \\ \sqrt{2\alpha_y\sigma_y^2} \\ \sqrt{2\alpha_z\sigma_z^2} \end{bmatrix} w(t) \quad (3.60)$$

$$\dot{d} = -\beta d + \sqrt{2\beta\sigma^2}$$

$$\delta \dot{w}^b = \begin{bmatrix} \delta \dot{w}_x \\ \delta \dot{w}_y \\ \delta \dot{w}_z \end{bmatrix} = \begin{bmatrix} -\beta_x & 0 & 0 \\ 0 & -\beta_y & 0 \\ 0 & 0 & -\beta_z \end{bmatrix} \begin{bmatrix} \delta w_x \\ \delta w_y \\ \delta w_z \end{bmatrix} + \begin{bmatrix} \sqrt{2\beta_x \sigma_x^2} \\ \sqrt{2\beta_y \sigma_y^2} \\ \sqrt{2\beta_z \sigma_z^2} \end{bmatrix} w(t) \quad (3.61)$$

Where α and β are the correlation times and σ^2 is the variance of the process.

See Rogers [2003] for the determination of correlation time. Consequently, a continuous 1st order differential equation with 15 error states can be obtained by augmenting Equations 3.46, 3.50, 3.56, 3.60 and 3.61

$$\dot{x} = Fx + Gw \quad (3.62)$$

Where F is the dynamics matrix, x is the error state vector, G is a noise coefficient matrix and w is system noise:

$$F = \begin{bmatrix} F_{rr} & F_{rv} & 0 & 0 & 0 \\ F_{rv} & F_{vv} & (f^l \times) & C_b^l & 0 \\ F_{er} & F_{ev} & -(w_{il}^l \times) & 0 & -C_b^l \\ 0 & 0 & 0 & -\alpha & 0 \\ 0 & 0 & 0 & 0 & -\beta \end{bmatrix} \quad x = \begin{bmatrix} \delta r^l \\ \delta v^l \\ e^l \\ \delta f^b \\ \delta w^b \end{bmatrix} \quad G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \sqrt{2\alpha\sigma^2} \\ \sqrt{2\beta\sigma^2} \end{bmatrix} \quad (3.63)$$

3.2.5 INS Error Characteristics

- East channel error model

According to El-Sheimy [2004a], the east channel error characteristics can be obtained by differentiating the velocity/attitude Equations 3.55 and 3.59 and by substituting into the new equation by the values of $\delta\dot{\phi}/\delta\dot{v}_E$ from Equations 3.59 and 3.55. Two non-homogenous linear second order differential equations are then obtained as follows:

$$\delta\ddot{v}_E + \frac{g}{N+h} \delta v_E = g \delta w_N \quad (3.64)$$

$$\delta\ddot{\phi} + \frac{g}{N+h}\delta\phi = \frac{1}{N+h}\delta f_E \quad (3.65)$$

The solution of these two 2nd order differential equations yields the δv_E and $\delta\phi$ that are oscillating over time with a frequency (1/5000Hz), f_s , called the Schuler frequency, with a period of 84.4 minutes, and is given as follows:

$$f_s = \frac{1}{2\pi} \sqrt{\frac{g}{N+h}} \approx \sqrt{\frac{g}{R}} \quad (3.66)$$

As a result, δv_E and $\delta\phi$ becomes bounded over time.

■ North channel error model

Similarly, the north channel error model can be obtained by differentiating the velocity component $\delta\dot{v}_N / \delta\dot{\theta}$ from Equation 3.55/3.59 and by substituting $\delta\dot{\theta} / \delta\dot{v}_N$ from Equation 3.59/3.55. Two non-homogenous linear second order differential equations are obtained:

$$\delta\ddot{v}_N + \frac{g}{M+h}\delta v_N = -g\delta w_E \quad (3.67)$$

$$\delta\ddot{\theta} + \frac{g}{M+h}\delta\theta = \frac{-1}{M+h}\delta f_N \quad (3.68)$$

Similarly, the solution of the above two 2nd order differential equation yields δv_N and $\delta\theta$ that are oscillating over time with a Schuler frequency as well. Consequently, δv_N and $\delta\theta$ becomes bounded over time. However, It can be observed from Equation 3.50 that δv_N and δv_E also contain non-stationary components that depend on the azimuth error ($-f_E\delta\psi$ and $f_N\delta\psi$). In fact, the major part of the horizontal velocity error is bounded in time due to the Schuler effect. However, the non-stationary components

change with time dependent azimuth drift and the velocity of the vehicle. Consequently, the azimuth error $\delta\psi$ plays an important role in determining the long term positioning accuracy ($\iint -f_E \delta\psi$ and $\iint f_N \delta\psi$). In fact, the effect of $\delta\psi$ becomes critical at high velocities [El-Sheimy, 2004a and Noureldin, 2002].

According to Titterton and Weston [1997], various error sources including the sensor errors (accelerometer bias and gyro drift), initial navigation errors (initial position, velocity and misalignment), and vehicle dynamics contribute to the characteristics of the errors in the derived navigation parameters. In the horizontal channels, position, velocity and attitude errors grow systematically but are bounded by the Schuler effect.

On the contrary, position error is quadratically dependent on time while the velocity is linearly dependent on time in the vertical channel. Also, they both drift away as the observation time elapses. Because of interaction between the azimuth misalignments (vertical attitude error) with the horizontal velocity, the azimuth error grows with time but is bounded by the Schuler effect. Its linear dependence on time causes it to drift.

Table 3.7 illustrates the different error sources of a nav-grade IMU and their magnitudes. In addition, Figure 3.4 simulates the impact of those errors on the position error, velocity error, and attitude error in a horizontal and vertical channel respectively. The equations applied for the simulation can be found in Britting [1971].

Table 3.7: Error sources and parameters of nav-grade IMU

Initial Position error	p_0	0 .1(m)	Gyro Drift	D	$0.001^\circ / h$
Initial Velocity error	v_0	0 .001(m/s)	Gyro Drift (vertical)	d_{Au}	$0.001^\circ / h$
Initial Altitude error	h_0	0.25	Accelerometer Bias	B	$100 \mu g$
Initial azimuth error	ε_{A0}	0.1(deg)	Misalignment error	ε_0	0.01(deg)

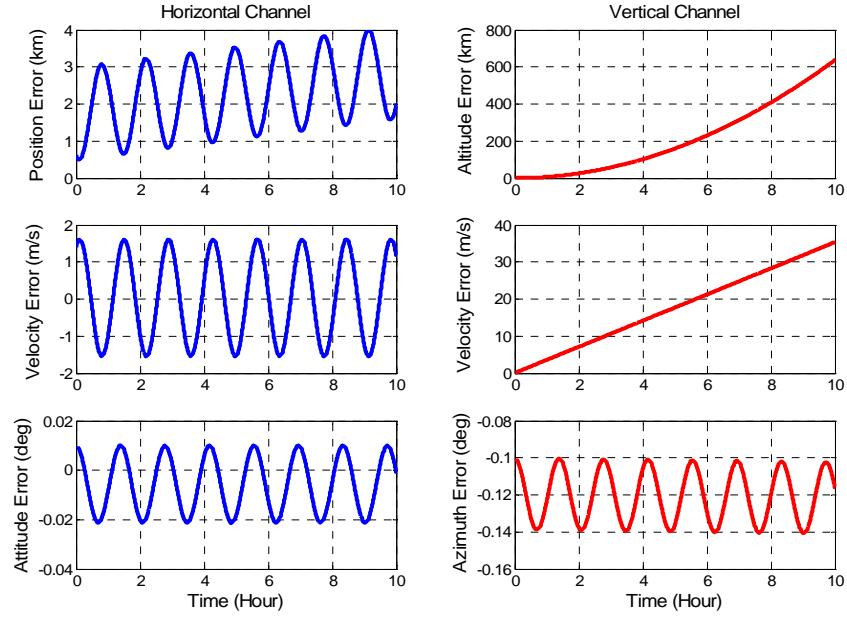


Figure 3.4: Simulation results of a navigation grade IMU

3.2.6 Future Development of INS Based Positioning Technology-MEMS IMU

Inertial technology has evolved over the last fifty years from mechanical gyros using zero velocity updates and star trackers for error control to ring laser and fiberoptic gyros updated with GPS observations of various types. Nowadays, IMUs are maturing to the state that low-cost commercialization is becoming feasible. At the same time, computer processor power is sufficient to allow both GPS and inertial data to be integrated on a single board. Micro-Electro-Mechanical Systems (MEMS) is a process technology used to create tiny integrated devices or systems that combine mechanical and electrical components [Sukkarieh, 2000]. They are fabricated using integrated circuit (IC) batch processing techniques and can range in size from a few micrometers to millimeters. These devices (or systems) have the ability to sense, control and actuate on the micro scale, and generate effects on the macro scale. In the most general form, MEMS consist of mechanical microstructures, microsensors, micro actuators and microelectronics, all integrated onto the same silicon chip [Gad El-Hok, 2001].

MEMS technology provides significant reductions in terms of cost, size, weight, and volume and power consumption [Schwarz and El-Sheimy, 1999]. In the near future, MEMS will be able to offer tactical grade and navigation grade performance sensors for a variety of applications. For land vehicular navigation applications, the integration of GNSS and tactical MEMS IMUs would be a reasonable option when the cost of MEMS tactical grade IMU becomes reasonable for general user. The accuracy and the stability of current MEMS sensors do not meet the specification of tactical grade IMU. However, with the latest development of MEMS technologies, it is anticipated to reach tactical grade MEMS based IMUs by the year 2010[El-Sheimy, 2000a]. Consequently, with the full constellation of GPS and Galileo, the positioning of autonomous mode can be expected to reach the 1 meter level. Such positioning accuracy is good enough for most of the land vehicular navigation applications. Meanwhile, the low cost MEMS tactical grade IMUs can be applied to bridge the system during GPS outages. In other words, an affordable, accurate, safety oriented and GNSS/MEMS augmented land vehicular navigation system can be expected to be available in the near future.

3.3 INS/GPS Integration and Kalman Filter

The Kalman filter has been widely adapted as a standard optimal estimation tool for INS/GPS integration applications. The derivation of an error model to be applied in the Kalman filter starts with the construction of full scale true error models, whose order is decided based on the complexity of the problem [Bar-Shalom, et al., 2001]. In general, the dynamical model is based on an error model that includes three position errors, three velocity errors and three attitude errors, augmented by some dominant sensor errors, such as accelerometer bias and gyro drifts. In general, models containing 15~21 states are often applied for high quality strapdown INS. The integration with GPS data reduces only the long term errors. Therefore, the remaining error budget is mainly affected by short-term error sources [Skaloud, 1999].

3.3.1 Fundamentals of the Kalman Filter

The Kalman filter is named after Rudolph E. Kalman, who in 1960 published his famous paper describing a recursive solution to the discrete-data linear filtering problem [Kalman, 1960]. The Kalman Filter is essentially a set of mathematical equations that implement a predictor-corrector type estimator that is optimal in the sense that it minimizes the estimated error covariance—when some presumed conditions are met. Since the time of its introduction, the Kalman Filter has been the subject of extensive research and application, particularly in the area of autonomous or assisted navigation.

■ Discrete Kalman Filter

The Kalman Filter addresses the general problem of trying to estimate the state of a discrete-time controlled process that is governed by the linear dynamic model [Brown and Huang, 1992]:

$$x_k = F_{k,k-1}x_{k-1} + G_{k-1}w_{k-1} \quad (3.69)$$

With a measurement that is

$$z_k = H_k x_k + v_k \quad (3.70)$$

where,

- $F_{k,k-1}$ is the state transition matrix
- G_{k-1} is the system noise coefficient matrix
- H_k is the design matrix
- z is the measurement vector
- x is the error state vector

Three assumptions are made to define the Kalman filter. The random variables w_k and v_k represent the system and measurement noise, respectively [Brown and Huang, 1992].

1. The system noise and measurement noise are uncorrelated and zero mean random processes. That is:

$$E[w_k v_j^T] = 0 \text{ for all } j \text{ and } k \quad (3.71)$$

$$w_k \sim N(0, Q_k)$$

$$v_k \sim N(0, R_k)$$

and the covariance matrices are

$$E[w_k w_j^T] = \begin{cases} Q_k, & j = k \\ 0, & j \neq k \end{cases} \quad (3.72)$$

$$E[v_k v_j^T] = \begin{cases} R_k, & j = k \\ 0, & j \neq k \end{cases} \quad (3.73)$$

In reality, the system noise covariance Q , and measurement noise covariance R matrices might change with each time step. However, they are assumed to be constant [Mohamed, 1999].

2. The initial system state vector x_0 is uncorrelated to both system noise and measurement noise

$$E[x_0 w_k^T] = 0; \quad E[x_0 v_k^T] = 0 \quad (3.74)$$

3. The initial mean value of the system state and the covariance matrix of the initial system state are known

$$\bar{x}_0 = E[x_0]; \quad P_0 = E[(x_0 - \bar{x}_0) (x_0 - \bar{x}_0)^T] \quad (3.75)$$

There are two steps in Kalman filtering. The first step is the prediction by the system model [Mohamed, 1999],

$$\hat{x}_k^- = F_{k,k-1} \hat{x}_{k-1} \quad (3.76)$$

$$P_k^- = F_{k,k-1} P_{k-1} F_{k,k-1}^T + G_{k-1} Q_{k-1} G_{k-1}^T \quad (3.77)$$

and the second step is the measurement update of the system model

$$\text{Kalman gain matrix: } K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \quad (3.78)$$

$$\text{Error covariance update: } P_k = (I - K_k H_k) P_k^- \quad (3.79)$$

$$\text{State prediction update: } \hat{x}_k = \hat{x}_k^- + K_k (z_k - H\hat{x}_k^-) \quad (3.80)$$

In the previous equation, $(z_k - H\hat{x}_k^-)$ is known as the measurement innovation or the residual.

■ Kalman filter for INS/GPS Integration

The detailed implementation of the Kalman filter can be found in [Schwarz and Wei, 2000]. An example of a 15 states Kalman filter is given as follows: The error states include three position parameters, three velocity parameters, three attitude parameters, three accelerometer bias parameters and three gyro drift parameters.

$$X = [\delta\varphi \quad \delta\lambda \quad \delta h \quad \delta v_N \quad \delta v_E \quad \delta v_D \quad \delta\varphi \quad \delta\theta \quad \delta\psi \quad \delta f_x \quad \delta f_y \quad \delta f_z \quad \delta w_x \quad \delta w_y \quad \delta w_z] \quad (3.81)$$

The state transition matrix $F_{k,k-1}$ can be obtained using the dynamics matrix, F, as follows:

$$F_{k,k-1} = \exp(F\Delta t) \approx I + F\Delta t \quad (3.82)$$

The measurement equation that uses GPS velocity and position as measurements update is given as follows [El-Sheimy, 2004a]:

$$z_k = \begin{bmatrix} r_{INS}^l - r_{GPS}^l \\ v_{INS}^l - v_{GPS}^l \end{bmatrix} = \begin{bmatrix} \varphi_{INS} - \varphi_{GPS} \\ \lambda_{INS} - \lambda_{GPS} \\ h_{INS} - h_{GPS} \\ v_{INS}^l - v_{GPS}^l \end{bmatrix} \quad H_k = \begin{bmatrix} I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix} \quad (3.83)$$

The measurement equation that uses zero velocity update (ZUPT) as measurements is given as follows [El-Sheimy, 2004a]:

$$z_k = \begin{bmatrix} V_N - 0 \\ V_E - 0 \\ V_D - 0 \end{bmatrix} = \begin{bmatrix} v_{INS}^l - 0 \end{bmatrix} \quad H_k = \begin{bmatrix} \boxed{0_{3 \times 3}} & \boxed{I_{3 \times 3}} & \boxed{0_{3 \times 3}} & \boxed{0_{3 \times 3}} & \boxed{0_{3 \times 3}} \end{bmatrix} \quad (3.84)$$

The degree of the complexity of the INS/GPS integration approach is purely application dependent [Kayton and Fried 1996]. In general, the INS/GPS integration architectures can be implemented in different ways. The main distinction is made according to the coupling status between INS and GPS (uncoupled, loosely coupled, tightly coupled, and ultra-tightly coupled), and whether the estimated sensor errors are fed back to correct the measurements (feedforward and feedback).

A brief comparison of commonly used INS/GPS integration architectures is given in Table 3.10, which is modified from Skaloud [1999]. See Jekeli , [2001], Kreye, et al., [2002], Shin [2001], Alban et al., [2003], Petovello [2003], and Scherzinger [2004] for a completed review.

3.3.2 Limitations of INS/GPS Integration Using the Kalman Filter

As mentioned previously, the Kalman filter depends on a set of measurements and a proper dynamics model to provide optimal estimates of the states [Skaloud, 1999]. Besides the quality of the measurements, the final quality of the filter states relies on the quality of the dynamic model. If the filter is exposed to input data that does not fit the model, it will not result in reliable estimates. Obviously, the model presentation depends on the initial knowledge and on the real process taking place in the system. The limiting factors of the Kalman filter based INS/GPS integration are described below in more details:

Table 3.10: A brief comparison of commonly used INS/GPS integration architectures
 (Modified from Skaloud, 1999)

Implementation	Advantages	Disadvantages
Uncoupled	<ul style="list-style-type: none"> <i>Integration algorithm is a simple decision algorithm</i> 	<ul style="list-style-type: none"> <i>System accuracy decrease rapidly in the absence of GPS</i>
Loosely coupled (decentralized)	<ul style="list-style-type: none"> <i>INS and GPS Kalman are implemented separately</i> <i>The size of individual Kalman filter is small</i> <i>Flexible, modular combination</i> <i>Suitable for parallel processing, reliability</i> <i>Less computation complexity</i> 	<ul style="list-style-type: none"> <i>Sub-optimal performance</i> <i>Four satellite required for a stable solution</i> <i>INS data is not used for ambiguity estimation</i>
Tightly coupled (centralized)	<ul style="list-style-type: none"> <i>One error state model</i> <i>Optimal solution, accuracy</i> <i>GPS measurements can be used with less than 4 satellites</i> <i>Faster ambiguity estimation</i> 	<ul style="list-style-type: none"> <i>Large size of error state model</i> <i>More complex processing</i>
Ultra-tightly coupled (deep)	<ul style="list-style-type: none"> <i>INS information (velocity) is added into tracking loops of the receiver</i> <i>Reduce the dynamic stress of the receiver, the thermal noise influence, and the loop bandwidth of the receiver</i> <i>Anti-jamming</i> <i>Faster lost GPS signal re-acquisition</i> 	<ul style="list-style-type: none"> <i>Realization of such coupling principle requires special hardware component and accesses to the firmware of receiver</i> <i>Implementation can only be conducted by hardware manufacturers</i>
Feedforward (open loop)	<ul style="list-style-type: none"> <i>Kalman filter may be run external to INS, suitable for platform INS</i> <i>Used when only navigation solution from INS available</i> 	<ul style="list-style-type: none"> <i>The INS mechanization is unaware of the existence of the external data</i> <i>The INS mechanization can experience unbounded error growth,</i> <i>Non-linear error model due to large second-order effect</i> <i>Extended Kalman filter needed</i>
Feedback (closed loop)	<ul style="list-style-type: none"> <i>Inertial system errors, linear model is sufficient</i> <i>Suitable for software level</i> 	<ul style="list-style-type: none"> <i>More complex processing</i> <i>Blunder in GPS may affect INS performance</i>

- The impact of INS short term error

Figure (3.5) illustrates a conceptual plot of the frequency spectrum of the errors in the measurements of inertial sensors. Notice the division into long term errors (low frequency) and short-term errors (high frequency). Figure (3.6) shows how each of these errors is reduced by the integration process. The long term errors are reduced by updating the filter with external measurements (e.g. the GPS position and velocity). The short term errors are reduced by the smoothing that is done by the numerical integration process of the INS mechanization. However, Figure (3.6) shows the benefits of the integration are band-limited as the lower boarder of the INS/DGPS error spectrum is mainly determined by the biases in GPS observations while the upper boarder is mainly determined by short term inertial sensor errors.

Consequently, the remaining GPS biases in the GPS navigation solution, such as ionospheric delay, tropospheric delay and multipath are responsible for the very long term errors in Figures (3.5 and 3.6). Due to the consequence of sampling theory, the utilization of DGPS data to reduce the short term INS error is not effective as the sampling rate of DGPS measurement (1Hz) is much lower than that of INS. Consequently, the long term INS errors that are reduced by the integration process with GPS are usually more significant than the short term errors.

In general, the long term errors usually include accelerometer bias and gyro drifts that are usually modeled as error states. Therefore, the impact of these long term errors for long periods of time can be limited with the external aiding. On the contrary, the remaining short term error in inertial sensors remains and is responsible for a certain amount of the error accumulation during GPS outage periods. As a result, this research presents a novel denoising technique, named the Cascade Denoising, which is developed utilizing a second generation wavelet denoising algorithm. The ultimate goal is to improve the quality of INS raw measurements by reducing the impact of INS short term errors; thus the positioning accuracy for an INS/GPS integrated vehicular navigation during GPS outages can be improved.

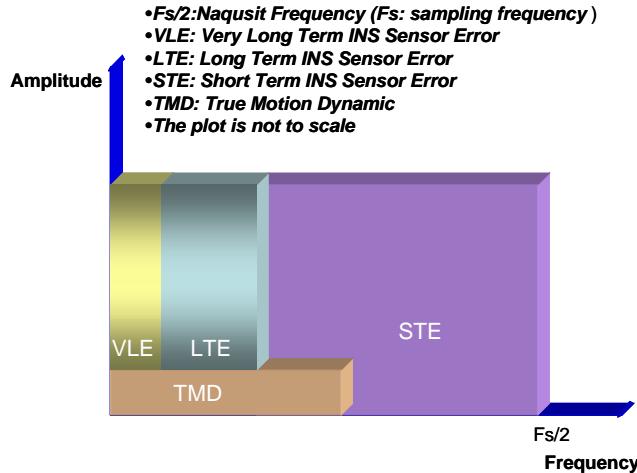


Figure 3.5: Conceptual plot of the spectrum of INS sensor errors (After Skaloud, 1999)

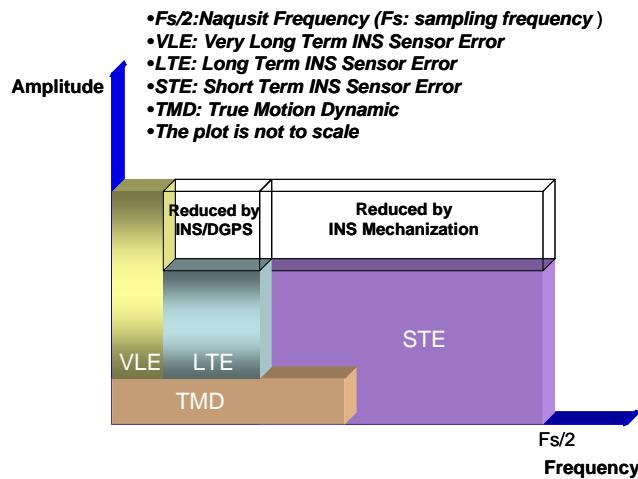


Figure 3.6: Conceptual plot of the spectrum of INS sensor errors (After INS/DGPS)
(After Skaloud, 1999)

To give an example of the impact of noise on the derived navigation parameters, Figure (3.7) illustrates the impact of X-Gyro noise on these parameters for a vehicle in static mode. Considering X-Gyro noise is the only error source of this IMU attached to a vehicle pointed to the north and assuming three IMU measurements with different noise levels (i.e. standard deviation is 0.01, 0.05 and $0.5 \text{ deg}/\sqrt{\text{hr}}$, respectively). The values represent the noise level of the gyro of navigation, tactical and MEMS IMUs, respectively. The position errors, velocity errors and attitude errors can be obtained by comparing the results generated by INS mechanization with the true value. Figure (3.7)

clearly indicates that the smaller the level of noise the better the accuracy of the navigation solutions.

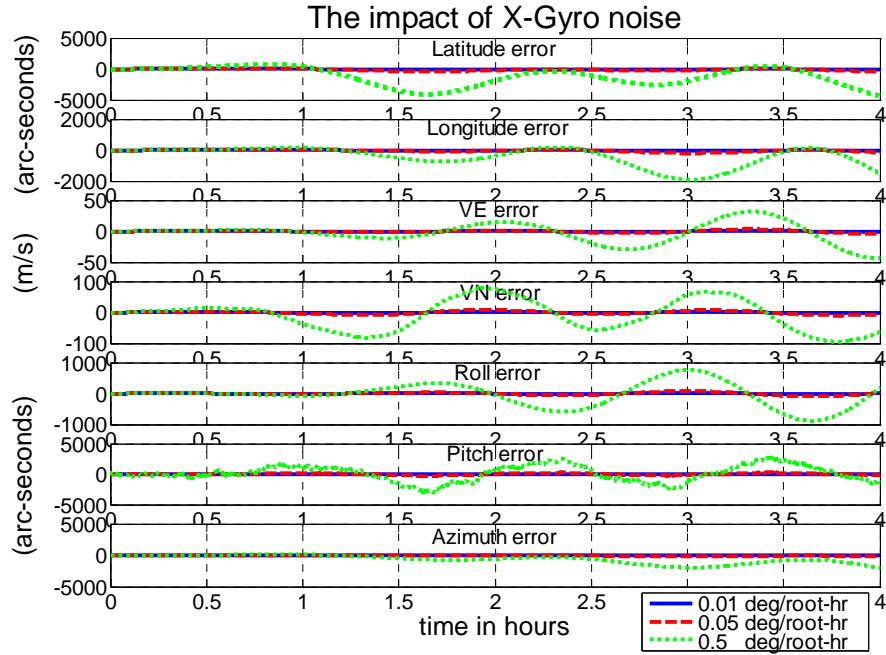


Figure 3.7: The impact of X-Gyro noise

Therefore, the denoising algorithm developed in the next chapter is expected to improve the overall accuracy of the navigation solutions during GPS outages.

■ Model dependency

Generally speaking, the development a model to be used in the Kalman filter starts with the construction of a full scaled “true-error model”, whose order is then reduced based on the prior knowledge and the insight gained into the physics of the problem, covariance analysis, and simulation [Salychev et al, 2000].

Typically, the dynamics model is based on an error model for three position errors, three velocity errors, and three attitude errors in an INS (the system error states). These errors are also augmented by some sensor error states such as accelerometer biases and gyroscope drifts, which are modeled as stochastic processes (i.e., 1st Gauss Markov

process or random walk) [Rogers,2003]. In fact, there are several random errors associated with each inertial sensor. For instance, noise contributions in typical optical gyroscope systems may include white noise, correlated random noise, bias instability and angle random walk (IEEE Std. # 647-1995). Therefore, it is usually difficult to set a certain stochastic model for each inertial sensor that works efficiently at all environments and reflects the long term behavior of the sensor errors. Hence a model-less navigation algorithm that can perform the self-following of the vehicle under all-conditions is required.

- Prior knowledge dependency

As mentioned previously, some initial knowledge is required to start a Kalman filter, such as the state transition matrix ($F_{k,k-1}$), the measurements design matrix (H_k), the noise coefficient matrix (G_{k-1}), the system noise covariance matrix (Q) and the measurements noise covariance matrix (R). Among them, the Q and R matrices are the most important factors for the quality of the Kalman filter estimation for an INS/GPS integrated system. Theoretically, the optimal Q and R matrices can guarantee the optimality of the estimation; however, it is not easy to obtain such information. In fact, tuning the Q and R matrices can be time consuming and it requires experience and background in both systems. Consequently, the requirement of human intervention for Q/R tuning is very high. In other words, the tuning process can be regarded as a special form of learning as it is usually done by an expert and needs time to obtain the optimal solution. Consequently, a new navigation algorithm that can reduce the level of human intervention and is capable of learning by itself to adapt the latest dynamic model is preferred.

- Sensor dependency

The need to re-design algorithms based on the Kalman filter (i.e., states) to operate adequately and efficiently on every new platform (application) or different systems (e.g. switch from navigation grade IMU to tactical grade IMUs) can be very costly. In

addition, the Q and R matrices tuning is heavily system dependent. For example, it is impossible to use the sets of parameters that are designated for navigation grade IMUs for estimation utilizing tactical grade IMUs. This is known as IMU-grade dependence. Further more, it has been shown that even with the same accuracy grade, the Q and R matrices required for different IMUs that are provided by different manufacturers might be different. This is known as sensor dependence. As a result, a new navigation algorithm that is adaptable and can reduce the level of sensor and IMU-grade dependence is highly desirable

- Linearization dependency

INS/GPS integration for land vehicular navigation is non-linear in nature. However, since the principle of Kalman filtering is to estimate a linear dynamical model using a recursive algorithm along with certain stochastic information, the linearization of INS or GPS dynamics model is required [Brown and Huang, 1992]. However, the linearization process is usually a 1st order approximation process that results in deviations between the assumed “true error model” and the real “true error model”. As a result, a new navigation algorithm that is nonlinear in nature and can reduce the impact of linearization is preferred.

Consequently, alternative algorithms for INS/GPS integration in land vehicular navigation are presented in Chapters 5 and 6. The ultimate goal of this research is to develop a conceptual intelligent navigator that can overcome most of previous stated limitations. The core component of such a navigator are: 1) the development of various AIAs based INS/GPS integration architectures that utilize neural networks as the data fusion algorithm for INS and GPS to generate navigation knowledge, 2) the implementation of Navigation Information database as the “brain” of the intelligent navigator to store navigation knowledge and 3) the development of an efficient learning scheme that enables the learning ability and accumulates navigation knowledge of such an intelligent navigator.

CHAPTER 4

CASCADE DENOISING OF IMU SIGNALS

As mentioned in the previous chapter, the INS errors can be divided into long term errors (low frequency) and short term errors (high frequency). The long term errors usually include accelerometer bias and gyro drifts that are usually modeled stochastically as state variables inside the INS error model. Therefore, the impact of those long term errors for long periods of time can be limited with external aiding. On the contrary, the remaining short term errors (including white noise) in inertial sensors are responsible for a certain amount of the error accumulation during GPS outage periods. Therefore, if these short term errors can be reduced or removed from the kinematic IMU raw measurements to improve the quality of the measurements, the overall positioning accuracy of INS navigation can be improved.

The wavelet denoising algorithm can be applied to separate the low frequency and high frequency error components of IMU signals [Skaloud, 1999, Burton et al., 1999, and Nassar, 2004]. However, for land vehicular navigation applications, the concern is how to remove INS short term errors including high frequency noises, vibrations and other disturbances and improve the accuracy of INS navigation without jeopardizing the true motion dynamic component of the vehicle. It requires the prior knowledge of the bandwidth of true motion dynamic of typical land vehicles and the spectrum characteristic of the wavelet denoising algorithm. Thus, the positioning errors after applying denoised kinematic IMU measurements can be expected to be improved if the true motion dynamic content can be well preserved and the short term INS sensor error can be reduced during the denoising operation.

In this chapter, the continuous wavelet transform (CWT) is presented first to gain some appreciation regarding the benefits of utilizing such a technique. Consequently, the Discrete Wavelet Transform (DWT) and the Multiresolution Analysis (MRA) are given to provide the spectrum perspective of MRA. The bandwidth of true motion dynamic is then investigated through the spectrum perspective of kinematic IMU signals. In addition,

the limitations of existing wavelet denoising are given through the spectrum analysis. Finally, the cascade denoising algorithm is implemented. Discussion of the cascade denoising implementation and its performance is tested using kinematic IMU measurements.

4.1 The Continuous Wavelet Transform

A Fourier analysis of a signal, $f(t)$, extracts information in the frequencies contained in $f(t)$. The standard Fourier transform is given [Mallat, 2001]:

$$F(w) = \int_{-\infty}^{+\infty} f(t) e^{-iwt} dt \quad (4.1)$$

Where w is frequency and t is time. Fourier analysis is capable of providing frequency information; however, it is unable to provide information concerning time-localization. Therefore, windowing the signal $f(t)$ is a step towards obtaining such information. The signal is first restricted to an interval (with smooth edges) by multiplying it by a fixed window function, prior to carrying out a Fourier analysis of the product. Repeating the process with shifted versions of the window function allows localized frequency information throughout the signal to be obtained. Since the window-width is the same for all frequencies, the amount of localization remains constant for different frequencies. Mathematically, the windowed Fourier transform, also known as short time Fourier transform (STFT), can be written as [Mallat, 2001]:

$$T^{STFT}(u, t) = \int_{-\infty}^{+\infty} f(t) g(t-u) e^{-iwt} dt \quad (4.2)$$

Where $g(t)$ is the windowing function and the u is the translation factor(window size). The Wavelet transform can be defined for different classes of functions. The intention in this transformation is to address some of the shortcomings of the STFT. Instead of fixing the time and the frequency resolutions Δt and Δf , one can let both resolutions vary in the time-frequency plane in order to obtain a multiresolution analysis. This variation can be

carried out without violating the Heisenberg inequality [Strang, 1993]. In this case, the time resolution must increase as frequency increases and the frequency resolution must increase as frequency decreases. This can be obtained by fixing the ratio of Δf over f to be equal to a constant c [Mallat, 2001]:

$$\text{Time-Bandwidth product} = \Delta t * \Delta f \geq \frac{1}{4\pi} \quad (4.3)$$

$$\frac{\Delta f}{f} = c \quad (4.4)$$

In terms of the filter bank terminology, the analysis filter bank consists of band-pass filters with constant relative bandwidth (so-called constant-Q analysis). The frequency responses of the analysis filters in the filter bank are regularly spaced in a logarithmic scale [Strang, 1993]. These filters are naturally distributed into octaves. Consequently, the time resolution becomes arbitrarily good at high frequencies, while the frequency resolution becomes arbitrarily good at low frequencies [Olivier and Vetterli, 1991]. Two very close short bursts can eventually be separated if one goes to higher analysis frequencies in order to increase time resolution. The wavelet analysis, as explained, works best if the signal is composed of high frequency components of short duration plus low frequency components of long duration. The continuous wavelet transform (CWT) is based on such ideas. In this case, all impulse responses of the analysis filters in the filter bank are defined as scaled (i.e., stretched or compressed) versions of the same prototype ψ , known as the mother wavelet [Daubechies 1992]. For example, for a scale factor of a , the filter impulse response becomes

$$\psi_a(t) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{t}{a}\right) \quad (4.5)$$

The function is normalized by the constant $\frac{1}{\sqrt{|a|}}$ [Daubechies 1992]. In the case of the

continuous wavelet transform, the translation parameter b and the dilation parameter a change continuously. In other words, the transformation utilizes the following equation

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-b}{a}\right) \text{ with } a, b \in R, a \neq 0 \quad (4.6)$$

Consequently, the continuous wavelet transform of a function f is defined by

$$w(a, b) = \langle f, \psi_{a,b} \rangle \quad (4.7)$$

Suppose that the wavelet ψ satisfies the admissibility condition, where $\hat{\psi}(w)$ is its Fourier Transform [Daubechies 1992];

$$C_\psi = \int_{-\infty}^{+\infty} \frac{|\hat{\psi}(w)|^2}{w} dw < \infty \quad (4.8)$$

Then the CWT $w(a, b)$ is invertible on its range, and an inverse transform is given by the relation

$$f(t) = \frac{1}{C_\psi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} w(a, b) \psi_{a,b}(t) \frac{dadb}{a^2} \quad (4.9)$$

From the admissibility condition, $\hat{\psi}(0)$ has to be 0 [Daubechies 1996], and in particular, ψ has to oscillate. This, together with decay property, has given the name wavelet or “small wave” (French: ondelette) [Daubechies 1996]. In applications, it is of interest to find that the inverse transform does not make use of $w(a, b)$ over the whole range of a and b . Transforms exist that only use positive values of a or even only discrete values for a . The common choice is to use a dyadic grid. In general, the fewer values of a and b one wants to use, the more restrictive the condition on the wavelet becomes. The CWT results in a very general wavelet. The transform that only uses the dyadic values of a and b is called Discrete Wavelet Transform (DWT) which is given in section 4.2.

4.1.1 Spectrum Perspective of CWT

In order to compare the performance of the Fourier Transform (FT), the STFT and the CWT, sine waves with three sampling rates (Fs) (200Hz, 100Hz and 50Hz) that contain various simulated dynamic frequency ranging from 1Hz to 90 Hz with 2Hz gap were

generated. The reason for choosing these sampling rates is that they represent the common sampling rate of IMUs. A hamming window function was assigned for the STFT. “DB5” of Daubechies wavelet family with scale ranging from 1 to 6 was utilized. The results are given in Figure (4.1).

As shown in the Figure (4.1), the FT transforms the signal into the frequency domain, but losses time information. When looking into a FT of a signal, it is impossible to tell when or where a particular event took place. The lack of time-localization information makes the FT unsuitable for analyzing non-stationary signals which have breakdown points or discontinuities in time domain [Aboufadel and Schlicker, 1999]. On the contrary, the STFT represents a compromise between the time- and frequency-based views of a signal. It provides some information about both when and at what frequencies a signal event occurred. The precision of such information is determined by the size of the window.

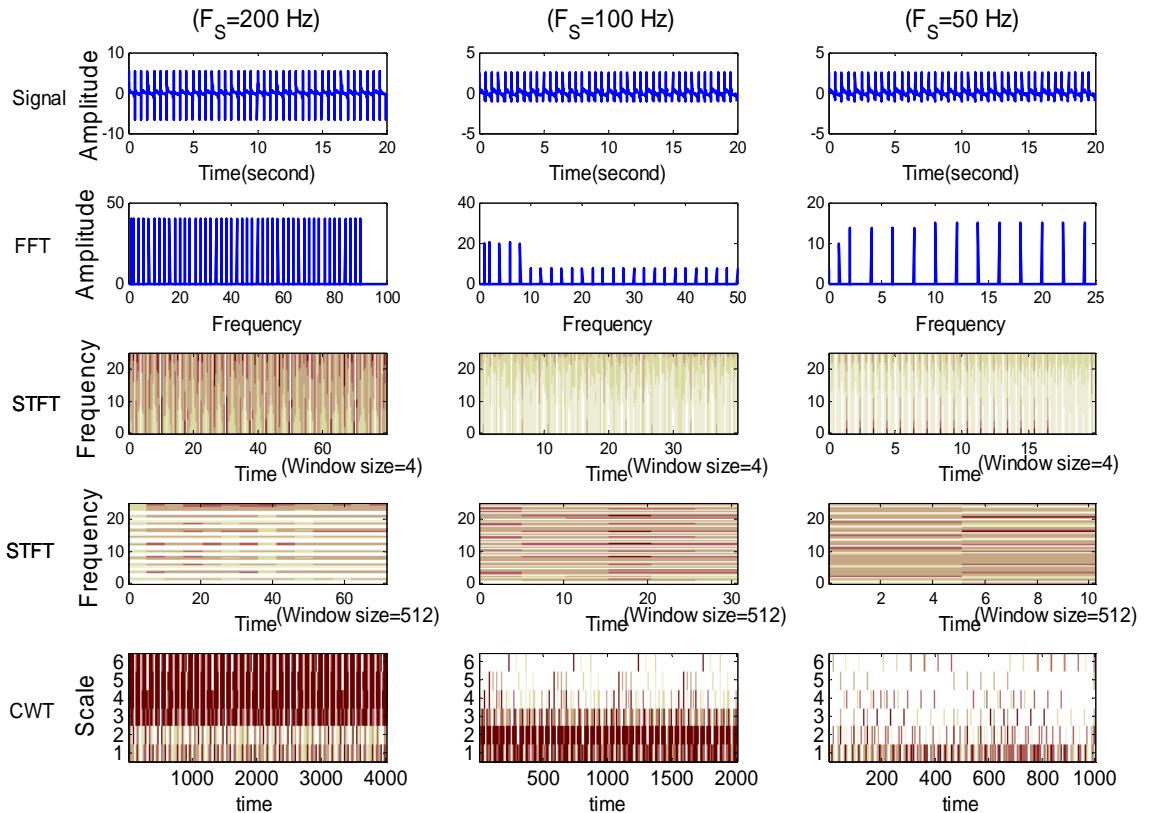


Figure 4.1: Comparisons between FT, STFT and CWT

While the STFT compromises between time and frequency information, the drawback is that once a particular size for the time window is chosen, that window is the same for all frequencies. It can be seen for Figure (4.1) that the narrow window results in good time resolution but poor frequency resolution. On the contrary, the wide window results in poor time resolution but good frequency resolution. Many signals require a more flexible approach, one where the window size can be varied to determine more accurately either time or frequency.

As illustrated in Figure (4.1), the CWT provides multiresolution analysis (MRA) to overcome the limitations of the FT and STFT. Unlike the STFT which has a constant resolution at all times and frequencies, the WT has a good time and poor frequency resolution at high frequencies, and good frequency and poor time resolution at low frequencies. The lower scales (higher frequencies) have better scale resolution, narrower in scale, which means that it is less ambiguous what the exact value of the scale is, which corresponds to poorer frequency resolution [Strang, 1993]. Similarly, higher scales have poorer scale frequency resolution, wider support in scale, which means it is more ambiguous what the exact value of the scale is, which corresponds to better frequency resolution of lower frequencies [Strang, 1993]. Consequently, MRA provides a superior way to analyze non-stationary signals in comparison to FT and STFT. Figure (4.2) illustrates two wavelets with different scales.

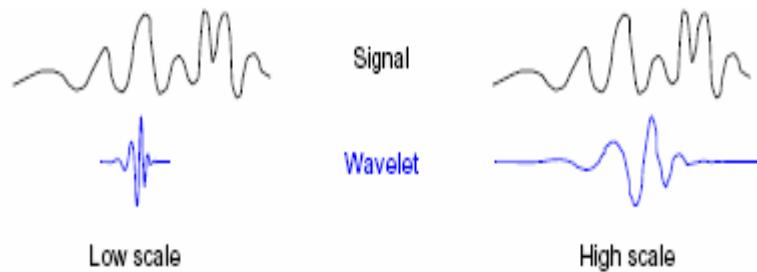


Figure 4.2: Wavelets with different scales (After Misiti et al., 1997)

Figure (4.3) illustrates the spectrum of the CWT coefficients on a dyadic grid with respect to different sampling rates.

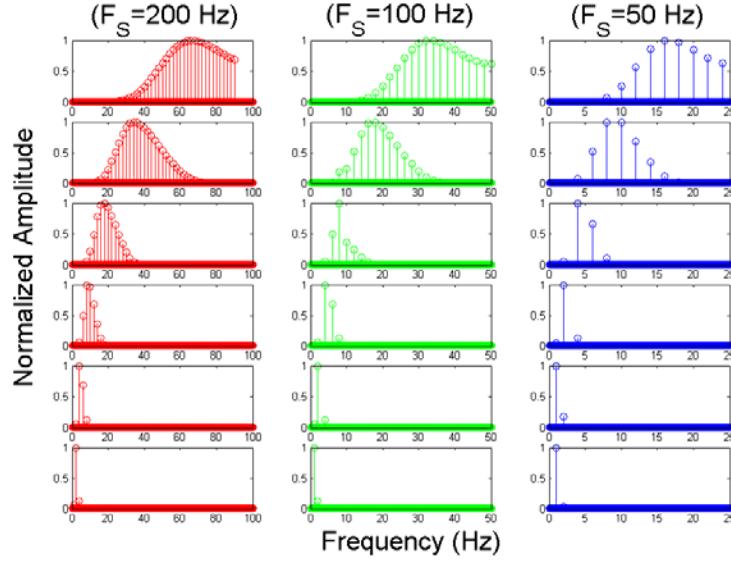


Figure 4.3: Spectrum of CWT

The CWT can be regarded as a series of band pass filter (highpass filters) whose centre frequency varies with the scale and sampling rates, as indicated in Figure (4.3). There is a correspondence between wavelet scales and frequency as revealed by wavelet analysis. Low scale corresponds to a compressed wavelet, which can retrieve rapidly changing details (high frequency components). On the other hand, high scale corresponds to a stretched wavelet that can retrieve slowly changing parts, approximations, of the signal (low frequency components) [Olivier and Vetterli, 1991].

4.1.2 The linkage between FFT and CWT

For developing the cascading denoising algorithm, investigating the relationship between the scale of the CWT and its corresponding frequency content in the spectrum domain was necessary. The traditional way to solve this mystery is through the use of pseudo-frequency corresponding to a scale. A way to do it is to compute the centre frequency F_c of the wavelet and use the following relationship [Misiti et al., 1997].

$$F_p = \frac{F_c}{F_s \times a} \quad (4.10)$$

Where a is a scale, F_s is the sampling frequency. F_c is the centre frequency of a wavelet in Hz. F_p is the pseudo-frequency corresponding to the scale a , in Hz. According to Sparto et al, [1999], the centre frequency is obtained through associating a given wavelet with a purely periodic signal of frequency F_c . In other word, the frequency maximizing the FFT of the wavelet modulus is F_c [Misiti et al., 1997].

Figure (4.4) illustrates the example of Daubechies wavelets of order 2 to 7. The centre frequency of the wavelet is decided by varying the centre frequency of the periodic signal until the correlation between the peak of the wavelet function and the wave form of the periodic signal reach maximum [Sparto et al,1999]. Extracting the centre frequencies of those bandpass filters at dyadic decomposition levels ($2^n, n=1, 2, \dots$) ($F_{c_{CWT}}$) that are generated through the Fourier transform of the CWT coefficients, as shown in the Figure (4.3), and comparing with the pseudo frequency (F_p) that are obtained through Equation 4.10 and Table 4.1 using scale as an index, the relationship between $F_{c_{CWT}}$ and F_p , named as associated coefficient, ω , is given as:

$$\omega('wavelet name') = \frac{F_{c_{CWT}}('wavelet name')}{F_p('wavelet name')} \quad (4.11)$$

Table 4.1 illustrates the ω of the Daubechies wavelet family with order 1 to 10 obtained. The proposed associated coefficients are independent of the sampling rates chosen in this analysis (i.e., 200 Hz, 100 Hz and 50 Hz), but are related to the choice of wavelets. Thus, the associated frequency F_A related to certain choice of wavelet and scale is given:

$$F_A('wavelet name') = F_p('wavelet name') * \omega('wavelet name') \quad (4.12)$$

The examples of associated frequency with the utilization of ‘DB3’ and ‘DB5’ are given in Appendix A.1. Associated frequency can be regarded as the centre frequency of the CWT at each scale. As a result, it provides the frequency information of the corresponding scale. In fact, associated frequency can be applied as an explicit index of

the frequency content when using the CWT to analyze kinematic IMU signals. Table A.1 and Table A.2 also confirm that the CWT has poor frequency resolution at high frequencies, and good frequency resolution at low frequencies.

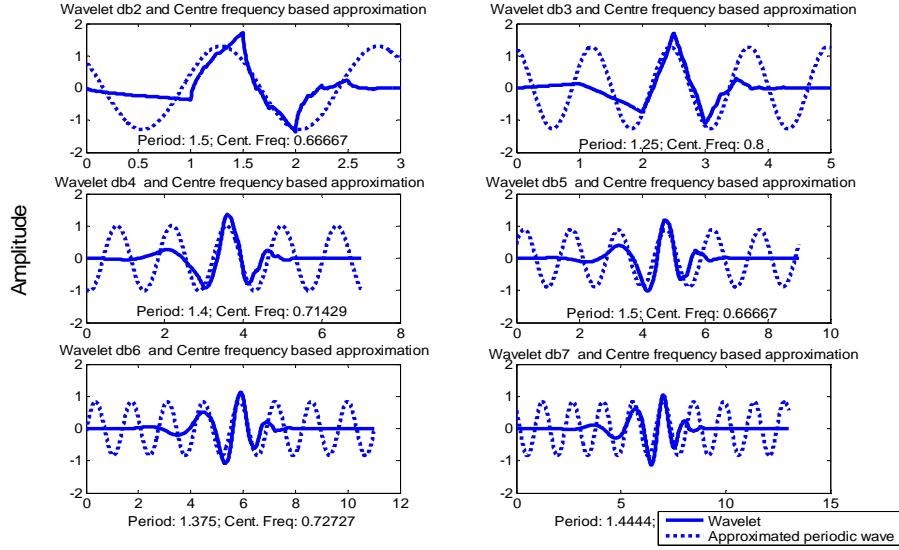


Figure 4.4: Examples of the centre frequencies of Daubechies wavelet family

Table 4.1: Associated coefficients of Daubechies wavelet family

Wavelet	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8	DB9	DB10
F_c	0.996	0.67	0.8	0.71	0.67	0.73	0.69	0.67	0.71	0.68
ω	0.64	0.96	0.8	0.90	0.96	0.88	0.93	0.96	0.91	0.94

4.2 Discrete Wavelet Transform

The key difference in a discrete wavelet analysis is that the scale parameter a and translation parameter b in Equation 4.6 are no longer continuous, but are instead integers. Indeed, in the majority of cases, the choice of a and b is limited to the following discrete set [Daubechies 1992]:

$$a = 2^j, \quad b = k2^j = ka \quad (4.13)$$

Where j and k are integers. The indices a and b in $\psi_{a,b}$ are replaced by j and k , respectively. Thus Equation 4.6 can be expressed as follows [Daubechies 1992]:

$$\psi_{j,k}(t) = 2^{-j/2} \psi(2^{-j}t - k) \quad (4.14)$$

Reconstruction of a signal from its DWT is possible provided the wavelet satisfies certain conditions (these are discussed in the Appendix A). The reconstruction formula is analogous to Equation 4.9 [Daubechies 1992]:

$$f(t) = \sum_{j,k} \langle f, \psi_{j,k} \rangle \psi_{j,k}(t) \quad (4.15)$$

The most common and general approach to constructing a wavelet basis is to use a Multiresolution Analysis (MRA) [Mallat, 1989a and 1989b]. For an overview see Appendix A. In signal processing, such ideas are implemented as subband filtering, or quadrature mirror filtering. The decomposition step consists of a lowpass (h) and a highpass (g) filter followed by downsampling ($\downarrow 2$) (i.e., retaining only the even index samples) [Strang and Nguyen, 1997]. Figure (4.5) illustrates an example of a three level decomposition and the conceptual spectrums of the approximation coefficients ($cA_i, i = 1, 2, 3$) and the detail coefficients ($cD_i, i = 1, 2, 3$).

In contrast, the reconstruction consists of upsampling ($\uparrow 2$) (i.e., putting a zero between every two samples) followed by filtering (\tilde{h} and \tilde{g}) and addition, as shown in Figure (4-6). More detail can be found in [Strang and Nguyen, 1997].

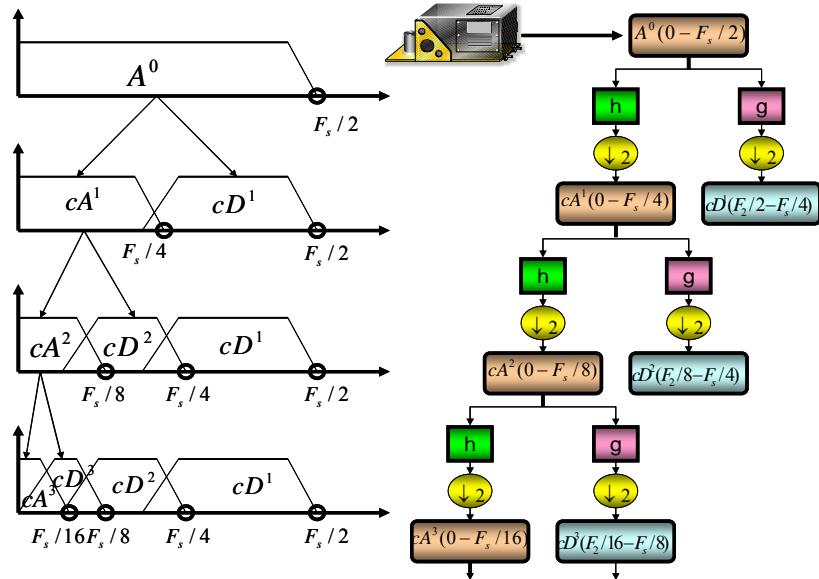


Figure 4.5: Three level wavelet decomposition and their conceptual spectrums

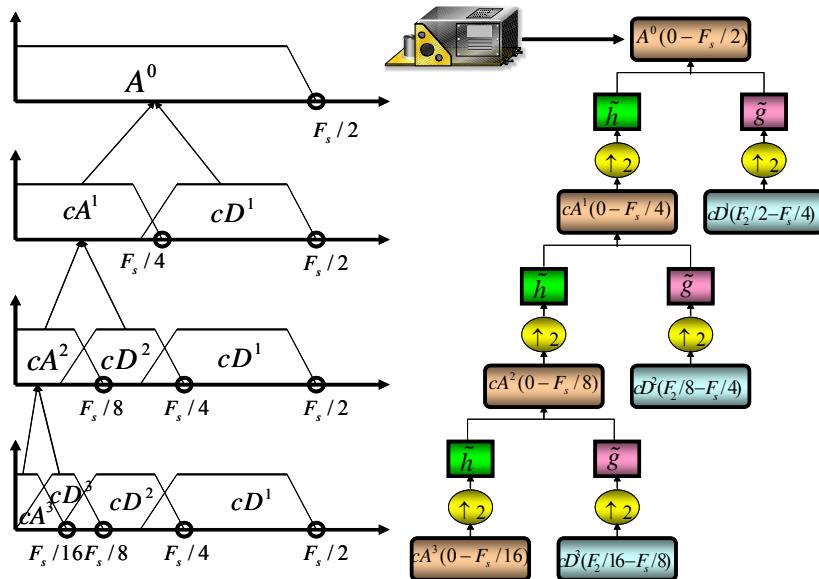


Figure 4.6: Three levels wavelet reconstruction

4.2.1 Spectrum Perspective of DWT

Figure (4.7) depicts the relationships between decomposition level and frequency content

using the simulated signal applied in Figure (4.1), where blue solid circles and red dashed squares represent the spectrums of the approximation and detail signals respectively. The distinction between the approximation signals/coefficients and detail signals is given in Appendix A. The approximation signals obtained through MRA at each decomposition level can be regarded as the signals generated by a series of wavelet based lowpass filters that are corresponding to different decomposition levels [Sparto, 1999], therefore, the stop bands of those lowpass filters can be extracted from Figure (4.7).

Consequently, Table 4.2 illustrates the relationship between the decomposition level and the residual frequency content in the approximation signal (i.e., red solid circle). It provides extremely important information for denoising IMU kinematic signals, which is given in detail in subsequent sections. Similar relationships between the sampling rate, decomposition level and stop bands can be founded in [Sparto, 1999].

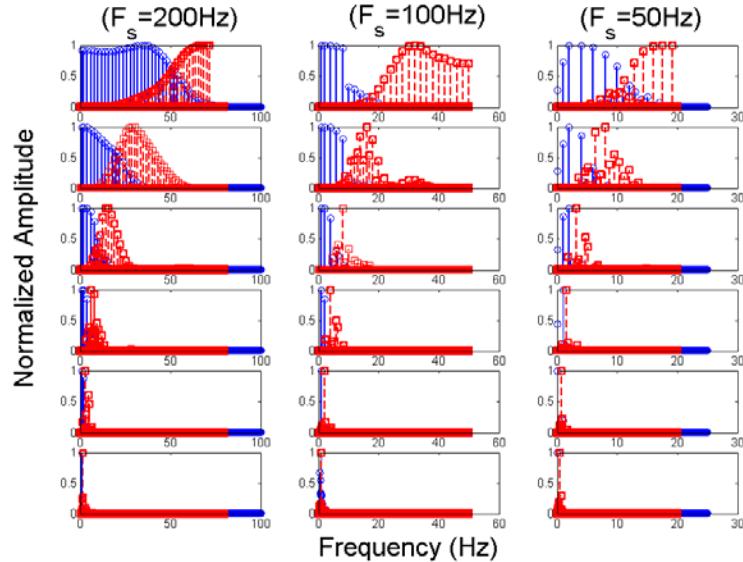


Figure 4.7: Spectrum of DWT (Approximation/Detail signals)

Table 4.2: Relationship between stop bands and decomposition level

	Fs=200 Hz Stop band (Hz)	Fs=100 Hz Stop band (Hz)	Fs=50 Hz Stop band (Hz)
DL=1	64	32	16
DL=2	32	16	8
DL=3	16	8	4
DL=4	8	4	2
DL=5	4	2	1
DL=6	2	1	0.5

4.3 Spectrum Analysis of IMU Raw Measurements

According to Czompo [1990], the spectrum domain of raw kinematic IMU signals contains the impact of the following factors: 1) true motion dynamic, 2) dither disturbances, 3) vibration disturbance and 4) white noise. Consequently, the ultimate goal of any pre-filtering development is to remove the impact of vibration disturbance and noise without jeopardizing the content of the true motion dynamic. The spectrum analysis of IMU raw measurements under different scenarios, such as alignment (engine off), ZUPT (engine on), short dynamic (engine on) and general navigation condition (motion dynamic + alignment + ZUPT), is very complicated.

It is desirable to evaluate the impact of these factors individually; thus the bandwidth of true motion dynamics that are monitored by each of the inertial sensors can be isolated from the massive spectrum plots. Figure (4.8) to Figure (4.10) illustrate the spectrum of the IMU raw measurements that were collected from three different IMUs, CIMU® (Honeywell, navigation grade, Fs=200Hz), LN200® (Litton, tactical grade, Fs=200Hz) and Crossbow® AHRS400-CC (Crossbow, MEMS, Fs=110 Hz). See chapter 7 for the details of this field test. Be aware that these IMUs were operated at the same time on the same vehicle with the same trajectory. Consequently, they experienced the same motion dynamic and vibration disturbance. The detail specifications of those IMUs are given in Appendix C.

The Y axis of Figure (4.8), Figure (4.9) and Figure (4.10) represents the normalized amplitude of the spectrum generated by each sensor in different scenarios. Be aware that

the scale of the Y-axis might have been adjusted in some scenarios to provide better visualization. Since the data length in each scenario is different and its maximum amplitude in the spectrum domain varies with the data length, consequently, comparing those normalized amplitudes directly between different scenarios cannot provide much useful information.

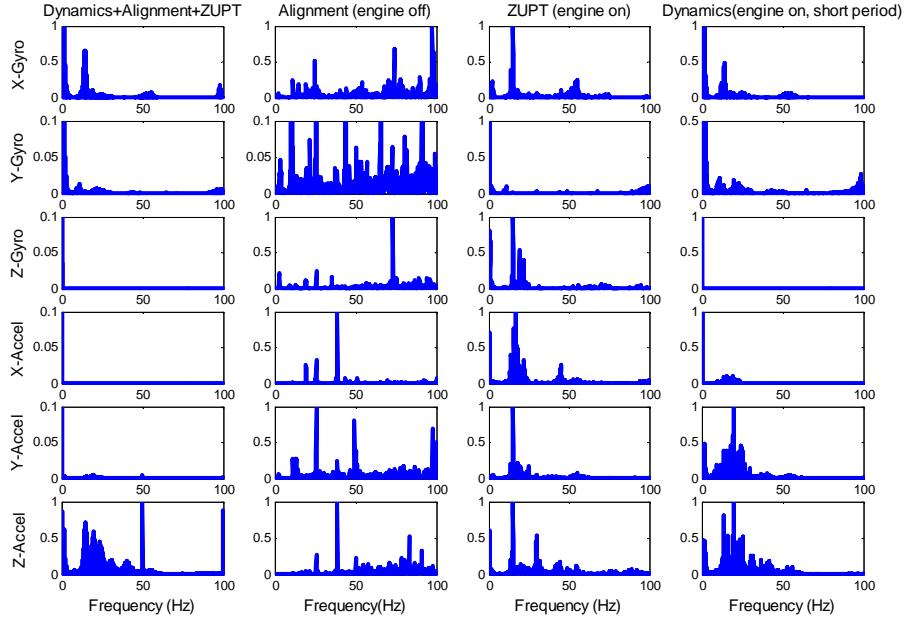


Figure 4.8: Spectrum of CIMU

The impact of dither disturbances, vibration disturbance, and noise are discussed as Table 4.3. The alignment period in Figure (4.8) to Figure (4.10) was carried out in static mode with the engine off. As a result, the impact of other vibrations can be reduced. The peaks, known as dither spikes, are illustrated in Figure (4.8) during alignment corresponding to several aliased frequencies as CIMU has three ring laser gyros. Furthermore, the engine vibration disturbance can be identified clearly by comparing the spectrum between alignment (engine off) and ZUPT (engine on) in Figure (4.8), Figure (4.9) and Figure (4.10).

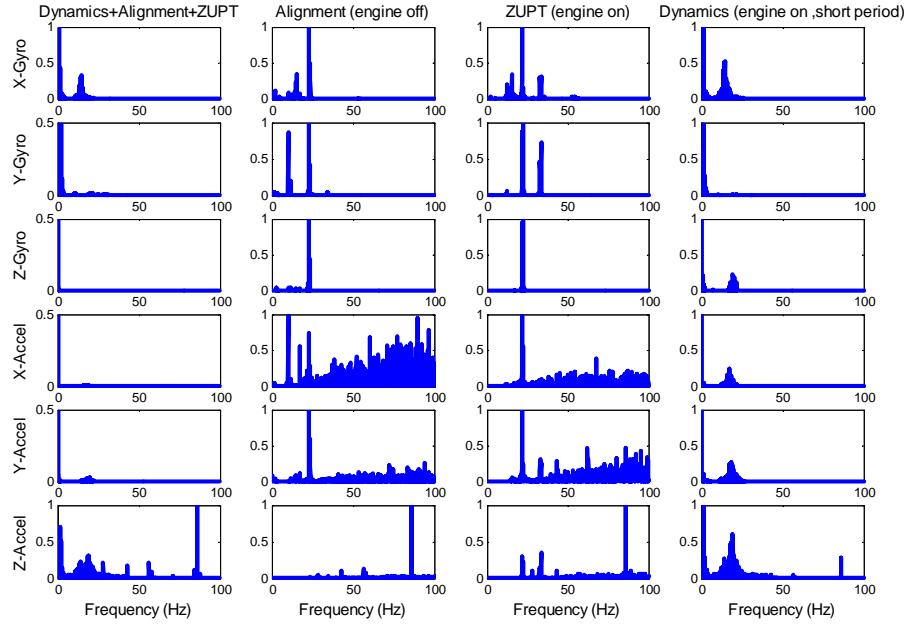


Figure 4.9: Spectrum of LN200

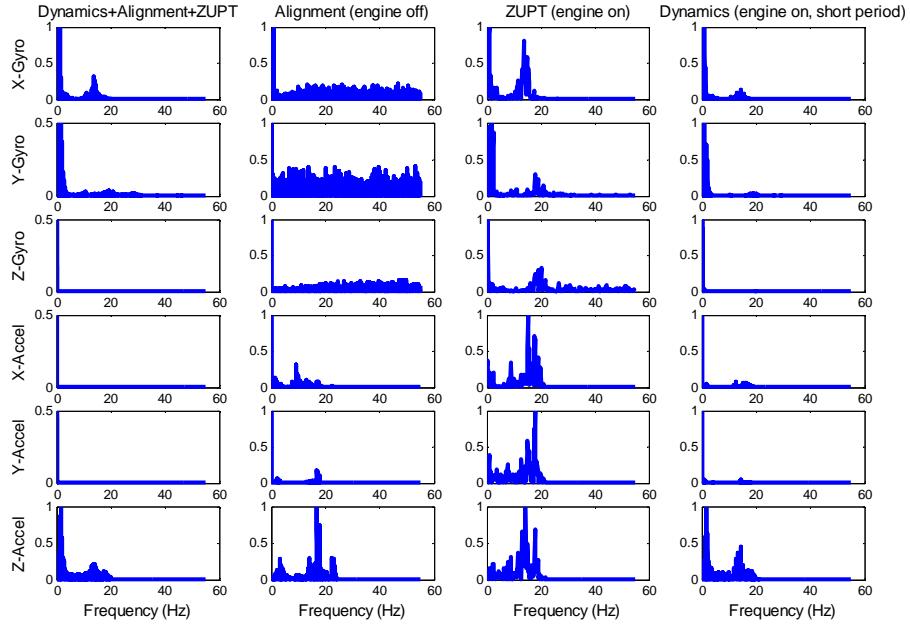


Figure 4.10: Spectrum of Crossbow AHRS400-CC

By conceptually removing the impact of dither spike and engine vibration in Figure (4.8), Figure (4.9) and Figure (4.10), all the IMUs indicate similar characteristics of true motion dynamic. Therefore, the true motion dynamic is IMU independent regardless of the

quality of the IMUs used. General speaking, the bandwidths of the true motion dynamic sensed by different sensors can be grouped into two clusters, as indicated in Table 4.3. The spectrum characteristics of X- Accelerometer, Y- Accelerometer and Z-Gyro meet the nature of land vehicle motion. Table 4.4 illustrates the bandwidths of true motion dynamic sensed by different sensors.

Table 4.3: The impact of different factors

Source	Characteristics
Dither spike	<ul style="list-style-type: none"> <i>Dithering motion of ring-laser gyros causes vibrations in the whole sensor block</i> <i>The dither spike can be removed through the use of special frequency filtering [Czompo, 1990]</i>
Engine Vibration	<ul style="list-style-type: none"> <i>The engine vibration disturbance can be identified in the 10-25 Hz frequency range. Similar results concerning the engine vibration of an airplane can be found in [Czompo, 1990].</i>
White noise	<ul style="list-style-type: none"> <i>As indicated in Figure (4.8) to Figure (4.10), the noise level increased when the accuracy level of IMU decreased,</i> <i>In principle, noise can be reduced by applying proper denoising algorithm which is given in later sections</i>
True motion dynamic	<ul style="list-style-type: none"> <i>X-Gyro, Y-Gyro and Z-Accelerometer; due to the road irregularities (i.e., bumps); the motion frequencies for those sensor mainly appear in the 0-6 Hz band. Thus, the upper bound is set at 8 Hz for denoising purpose.</i> <i>X- Accelerometer, Y- Accelerometer and Z-Gyro; very low frequency components are dominant. The frequency band of acceleration and deceleration motion in X or Y direction mainly appear in the 0-1Hz band. This indicates a much smoother translation motion along the trajectory, similarly, the frequency band of azimuth or heading change mainly appear in the 0-1Hz band as well. This indicates a much smoother rotation about the vertical body axis. Thus, the upper bound is set at 2 Hz for denoising purposes.</i>

Table 4.4: Bandwidth of true motion dynamic

Sensor	Gyro			Accelerometer		
	W x	W y	W z	A x	A y	A z
Bandwidth	<8Hz	<8Hz	<2Hz	<2Hz	<2Hz	<8Hz

4.4 Existing Denoising Algorithms

In general, a denoising filter is designed to suppress or remove the noise component and other disturbances; thus, the quality of the signals can be improved for further analysis. The denoising process may take place in the time domain or in a transform domain. The transform domain can be the time-frequency domain via the Fourier transform or the time-scale domain via the wavelet transform.

4.4.1 First Generation Denoising Algorithm

The DWT often has a “concentrating” effect on a signal. That is, signals with characteristic time frequency behavior are succinctly represented in the DWT domain. The DWT of a coherent signal contains a small amount of large amplitude coefficients [Jansen, 2001]. This suggests two similar applications at which the DWT excels; compression and denoising. Data compression can be archived by assuming that only the large amplitude coefficients are necessary for characterizing the signal, and hence the others are discarded. Denoising uses a similar strategy, wavelet shrinkage, although the philosophy is that small coefficients are not only unnecessary but undesirable [Jansen, 2001].

Just as a coherent signal is represented by a small number of large amplitude coefficients, an incoherent signal is represented by a large amount of small coefficients. Additive white noise is, by definition, incoherent. By keeping only large coefficients, thresholding the DWT, a white-noise corrupted signal can be largely recovered [Jansen, 2001]. Classical linear filtering in the frequency domain relies on a frequency separation between desired signal and unwanted noise. In some cases, the signal and the noise are overlapped on certain frequency ranges. Thus, this method of denoising is ineffective [Donoho, 1992]. Thresholding in the DWT domain relies on an amplitude separation between the coefficients of the desired signal and the unwanted signal. As long as the desired signal is relatively coherent in the DWT domain and the noise is incoherent, thresholding can effectively denoise the signal [Donoho, 1992]. It is worth noting that thresholding in the DWT domain is a nonlinear operation.

The basic principle of first generation denoising is to perform thresholding on the DWT of the noisy signal, and then take the inverse DWT of the thresholded coefficients to obtain the denoised signal.

Donoho [1992] proposed the wavelet shrinkage schemes of denoising:

- (1) Suppose $x(n)$ is the original signal of length N, $y(n) = x(n) + e(n)$, where $y(n)$ is corrupted by $e(n) \sim N(0,1)$. Find the DWT of $y(n)$ which is called $Y_{j,k}(n)$
- (2) Perform hard thresholding on $Y_{j,k}(n)$, which can be described with,

$$\hat{X}_{j,k} = \begin{cases} Y_{j,k}, & \text{if } |Y_{j,k}| \geq \delta \\ 0, & \text{if } |Y_{j,k}| < \delta \end{cases} \quad (4.16)$$

Or soft thresholding,

$$\hat{X}_{j,k} = \begin{cases} \text{sgn}(Y_{j,k})(|Y_{j,k}| - \delta), & \text{if } |Y_{j,k}| \geq \delta \\ 0, & \text{if } |Y_{j,k}| < \delta \end{cases} \quad (4.17)$$

Where δ can be decided empirically using static IMU data or utilizing a more sophisticated algorithm, Stein's Unbiased Risk Estimation of Risk (SURE), which is applied in this research, See Donoho and Johnstone [1995] and Jansen [2001] for more details about various strategies that can be applied to decide δ . The original idea of denoising using wavelet shrinkage is to remove white noise, $e(n) \sim N(0,1)$, however, it has been reported that it is capable of removing non-white noise using level-dependent δ [Jansen, 2001]

- (3) Take the inverse DWT of $\hat{X}_{j,k}$ to recover the denoised signal $\hat{X}(L)$.

The denoising procedure for IMU raw measurements is shown in Figure (4.11).

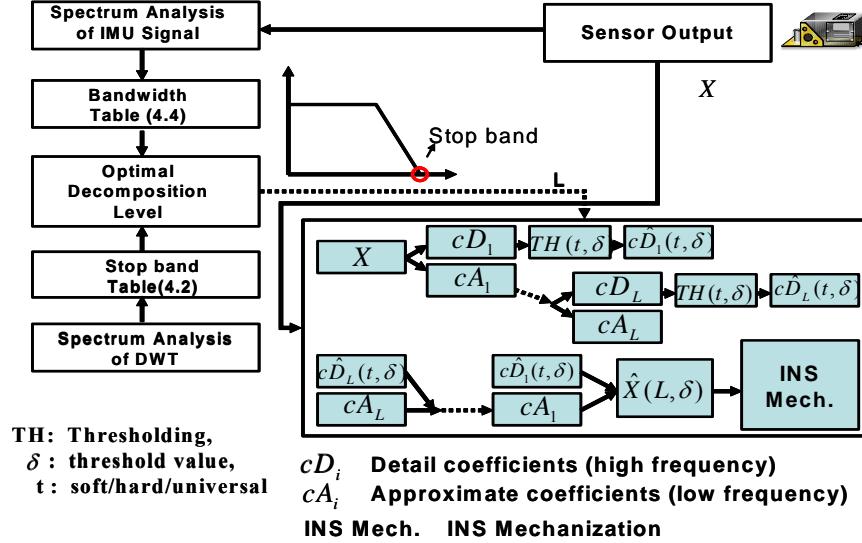


Figure 4.11: IMU signals denoising

Table 4.5: Optimal decomposition level for kinematic IMU measurements

	Fs=200 Hz	Fs=100 Hz	Fs=50 Hz
	BW/L	BW/L	BW/L
Wx	8/4	8/3	8/2
Wy	8/4	8/3	8/2
Wz	2/6	2/5	2/4
Ax	2/6	2/5	2/4
Ay	2/6	2/5	2/4
Az	8/4	8/3	8/2

The optimal decomposition level (L) varies with the bandwidth of true motion dynamic in each sensor. Comparing Table 4.2 and Table 4.4, an optimal decomposition level (L) can be decided for each sensor, as shown in Table 4.5..The characteristics of hard and soft thresholding is given as follows

- Hard-thresholding is a “keep or kill” procedure in which small amplitude coefficients are removed while the others are left unchanged.
- Soft-thresholding is a continuous procedure in which coefficients above the threshold are shrunk in absolute value. The amount of shrinking equals the threshold values, hence the input –output plot becomes continuous.

In this research, a new thresholding technique, universal thresholding, is developed. The proposed thresholding technique is expected to be able to preserve the advantages of soft and hard thresholding. It can be determined from Equation 4.18 shown below that when $n=1$, the universal thresholding is equivalent to soft thresholding and when $n=\infty$, it becomes equivalent to hard thresholding.

$$\hat{X}_{j,k} = \begin{cases} Y_{j,k} \left(1 - \frac{|\delta|^n}{|Y_{j,k}|^n}\right), & \text{if } |Y_{j,k}| \geq \delta \\ 0, & \text{if } |Y_{j,k}| < \delta \end{cases}, n=3 \quad (4.18)$$

Figure (4.12) compares the difference between those thresholding algorithms. While at first sight hard thresholding may seem a more natural approach, the continuity of the soft thresholding operation has important advantages.

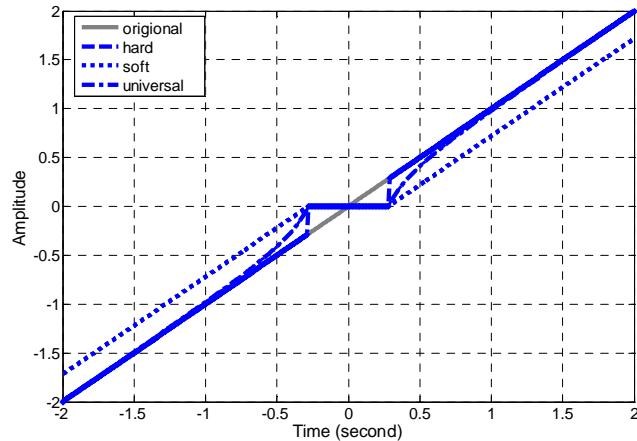


Figure 4.12: Comparison between thresholding algorithms

However, soft thresholding generates biased outputs, which might result in additional error sources in certain conditions while reconstructing denoised signals using soft thresholding high frequency components [Jansen, 2001]. A compromise is a more continuous approach which preserves the highest amplitude coefficients and has a smooth transition from noisy to important coefficients. Therefore, the advantage of the universal

thresholding (red line) is that it can keep the continuity and reduce the impact of a biased signal ($n=3$). A performance analysis of different thresholding algorithms is given in Appendix A.3. It can be seen from Appendix A.3 that the universal thresholding algorithm is capable of providing superior performance than soft, hard and no thresholding operations.

4.4.2 Limitations of the First Generation Denoising Algorithm

The DWT is not translation invariant (shift invariant), meaning that, if a DWT is applied to a shifted version of a signal x , it is not able to get the shifted version of the DWT. Due to this drawback, denoising with the traditional DWT suffers from additional artifacts that can deteriorate its overall performance [Coifman and Donoho, 1995].

The non-translation invariance is the side effect of the downsampling in the filter bank algorithm of the DWT. However, if the shift is a multiple of 2^n , with n being the number of the decomposition level, the DWT will be a shifted version of the original DWT [Coifman and Donoho, 1995], as shown in the left side of Figure (4.13). The figure contains an original signal, two shifted version of the original signals by one sample and 2^1 ($n=1$) sample, respectively.

The DWT was performed using ‘DB3’ with one level ($n=1$) decomposition. The DWT of the original signal (solid) and shifted signal (dash) are clearly not shifted versions of one another when shift=1. However, as the shift becomes 2^1 (dot), the corresponding DWT is the shift version of the original DWT. In contrast, the results obtained by a translation invariant wavelet, Undecimated Wavelet Transform (UDWT) [Guo, 1995], which is given in a later section, is shown to the right side of Figure (4.13).

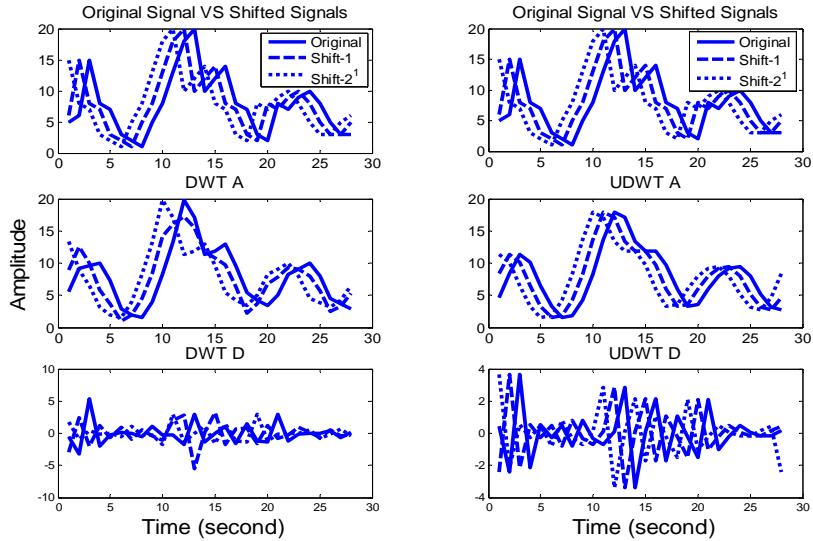


Figure 4.13: Example of non-translation invariance

The lack of translation invariance introduces artifacts when using transform domain thresholding depending on the kind of transform domain one is working in. For wavelet de-noising, the artifacts have to do with behavior near singularities [Guo et al, 1995]. In the neighborhood of discontinuities, wavelet denoising can exhibit pseudo-Gibbs phenomena, which is shown in Figure (4.14), alternating undershoots and overshoots of a specific target level [Coifman and Donoho, 1995]. The red line and blue line represent the original and denoised signal.

While these phenomena are much better than in the case of Fourier-based denoising, in which Gibbs phenomena are global rather than local, and of large amplitude, [Coifman and Donoho, 1995], as shown in Figure (4.15), it seems reasonable to try to do better still. An important observation about such artifacts is that their size is connected intimately with the actual location of the discontinuity. The treatment of reducing the impact of pseudo-Gibbs phenomena for denoising operation is given in the next section.

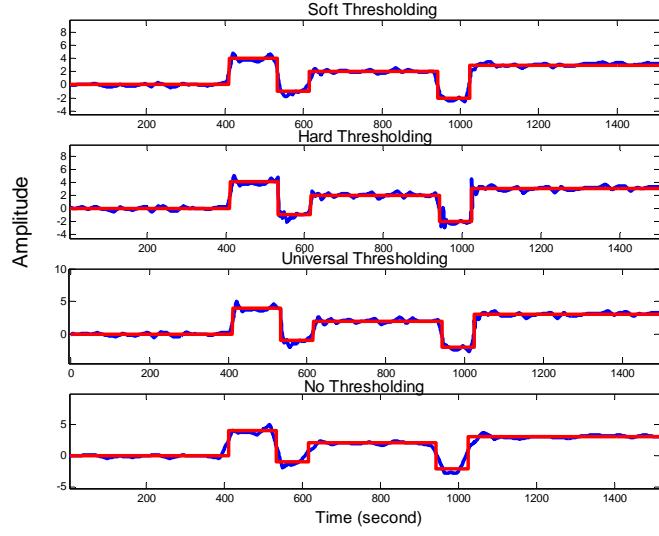


Figure 4.14: Examples of Pseudo Gibbs Phenomema

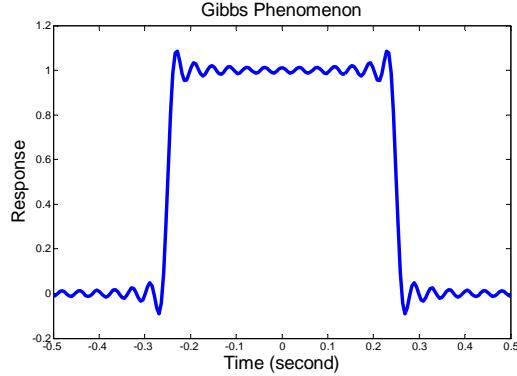


Figure 4.15: Gibbs Phenomena

4.4.3 Second Generation Denoising Algorithm

To reduce the impact of pseudo-Gibbs phenomena, the Undecimated wavelet transform (UDWT) [Guo, 1995], which has been independently discovered under several names, e.g., shift/translation invariant wavelet transform (TIW) [Coifman and Donoho, 1995], stationary wavelet transform (SWT) [Nason and Silverman, 1995] or redundant wavelet transform[Percival et al., 1999], can be applied. The key point is that UDWT is redundant, shift invariant, and it fills in the gap in the DWT and gives a denser approximation to the continuous wavelet transform (CWT) than the approximation

provided by the DWT. The advantage of the UDWT is in signal denoising [Guo et al, 1995]. Generally speaking, the denoising algorithm that utilizes the UDWT is known as a second generation denoising algorithm [Coifman and Donoho, 1995]. A 2 level UDWT decomposition and reconstruction example is given as Figure (4.16) and (4.17)

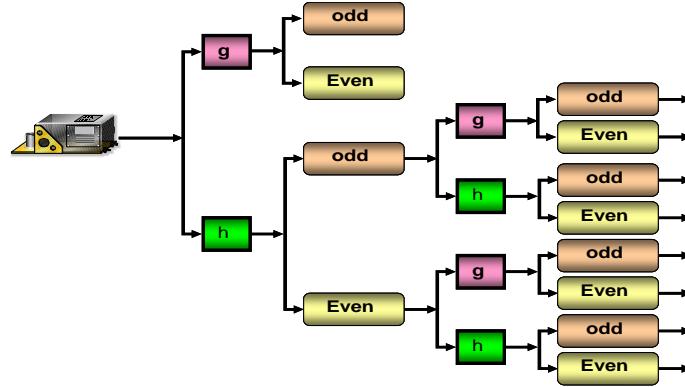


Figure 4.16: UDWT decomposition

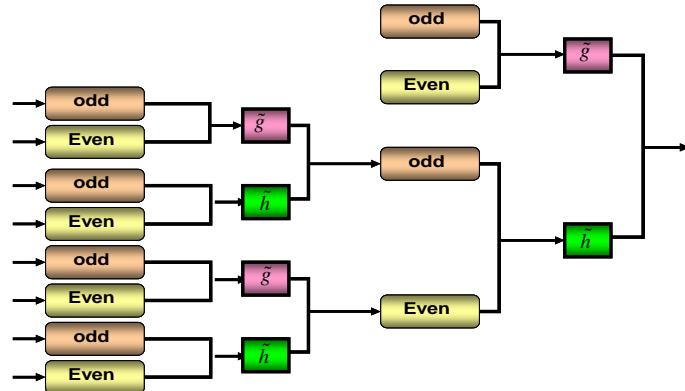


Figure 4.17: UDWT reconstruction

From a signal processing point view, the key of UDWT is that both even and odd decimations are used and the results from both are kept at each stage. As indicated in Figure (4.16). For a given data set, UDWT contains the discrete wavelet transform for every possible origin in the data. Coifman and Donoho [1995] studied extensively the similar characteristics of UDWT and implemented a so called “Translation Invariant Wavelet Transform” (TIW) based on the idea of Cycle-Spinning, or denoising all possible shifts of a signal and then averaging. The idea was originally explored to reduce

pseudo-Gibbs phenomena. Let S_h represent the circular shift operator. For a signal X with length N, $S_h X(k) = X((k + h) \bmod N)$ [Coifman and Donoho 1995]. Let L represent the DWT operator, T represent the thresholding operator, and S_h^{-1} and L^{-1} are un-shift and IDWT operators respectively. The denoised signal is then given by the following Equation:

$$\hat{X} = \frac{1}{N} \sum_{h=0}^{N-1} S_h^{-1} L^{-1} T L S_h X \quad (4.19)$$

An example of TIW denoising is given in Appendix A.3. It can be seen from Figure (A.6) and Table A.5 that TIW reduces the impact of pseudo Gibb's phenomena and improves the denoising results in terms of visualization, SNR and RMS. Similar to the first generation denoising algorithm, the universal thresholding algorithm is capable of providing superior performance than other thresholding algorithms with the utilization of the second generation denoising algorithm.

4.4.4 Spectrum Perspective of Existing Denoising Algorithms

To apply existing denoising algorithms for reducing the impact of noise and disturbances, gaining the knowledge of their characteristics in the spectrum domain is essential. Figure (4.18) illustrates the simulated signal with three sampling rates, 200Hz, 100 Hz and 50 Hz, respectively. The red line and blue line represent the original signal/spectrum and noised signal/spectrum, respectively. Figure (4.19) and Figure (4.20) illustrates the spectrum of the DWT denoising and TIW denoising using those signals in Figure (4.18), respectively. Figure (4.19) demonstrates the DWT denoising with decomposition levels from 1 to 6. It is obvious that the denoised results contain two components, which are given as follow;

- Low frequency component

The frequency of the low frequency content is smaller than the stop bands corresponding to each decomposition level. It can be regarded as the result obtained through the lowpass filter; see Table 4.2 for detailed information regarding the relationship between the decomposition level and stop band.

- High frequency component

The frequency of the high frequency content is higher than the stop bands corresponding to each decomposition level. It can be regarded as the result obtained through the high pass filter plus the denoising algorithm. In general, the thresholding algorithm is applied to remove those components that are identified as noise. As a result, some of the high frequency component can be preserved as they pass through the high pass filter and thresholding algorithm. Such a characteristic might be to the advantage of certain applications; however, it might not be appropriate for kinematic IMU signal denoising. As the general bandwidth of true motion dynamics of land vehicle is very low, see Table 4.4, it is undesirable to keep these high frequency components.

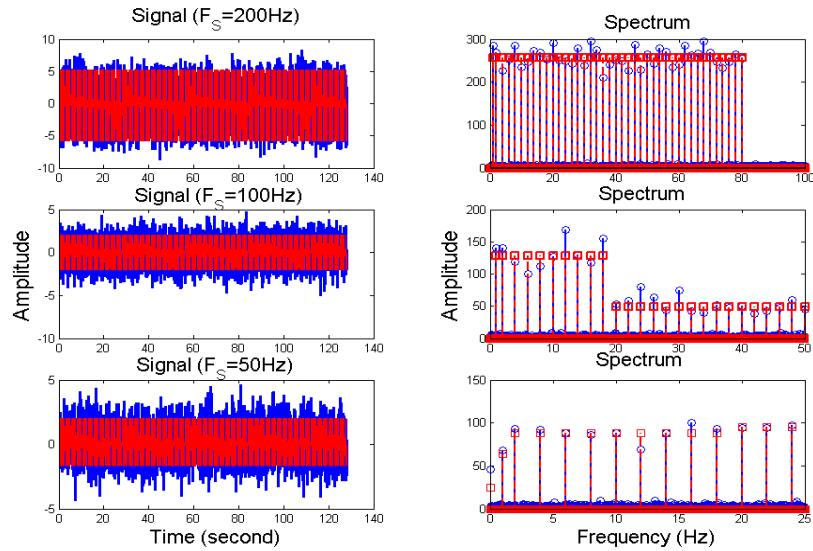


Figure 4.18: Simulated signals

It can be seen from Figure (4.19) that the bandwidth of the preserved high frequency

components range from 20Hz to a maximum dynamic frequency (i.e., 80Hz in this simulation) with a peak that is located between 40 Hz to 50 Hz. Consequently, due to the limitations of the nyquist frequency, such undesirable high frequency components appear only when the sampling rate is higher than 100Hz. In other words, it vanishes when the sampling rate is lower than 50 Hz. Similar to Figure (4.19), Figure (4.20) illustrates TIW denoising with decomposition levels from 1 to 6. The denoising results also contain low and high frequency components. It can be seen from Figure (4.20) that the TIW suppressed those undesirable high frequency components, whose bandwidth range is from 20 Hz to the maximum dynamic frequency (i.e., 80Hz in this simulation) with peaks that are located between 40 Hz to 50 Hz, significantly in comparison with DWT denoising. However, those remaining high frequency components should be further reduced for IMU signals denoising applications as the impact of such remaining high frequency components appear when the sampling rate of the IMU is higher than 100 Hz.

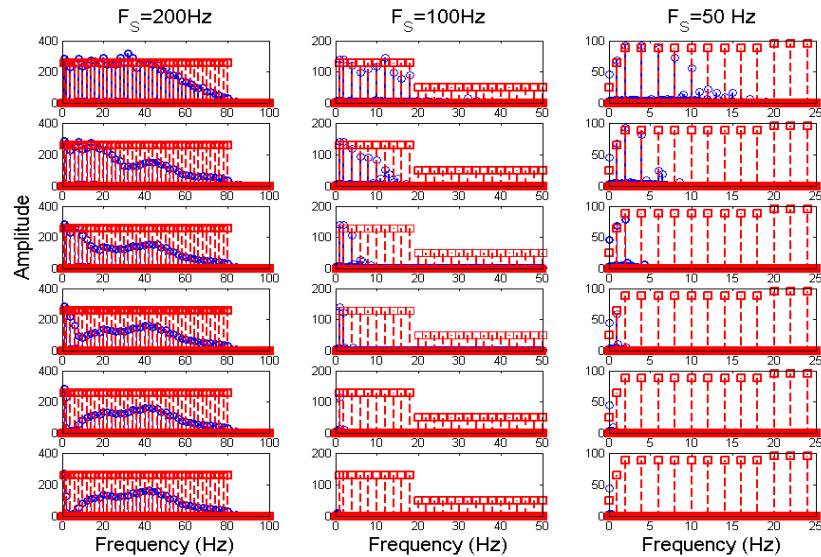


Figure 4.19: Spectrum of DWT denoising

In spite of these limitations, both Figure (4.19) and Figure (4.20) demonstrate an important fact that the existing denoising algorithm can preserve information that is smaller than the pass band. This characteristic is very important for IMU signal denoising as losing any true motion dynamic information degrades the navigation accuracy

significantly. Consequently, Tables 4.2 and Table 4.4 can be applied to set up the optimal decomposition level required for the denoising operation without losing true motion dynamic information, as indicated in Table 4.5. Recalling Figure (3.12) and Figure (3.13), a conceptual plot of the frequency spectrum of the errors in the measurements made by the inertial sensors in an INS/GPS integrated system with the utilization of a perfect denoising algorithm is shown in Figure (4.21). The goal is to remove all the short term errors and disturbances.

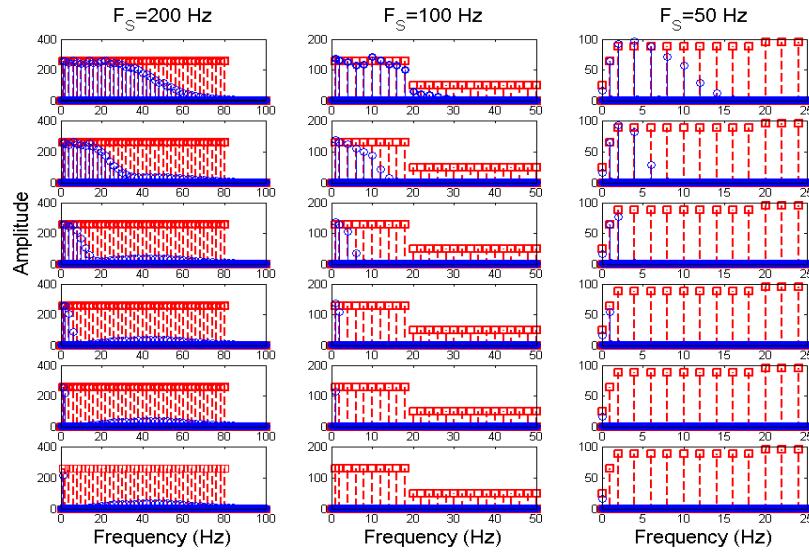


Figure 4.20: Spectrum of TIW denoising

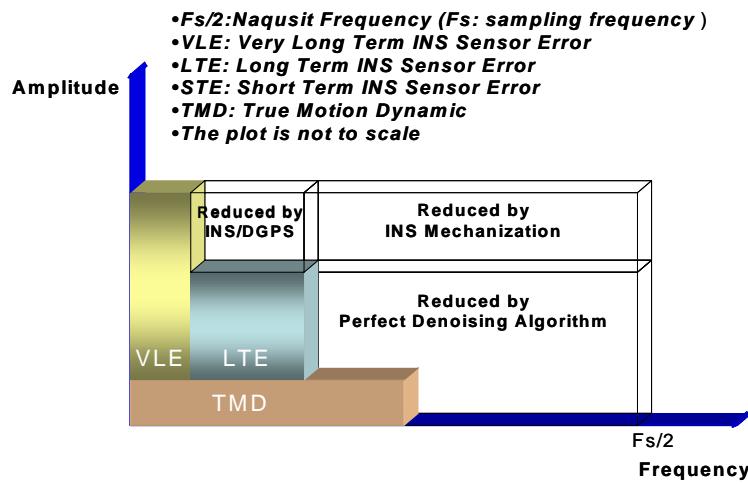


Figure 4.21: Conceptual plot of the spectrum of INS sensor errors

(After INS/DGPS+ perfect denoising algorithm)

In addition, Lin [1993] and Skaloud [1999] suggested the utilization of a lowpass filter to band limit the IMU signals and remove part of the short term errors and disturbances, as shown in Figure (4.22). It can remove those short term errors whose frequencies are higher than the stop band of the lowpass filter; however, it has cannot remove those short term errors whose frequencies are lower than the stop band.

Figure (4.19) and (4.20) depict the major limitation of applying either the 1st or the 2nd generation denoising algorithms in the remaining high frequency component, as shown in Figure (4.23). They are capable of suppressing the short term errors whose frequencies are lower than the stop band and reducing part of those short term errors whose frequencies are higher than the stop band. The remaining question is how to remove those remaining short term errors whose frequencies are higher than the stop band and reduce those short term errors whose frequencies are lower than the stop band when the existing wavelet denoising algorithm is applied to denoise the IMU signals? Therefore, a cascade denoising that can achieve this goal is given in next section.

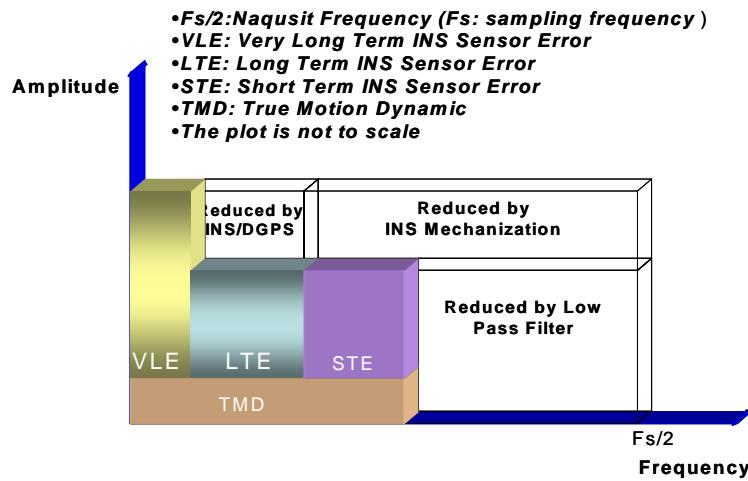


Figure 4.22: Conceptual plot of the spectrum of INS sensor errors
(After INS/DGPS+ lowpass filter)

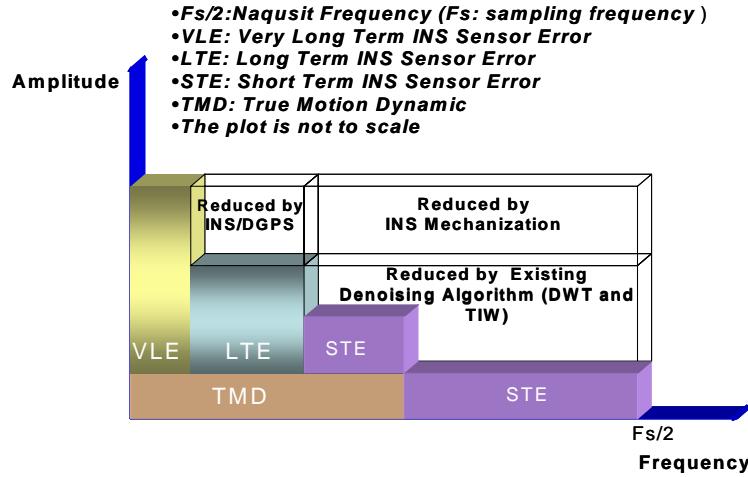


Figure 4.23: Conceptual plot of the spectrum of INS sensor errors
(After INS/DGPS+ existing denoising algorithm)

4.5 Cascade Denoising Algorithms

Through the spectrum analysis of DWT/TIW and kinematics IMU signals, it is possible to determine the bandwidth of true motion dynamics that are sensed by each inertial sensor and the stop band of wavelet based lowpass filters, see Table 4.2 and 4.4 for details. As a result, an optimal decomposition level of the wavelet based lowpass filter can be determined. Since the signals whose frequency ranges out of the bandwidth of true motion dynamic are undesirable, the wavelet based lowpass filter with an optimal decomposition level (L) for each sensor can be applied first to remove the undesirable high frequency components whose frequencies are higher than the stop bands of the lowpass filters. Then a denoising algorithm can be applied to remove the remaining short term errors whose frequencies are lower than the stop bands of the lowpass filters. This algorithm is named the cascade denoising algorithm [Chiang, et al., 2004b], as shown in the Figure (4.24).

Figure (4.25) and Figure (4.26) illustrates the spectrum of the cascade denoising algorithm using DWT and TIW with decomposition levels from 1 to 6 respectively. The red line and blue represent the original spectrum and noised spectrum. These figures demonstrate that cascade denoising can remove the undesirable high frequency

components completely in comparison with existing denoising algorithms that were discussed in the previous section. In addition, Figure (4.26) shows that the cascade denoising algorithm with TIW can provide better performance for reducing the noise components contained in the low frequency components with frequencies lower than the stop band without losing any of the true motion dynamic information.

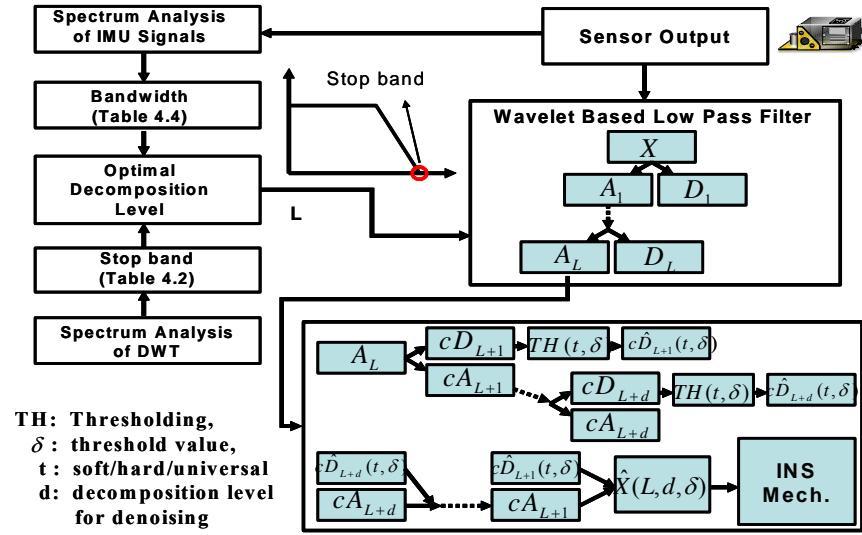


Figure 4.24: Cascade denoising algorithm

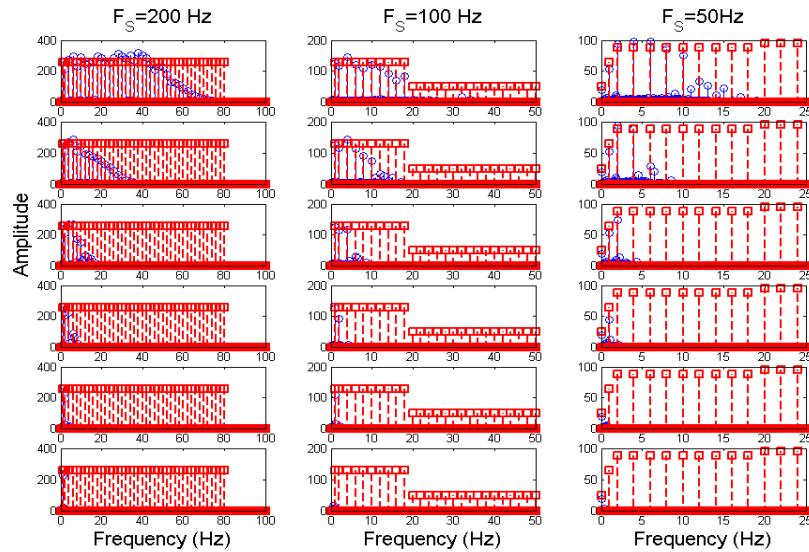


Figure 4.25: Spectrum of cascade DWT denoising

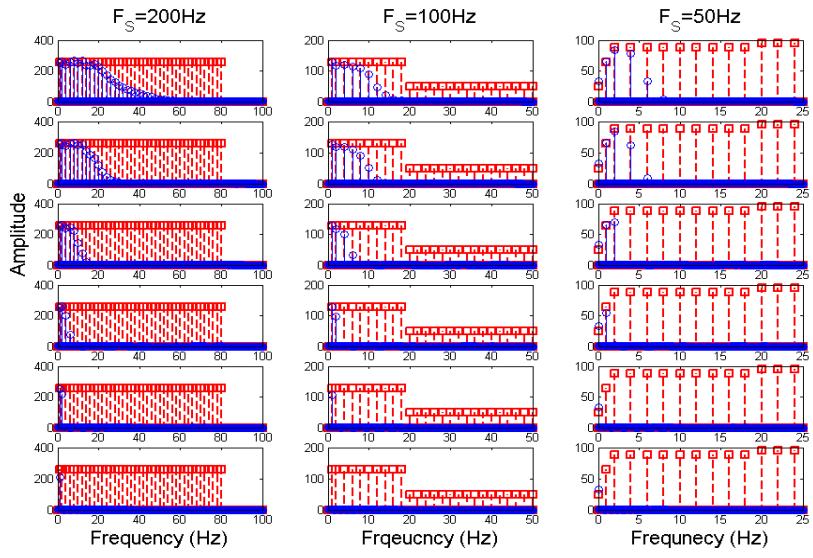


Figure 4.26: Spectrum of Cascade TIW denoising

Figure (4.27) and Figure (4.28) illustrate the spectrums of the raw X-Gyro measurements of the CIMU (sampling rate = 200Hz) after applying the DWT denoising and cascade TIW denoising, respectively. Only the high frequency components whose frequencies are higher than the stop bands at each decomposition level are presented. The decomposition levels range from 1 to 6.

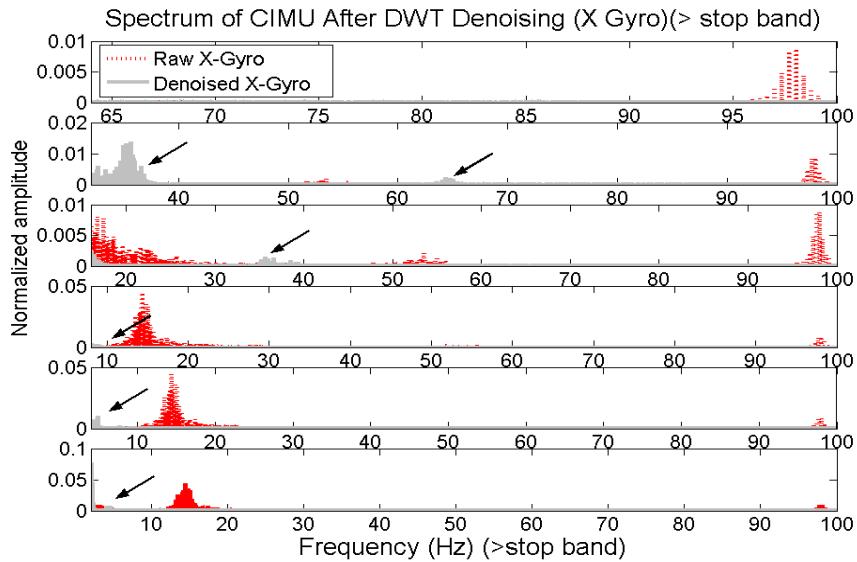


Figure 4.27: Spectrum of CIMU X-Gyro after DWT denoising ($>\text{stop band}$)

Similar to Figure (4.25), it can be seen from Figure (4.27) that the remaining high frequency noise components appear, as indicated by the arrow, at each decomposition level after applying DWT denoising. In contrast, these high frequency components are removed completely when the cascade denoising algorithm is applied, as indicated in Figure (4.28). Consequently, the cascade denoising algorithm is superior to existing denoising algorithms in the spectrum domain as it can remove the remaining high frequency noise components.

Consequently, the utilization of the cascade denoising algorithm can remove those remaining high frequency components in Figure (4.23) completely. In addition, it is capable of removing more short term errors whose frequencies are lower than the stop band, as illustrated in Figure (4.29). Comparing Figure (4.29) with Figures (4.22) and (4.23), it is clear that the developed cascade denoising algorithm is superior to traditional denoising algorithms or lowpass filters in the spectrum domain when it is applied to remove short term INS sensor errors.

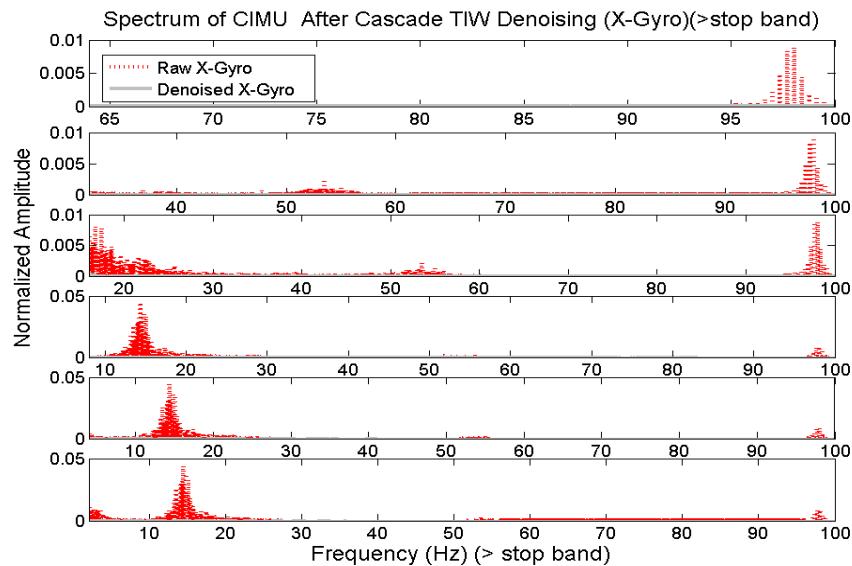


Figure 4.28: Spectrum of CIMU X-Gyro after cascade TIW denoising (>stop band)

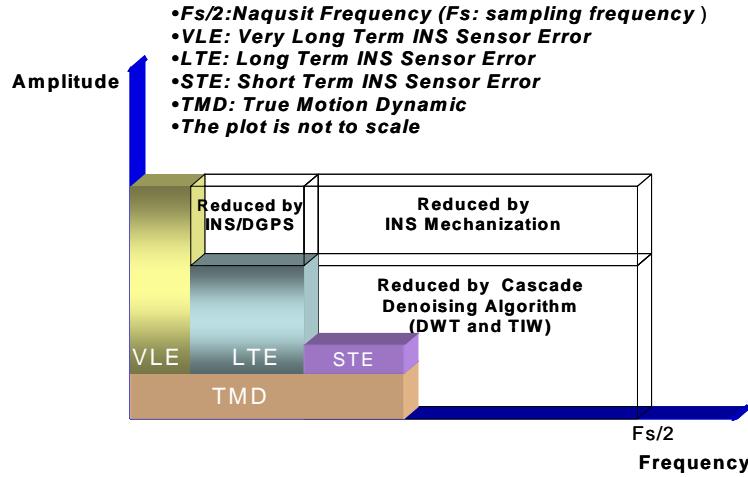


Figure 4.29: Conceptual plot of the spectrum of INS sensor errors
(After INS/DGPS+ cascade denoising algorithm)

4.6 Performance Analysis of Cascade Denoising Algorithm

To evaluate the performance of the cascade denoising algorithm in the position domain, a field test in a land-vehicle was collected on March 5, 2003. The used IMU is a prototype from Inertial Science INC. (ISI), and contains Analog Devices MEMS gyros and Colibrys MEMS accelerometers. The reference system is an Applanix POS LV 320 that uses a LN200 IMU (see appendix C for detailed specifications of these IMUs). The vehicle drove two standard circles (large rectangle circle) and then drove three times around a small round circle.

To assess the performance of the cascade denoising algorithm, GPS signal blockages were simulated, by removing GPS data, along various portions of the test trajectory. Ten simulated GPS signal blockages that contain several dynamic variations were introduced. As shown in Figure (4.30), the red line represents the reference trajectory and the blue line segments indicate the trajectories during simulated GPS signal blockages.

Figure (4.31) and Table 4.6 illustrate the performance of existing denoising algorithms using the DWT. The sampling rate of the ISI IMU is 200 Hz thus the optimal decomposition levels (L) for each sensor can be obtained using Table 4.5.

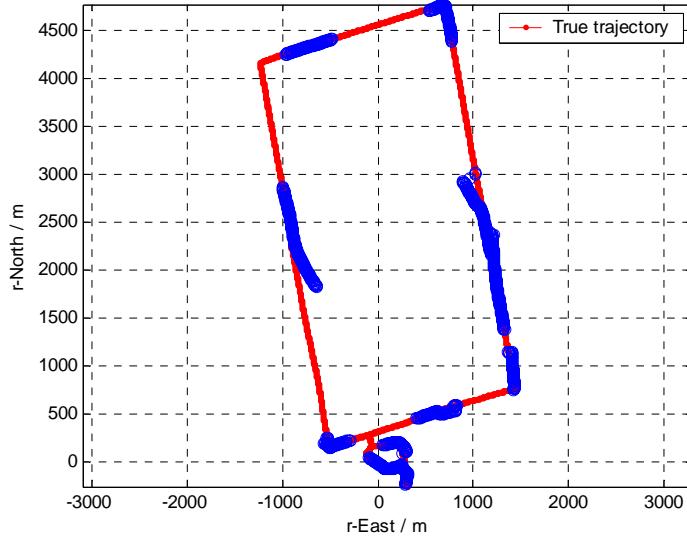


Figure 4.30: The reference trajectory and simulated GPS blockages

It should be mentioned here that the maximum of the absolute position error (MAX_N and MAX_E) always happens at the end of the blockage periods. It can be seen from Table 4.12 that the Root Mean Square error (RMS) of seven GPS blockages were improved up to 13% (4.47m) and the maximum position errors in both directions were successfully reduced in these improved cases. In contrast, the RMS errors of the remaining three GPS blockages were degraded up to 0.6 metres. Consequently, the rate of improvement reached 70% (7/10).

Figure (4.32) and Table 4.7 illustrate the performance of the cascade denoising algorithm using TIW. Similarly, the optimal decomposition levels (L) for wavelet lowpass filters were decided using Table 4.5 and the decomposition level (d), as indicated in Figure (4.30) for the denoising operation was given empirically. In general, $d=2$ or 3 is recommended for IMU signals denoising based on empirical trials. The results indicated that the true motion dynamic signal might be deteriorated when d is larger than 3. It can be seen from Figure (4.32) that the improvements with the utilization of the cascade denoising algorithm during several GPS blockage periods were significant enough to be visualized.

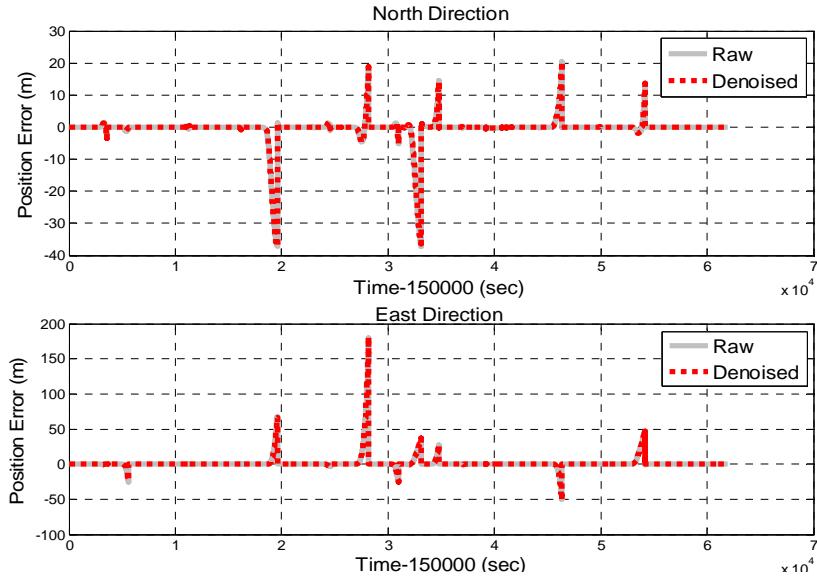


Figure 4.31: Position errors during GPS blockages

Table 4.6: Performance of DWT denoising

Blockage no.	RAW				Denoising (DWT)		
	Blockage Length(S)	MAX_N (m)	MAX_E (m)	RMS Total	MAX_N (m)	MAX_E (m)	RMS Total
1	30	3.43	1.12	1.18	3.41	1.20	1.19
2	30	1.23	25.82	9.89	1.08	26.02	10.65
3	60	37.07	68.27	33.86	36.95	64.78	29.39
4	30	1.45	3.50	1.84	1.67	3.91	2.08
5	60	18.90	179.30	70.58	18.87	180.52	70.11
6	30	5.16	25.53	10.92	5.06	25.45	10.68
7	70	37.04	36.97	25.23	36.64	36.45	24.50
8	30	14.39	27.27	15.26	14.55	26.53	14.83
9	44	20.39	50.52	18.43	20.28	50.12	13.77
10	60	13.50	47.48	23.69	13.70	47.21	22.43

As indicated in Table 4.13, the rate of improvement was 80% (8/10). The improvement in terms of the RMS errors ranged from 20 centimetres to 10 metres and the improvement in terms of the percentage ranged from 5% to 25%. Consequently, the cascade denoising algorithm is superior to the DWT denoising in both the position domain and spectrum domain. For comparison, the performance of the lowpass filter denoising is given in Appendix A.4. As indicated in Figure (A.7) and Table A.6, the overall performance was similar to the DWT denoising algorithm.

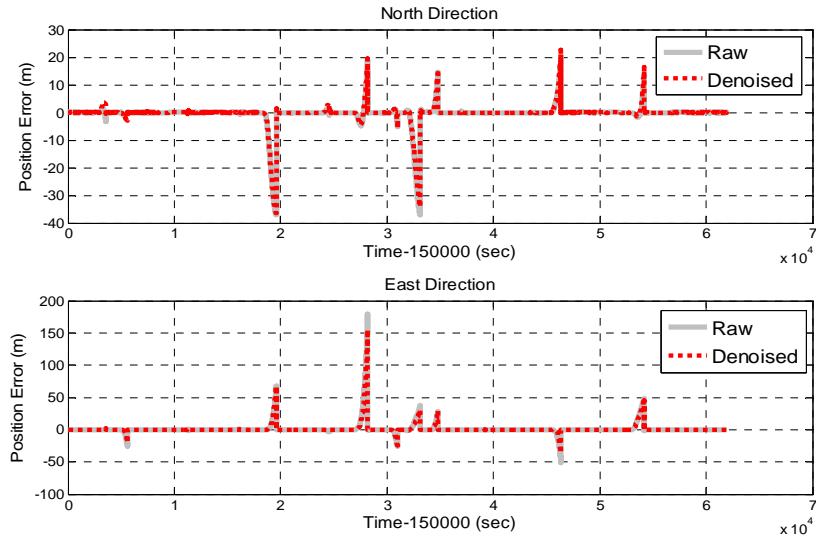


Figure 4.32: Position errors during GPS blockages

Table 4.7: Performance of cascade TIW denoising

Blockage no.	Denoising (DWT)				Cascade Denoising (TIW)		
	Blockage Length(S)	MAX_N (m)	MAX_E (m)	RMS Total	MAX_N (m)	MAX_E (m)	RMS Total
1	30	3.41	1.20	1.19	3.56	1.65	1.19
2	30	1.08	26.02	10.65	2.87	19.54	7.47
3	60	36.95	64.78	29.39	36.46	67.77	32.10
4	30	1.67	3.91	2.08	1.78	3.28	1.89
5	60	18.87	180.52	70.11	19.66	155.72	60.36
6	30	5.06	25.45	10.68	5.82	24.10	10.63
7	70	36.64	36.45	24.50	34.37	30.29	21.80
8	30	14.55	26.53	14.83	14.19	29.07	14.32
9	44	20.28	50.12	13.77	20.12	36.50	14.01
10	60	13.70	47.21	22.43	13.22	44.67	22.42

The cascade denoising algorithm has been shown to be capable of providing superior performance than existing 1st generation (DWT) and 2nd generation (TIW) denoising algorithms in the spectrum and position domains. However, the impact of the cascade denoising algorithm on different IMUs with different accuracy levels remains unknown; consequently, an investigation of this issue is given in Chapter 7.

CHAPTER 5

ARTIFICIAL NEURAL NETWORKS METHODOLOGY

With the evolution of modern computer technology in hardware and software, the field of artificial intelligence has been receiving more attention in the development of future technology. Artificial intelligence has been verified as a successful and effective tool for providing solutions to certain engineering and science problems that can not be solved properly using conventional techniques [Cawsey.1998]. The intelligence can be defined as the ability to learn, understand and adapt [Honavar and Uhr, 1994]. Humans possess some robust attributes of learning and adaptation abilities, and that's what makes them so intelligent. Thus the goal of artificial intelligent technologies, which include artificial neural networks (ANNs), fuzzy logic, evolutionary computing, probabilistic computing, expert systems, and genetic algorithms, is to provide some intelligence and robustness in the complex and uncertain systems similar to those seen in natural biological species [Honavar and Uhr,1994].

Among these artificial intelligent techniques, ANNs was chosen in this research for building the foundation for developing the conceptual intelligent navigator that has an artificial intelligent based INS/GPS data fusion and navigation algorithm. The requirement of such an algorithm is to overcome the limitations of current INS/GPS data fusion which is based solely on the Kalman filter. Kalman filter approach has limitations, which have already been discussed in Chapter 3. Chiang [2003] summarized the comparison between the Kalman filter and ANNs based INS/GPS data fusion and navigation algorithm as in Table 5.1.

ANNs have been extensively studied with the aim of achieving human-like performance, especially in the field of pattern recognition and robot control and navigation [Mandic and Chambers, 2001]. ANNs are composed of a number of nonlinear computation elements which operate in parallel and are arranged in a manner reminiscent of biological neural interconnections. In addition, ANNs are designed to mimic the human brain and

duplicate its intelligence by utilizing adaptive models that can learn from the existing data and then generalize what it has learnt [Ham and Kostanic, 2001].

Table 5.1: Comparison between KF and NN for INS/GPS data fusion and navigation
 (After Chiang, 2003)

	KF	NN
Model Dependency:	<i>States (Mathematical model; deterministic model + stochastic model).</i>	<i>Empirical and adaptive model</i>
Prior Knowledge Dependency:	<i>Required (Mainly Q and R matrix), provided by human experts</i>	<i>Not required</i>
Sensor Dependency:	<i>Re-design or re-tuning parameters of Kalman filter are needed for different sensor</i>	<i>An adaptable, sensor independent algorithm</i>
Linearity dependency:	<i>Linear Processing</i>	<i>Nonlinear Processing</i>

ANNs have shown promise in offering alternative solutions to a lot of engineering problems, where traditional models have failed or are too complicated to build. Due to their nonlinear nature, ANNs are capable of expressing much more complex phenomena than some linear modeling techniques. They extract the essential characteristics from the numerical data as opposed to memorizing all of it [Abhijit and Robert 1996]. ANNs offer a convenient way to form an implicit model without establishing a traditional, physical mathematical model of the underlying phenomenon. In contrast to traditional models, ANNs work in a way that little, or no, a priori knowledge of the mathematical process is required. ANNs can be applied for the construction of mapping functions that transform inputs to outputs via a “black box” [Bender, 1996]. As indicated in section 3.3.1, Kalman filter requires certain assumptions to make it work properly, on the contrary, ANNs build its model based on empirical learning thus it does not require certain statistical assumption. However, such empirical and adaptive model requires the existence of the implicit nonlinear function relationships between inputs and outputs. Wu [1995] summarized the main reason for using neural networks for prediction rather than classical time series algorithms as

- Self monitoring , self learning
- Iterative forecast
- Capability to cope with non-stationary and non-linear processes
- Parametric and nonparametric prediction
- Fast and accurate (at least as fast/accurate as current algorithms)

Thus, the fundamentals of ANNs are given first in this chapter followed by several aspects regarding static versus dynamic neural networks. Topologies of different neural network architectures, standard backpropagation learning algorithms, second order learning algorithms and linearized recursive estimation learning algorithms will also be discussed. The performance evaluation of a dynamic neural network and a static neural network in terms of the position and time domains is given using real data obtained through INS/GPS integrated land vehicular systems. The conclusions will be the decisive factor in which ANN (static or dynamic) to use as the core algorithm for developing the conceptual intelligent navigator, which is given in the next chapter.

5.1 Fundamentals of Artificial Neural Networks

The history of neural networks comes from attempts of modeling a system by simulating the most basic functions of human brains. In fact, the motivation of studies in neural networks comes from the flexibility and power of information processing that conventional computing machines do not have. Although most computers can process faster and more precisely than human brains, human beings have the ability to obtain experience then make more sensible decisions [Honavar and Uhr, 1994]. Similar to the fact that the human brain generalizes the rules, the neural network system can ``learn by examples and experience'' and perform a variety of nonlinear functions that are difficult to describe mathematically [Haykin, 1999].

ANNs are a narrow-sensed abstraction of the human brain, thus the organization of the artificial neural system is very similar to that of biological neurons. The fundamental element, an artificial neuron, is a model based on the behavior of biological neurons that exhibit most of the characteristics of human brains. This is the most significant difference

from conventional computers, which have internal fixed instructions to perform specific functions.

ANNs can also be described as highly parallel distributed computing models [Kohonen, 2001]. The neurons are highly connected with strengths which are dynamically changed during learning process. Though ANNs are not an exact duplication of the biological human brain, it is important to begin with understanding fundamental concepts of biological neurons and the human brain. A brief review of the fundamental concepts of biological neurons and the human brain is given as Appendix B.1.

5.1.1 Artificial Neurons and Neural Networks

To gain more appreciation to NNs, a simplified neuron model is presented in Figure (5.1a). The function schemes of this multiple input and signal output neuron are illustrated in Table 5.2 [Ham and Kostanic, 2001]. As illustrated in Figure (5.1a), two key elements in a biological neuron can be identified: the synapse and soma. They are responsible for providing learning adaptation knowledge (storage of knowledge or memory of past experience) and nonlinear mapping operations on neural information. From a mathematical perspective, the processing of information within a neuron involves two distinct mathematical operations, which are given in Table 5.3 [Anderson and McNeill, 1992]:

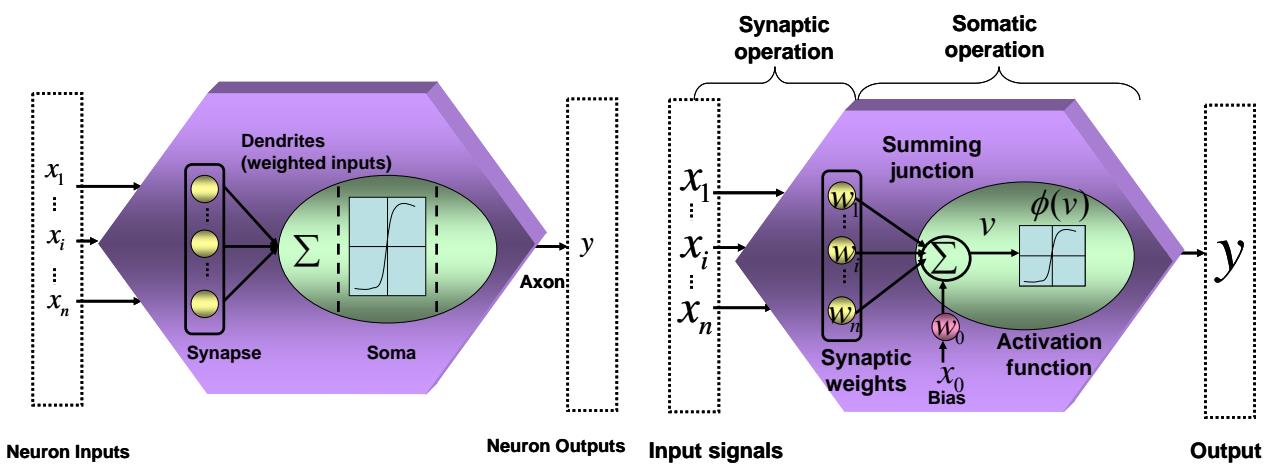


Table 5.2: The function scheme of a neuron

Element	Function Scheme
Dendrites:	<i>They are made of a highly branching tree of fibers, and function as input points to the main body of the neuron.</i>
Synapse:	<i>It provides the storage space for the past experience or knowledge. In other words, It withholds long term memory to the past accumulated knowledge or experience. It receives information from sensors and other neurons and provides outputs through the axons.</i>
Soma:	<i>The neural cell body is called soma. It receives synaptic information and performs further processing of the information</i>
Axon:	<i>The neuron output line is called the axon.</i>

Table 5.3: Mathematical operations of an artificial neuron

Element	Function
Synaptic operation:	<i>It assigns a weight to each incoming signal according to the knowledge or past experience that was stored in the synapse. In fact, the strength (weight) of the synapse is the key representation of the storage of knowledge and the memory for previous knowledge.</i>
Somatic operation:	<i>It provides various mathematical operations such as aggregation (summation), thresholding, nonlinear activation, and dynamic processing to the synaptic inputs. If the weighted aggregation of the neuron inputs exceeds a certain threshold, the soma will produce an output signal to its axon</i>

Being inspired by the biological neuron, an artificial neuron model is illustrated in Figure (5.1b). From a mathematical perspective, three basic elements of an artificial neuron are identified as follows [Haykin, 1999]:

- **Synapses links (weights links, synaptic):** Each of the links is labeled by a weight of its own. A signal x_i at the input of the synapse i connected to a neuron is multiplied by the synaptic weight w_i . The synaptic weight of an artificial neuron can be a negative or positive value.
- **An adder (somatic):** It adds up the weighted input signals. Such a summation is expressed as Equation 5.1. As shown in Figure (5.1b), an artificial neuron model also includes an external bias, denoted by x_0 . The bias x_0 is applied to increase or

lower the input of the activation function. The synaptic weight of the bias, w_0 , is usually assigned to be equal to 1.

$$v = \sum_{i=0}^n w_i x_i \quad (5.1)$$

- **An activation function (somatic) (φ):** It limits the amplitude of the neuron output when φ is nonlinear. In general, φ can be a continuous, binary, bipolar, or linear function in certain cases. When φ is nonlinear, its finite limits are typically normalized in the range of either [0, 1] (binary) or [-1, 1] (bipolar). It is recommended that the nonlinearities can serve to enhance the networks abilities for function approximation and noise-immunity [Ham and Kostanic, 2001]. In some literatures, an activation function is named as a transfer function as well.

$$y = \varphi(v) \quad (5.2)$$

The activation function can be a linear or nonlinear function. In fact, there are many different types of activation functions. The choice of one type over another is application dependent. Four of the most common types of activation functions are:

1. **Hard limiter:** This is a binary function (or bipolar) that hard-limits the input to the function to either a 0 or 1 for the binary type and either -1 or 1 for the bipolar type. The binary hard limiter (φ_{hl}) is sometimes referred to as the threshold function and the bipolar hard limiter is referred as the symmetric hard limiter (φ_{shl}).The outputs can be expressed as

$$y = \varphi_{hl}(v) = \begin{cases} 0 & v < 0 \\ 1 & v \geq 0 \end{cases} \quad \text{and} \quad y = \varphi_{shl}(v) = \begin{cases} -1 & v < 0 \\ 0 & v = 0 \\ 1 & v > 0 \end{cases} \quad (5.3)$$

2. **Linear function (φ_{lin}):** It is a continuous function. Mathematically, the output of the linear activation function is written as $y = \varphi_{lin}(v) = v$.

3. Saturating linear function or piecewise linear function: This type of function can be binary or bipolar for the saturation limits of the output. The bipolar saturating linear function will be referred to as a symmetric saturating linear function. The output for the saturating linear function (φ_{sl}) (i.e., binary outputs) and the symmetric saturating linear function (φ_{ssl}) are given by

$$y = \varphi_{sl}(v) = \begin{cases} 0 & v < -\frac{1}{2} \\ v + \frac{1}{2} & -\frac{1}{2} \leq v \leq \frac{1}{2} \\ 1 & v > \frac{1}{2} \end{cases} \quad \text{and} \quad y = \varphi_{ssl}(v) = \begin{cases} -1 & v < -1 \\ v & -1 \leq v \leq 1 \\ 1 & v > 1 \end{cases} \quad (5.4)$$

4. Sigmoid function (s-shaped) function: The nonlinear sigmoid function is the most common type of activation function applied to construct artificial neural networks. The first type of sigmoid function is the binary sigmoid function (φ_{bs}). The output has a binary range and is expressed as

$$y = \varphi_{bs}(v) = \frac{1}{1 + e^{-\alpha v}} \quad (5.5)$$

Where α is the slope parameter of the binary sigmoid function. The binary sigmoid is a continuous and differentiable function. The differentiability of an activation function plays an important role in neuron computing. Equation 5.6 illustrates the derivative of the binary sigmoid function.

$$\frac{dy}{dv} = \frac{d\varphi_{bs}(v)}{dv} = \frac{\alpha e^{-\alpha v}}{(1 + e^{-\alpha v})^2} = \alpha \varphi_{bs}(v)[1 - \varphi_{bs}(v)] \quad (5.6)$$

The bipolar type of the sigmoid function is called the hyperbolic tangent sigmoid (ϕ_{hts}).

The output has a bipolar range and is expressed as

$$y = \phi_{hts}(v) = \tanh(\alpha v) \quad (5.7)$$

Equation 5.8 illustrates the derivative of the hyperbolic tangent sigmoid

$$\frac{dy}{dv} = \frac{d\phi_{hts}(v)}{dv} = \alpha[1 + \tanh(\alpha v)][1 - \tanh(\alpha v)] \quad (5.8)$$

Figure (5.2) shows the shape of these activation functions and the derivatives of binary and bipolar sigmoid activation functions. The slop parameter α is set to be 1.

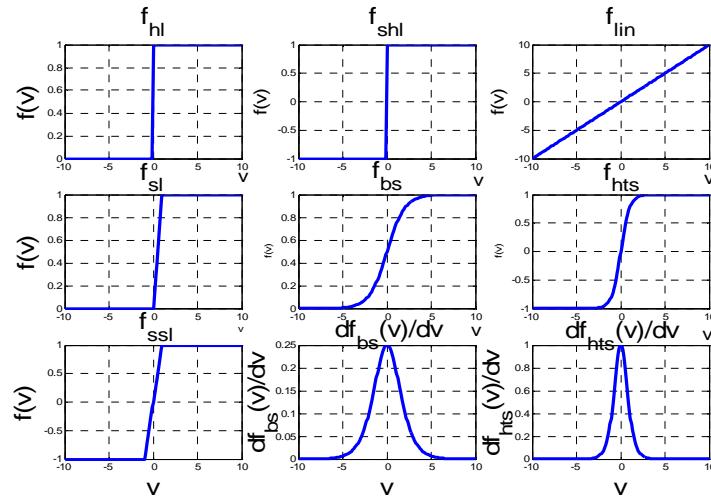


Figure 5.2: Shapes of activation functions

5.1.2 ANNs Architecture

Prior to looking into the general ANNs architectures, two fundamental elements of constructing any ANN architecture are given first:

- **Layers:** Neurons are organized in layers. In general, the architecture of ANNs might consist of one input layer, one or more hidden layers and one output layer, as shown in Figure (5.3). The function scheme of each layer is given in Table 5.4.

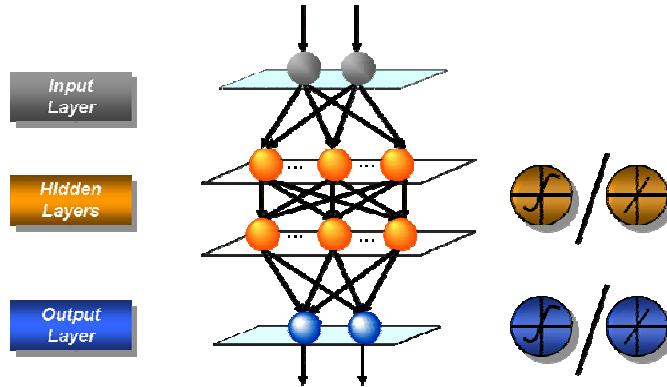


Figure 5.3: Layer organization

Table 5.4: Function scheme of each layer

Layers	Function
Input layer:	<i>The input neurons receive the input data from the outside environment (e.g. sensor) and pass it to the hidden layer for further processing. It is simply an interface between the network and the sensors.</i>
Hidden layers:	<i>The hidden neurons are placed between the input layer and output layer. They are the real processing units of the network. For nonlinear input/output mapping or function approximation applications it is recommended that the activation function of hidden neuron to be nonlinear (e.g. sigmoid).</i>
Output layer:	<i>The output neurons transform the data from the input layer or the hidden layer to the outputs of the network. The output neuron can be linear or nonlinear.</i>

- **Connections:** Neurons are connected via a network of paths carrying the output of one neuron as input to another neuron. These connection can be divided into two major categories, as shown in Table 5.5 [Anderson and McNeill, 1992]:

Using the above elements, the general architectures of ANNs can be given as follows:

- **Single-Layer Feed-Forward Neural Networks (SFNNs):** A layered NN is a network of neurons organized in the form of layers. The simplest layered NN is

the single layered NN that consists of an input layer and an output layer, as shown in the Figure (5.4). Weight connections are shown emanating from a layer of certain index to a layer of higher index. No weight connections are allowed amongst the neurons belonging to the same layer; thus, they are denoted as feed-forward NNs.

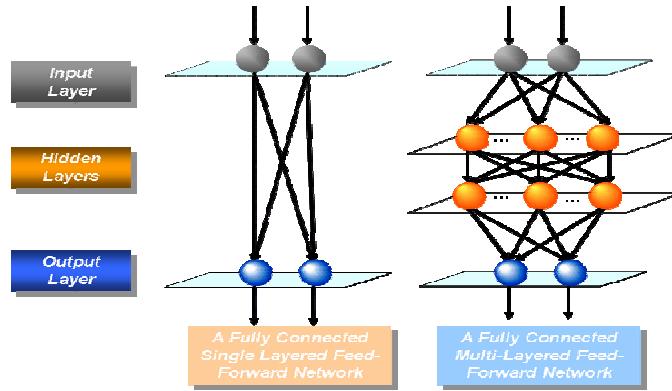


Figure 5.4: A single layered and a multi-layered feed forward architectures

- **Multi-Layer Feed-Forward Neural Networks (MFNNs):** The second type of a feed-forward neural network distinguishes itself by one or more hidden layers. The function of a hidden layer is to intervene between the external input and the network output in some useful manner. The source nodes in the input layer of the network supply respective elements of the input vector, which constitute the input signals applied to the neurons in the first layer (i.e., the first hidden layer). The output signals of the first layer are used as inputs to the second layer, and so on for the rest of the network. Figure (5.4) shows a fully connected multi-layer feed-forward neural network with a single hidden layer. Such architecture is known as static network architecture and is discussed in more details in section 5.2.
- **Recurrent Neural Networks (RNNs):** The difference between the feed-forward architecture and the recurrent architecture is that the latter has at least one feedback loop. A recurrent architecture might consist of inter-layer feedback (i.e., from output layer to input layer) and an intra-layer feedback or self-feedback loops [Anderson and McNeill, 1992]. Self-feedback loops refer to a situation

where the output of a neuron is fed back into its own input. Figure (5.5) illustrates a single layer recurrent architecture with inter-layer feed back loops and a simple recurrent network (SRN), also known as Elman network [Elman, 1990]. Such architecture is known as dynamic network architecture and is discussed in more detail in section 5.3.

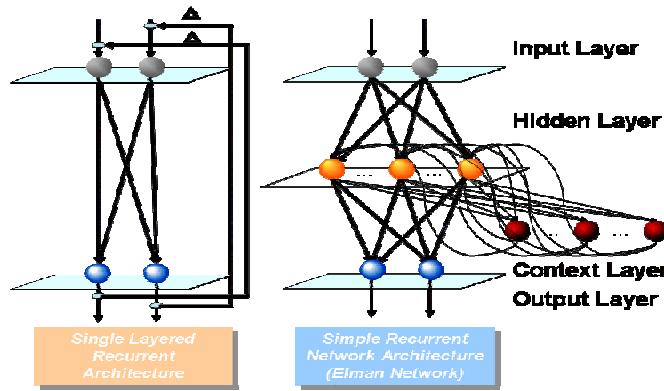


Figure 5.5: Recurrent network architectures

Table 5.5: Types of connection

Connection	Function
Inter-layer connection:	<ul style="list-style-type: none"> Fully connected: Each neuron on the first layer is connected to every neuron on the second layer. Partially connected: A neuron of the first layer does not have to be connected to all neurons on the second layer. Feed forward: The neurons on the first layer send their output to the neurons on the second layer, but they do not receive any input back from the neurons of the second layer. Bi-directional: There is another set of connections carrying the output of the neurons of the second layer into the neurons of the first layer.
Intra-layer connection:	<ul style="list-style-type: none"> Recurrent: The neurons within a layer are fully- or partially connected to one another or fed back to themselves. After these neurons receive input from another layer, they communicate their outputs with one another a number of times before they are allowed to send their outputs to another layer On-center/off surround: A neuron within a layer has excitatory connections to itself and its immediate neighbors, and has inhibitory connections to other neurons.

5.1.3 Learning Procedures

The common learning procedures in NN can be grouped as follows [Anthony and Bartlett, 1999];

- **Supervised learning:** An essential ingredient of supervised learning is the availability of an expert, see Figure (5.6). The expert has full knowledge of the model or the problem that is represented by a set of nonlinear input-output mapping relationships. However, the model is unknown to the NN. Suppose that the teacher and the NN are exposed to a training vector (i.e., example) drawn from the unknown model. By virtue of built-in knowledge, the teacher is able to provide the NN with a desired or target response for the training vector. Indeed, the desired response represents the optimal action to be performed by the NN. The network parameters (i.e., synaptic weights and biases) are adjusted under the combined influence of the training vector and the error signal that is defined as the difference between the actual response of the network and the desired response. This adjustment or tuning procedure is carried out iteratively in a step by step fashion with the aim of eventually having the NN emulate the teacher.
- **Unsupervised learning:** In contrast, there is no expert in unsupervised or self-organized learning, as shown in the Figure (5.6). Suppose the NN is exposed to a training vector drawn from the model. Since the teacher is absent in this procedure, it is not possible to provide the NN with a desired response for that training vector. Instead, a provision that is made to identify a measure of the quality of the representation is required to learn and the free parameters are optimized with respect to that measure [Kohonen, 2001]. After training is over, a grouping of the training inputs presented to the network is achieved, based on the similarity measure imposed by the network.
- **Reinforcement learning:** This is another training procedure that is conducted without an external expert. The learning of an input-output relationship is performed through continued interaction with the environment in order to

minimize a scalar index of performance. In general, a reinforcement learning system is built around a critic that converts a primary reinforcement signal to a heuristic reinforcement signal [Barto et al., 1983]. The goal of such a learning procedure is to minimize a cost function, which is defined as the expectation of the cumulative cost of actions taken over a sequence of steps instead of simply immediate cost. See Haykin [1999] for more detail.

- **Hybrid learning:** Sometimes a purely supervised learning procedure is not very efficient and the incorporation of supervised learning with unsupervised learning to solve certain questions is required. For example, an appropriate unsupervised NN can be applied first to reduce the training data set for a classification problem by clustering original data and a supervised learning NN architecture can be applied to categorize the clustered data. As a result, the training time of the supervised learning can be reduced significantly.

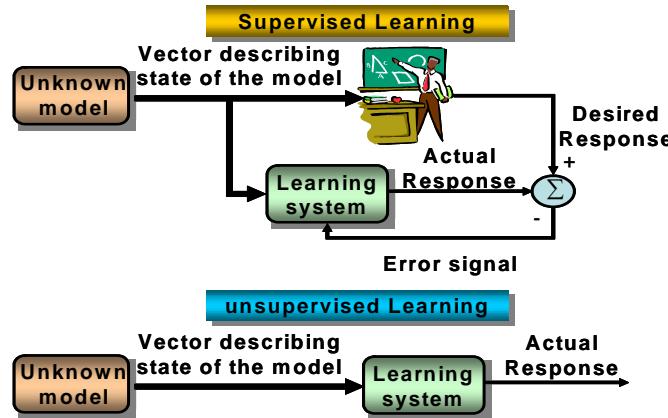


Figure 5.6: Learning procedures

5.2 Multi-Layered Feed-Forward Neural Networks

A supervised neural network that has either a static or dynamic network architecture can be applied for nonlinear input-output mapping applications, such as pattern recognition, function approximation and estimation. A dynamic network architecture is given in section 5.3. According to Ham and Kostanic [2001], four of the most common static

neural networks are: (1) associate memory networks, (2) radial basis function networks, (3) counter-propagation networks, and (4) multi-layered feed-forward networks. The scope of this research is limited to multi-layered feed-forward networks. Therefore, see Ham and Kostanic [2001] for more details associated with other networks.

The multi-layered feed-forward neural network (MFNN) trained by the backpropagation algorithm is the most well-known and most common used neural network today. The standard backpropagation algorithm is based on the gradient descent algorithm and the synaptic weights are updated proportionally to the computed error between the actual response and the desired response. The result after training is a specific nonlinear mapping from input to output. The advantages of the backpropagation learning algorithm include its parallel computation structure and its ability to acquire a complex nonlinear mapping. As it is the most widely applied static neural network, the following section is given to present more detail information about static MFNN and associated learning algorithms.

5.2.1 Nonlinear Mapping and MFNNs

As previously stated, the topology of a MFNN consists of an input layer, at least one hidden layer and an output layer. In general, an individual neuron aggregates its weighted inputs (synaptic operation) and yields outputs through a linear or nonlinear activation function (somatic operation). As these neurons form layered network configurations through only feedforward interlayered synaptic connections in terms of the neural signal flow, it has only feedforward information from the lower to the higher neural layers. As a result, a MFNN is a static neural model in the sense that its input-output relationship can be described by a nonlinear mapping function. In other words, MFNN has the capability of implementing a nonlinear mapping from many inputs to many outputs. Indeed, MFNNs have been widely applied to provide alternative solutions to various engineering and science applications that can not be solved by conventional methods.

Since the complexity of the problem varies from one application to another, the complexity of the applied MFNNs varies according to the complexity of the application. In general, the complexity of MFNN depends on its topology which consists of the

number of the hidden neurons and the number of the hidden layers. In other words, the complexity also implies a computation burden. As a result, a MFNN with optimal topology should be able to provide the best approximation accuracy to the unknown model using the most appropriate number of hidden neurons and hidden layers [Golden, 1996].

There are many ways to decide the most appropriate number of hidden neurons, see Bishop [1995] and Haykin [1999] for details. In general, the common principle indicates that the most appropriate number of hidden neurons is application dependent and can only be decided empirically during the early stages of topology design. It is very common in the design phase of neural networks to train many different candidate networks that have different numbers of hidden neurons and then to select the best, in terms of the performance on an independent validation set. The disadvantages include, first, all of the effort involved in training the remaining networks is wasted. Second, the generalization of performance on the validation set has a random component due to the noise on the data, thus the network that has the best performance on the validation set might not be able to provide the best performance on a new test set. See Bishop [1995] for details.

The problems can be solved by combining the networks together to form a committee. It can be done by taking the output of the committee to be the average of the outputs of the L candidate networks which have different numbers of hidden neurons and comprise the committee. The importance of this approach can lead to significant improvements in the predictions of new data. See Bishop [1995] for more details. Similarly, the most appropriate number of hidden layers can also be decided empirically. However, according to Principe et al., [2000], theoretically, an MFNN is a universal approximator, which can be applied to approximate the unknown model to any degree of accuracy, and that one or two hidden layers are all it takes to reach this arbitrary mapping capability. Consequently, a two layered neural network and the derivation of its associated standard backpropagation learning algorithm is given in this section. Figure (5.7) illustrates a general structure of a two-layered feed-forward neural network [Anderson and McNeill, 1992]. The input layer has n inputs, the hidden layer has p neurons and the output layer

has m neurons. Figure (5.7) also indicates the detailed structure of the hidden and output neurons.

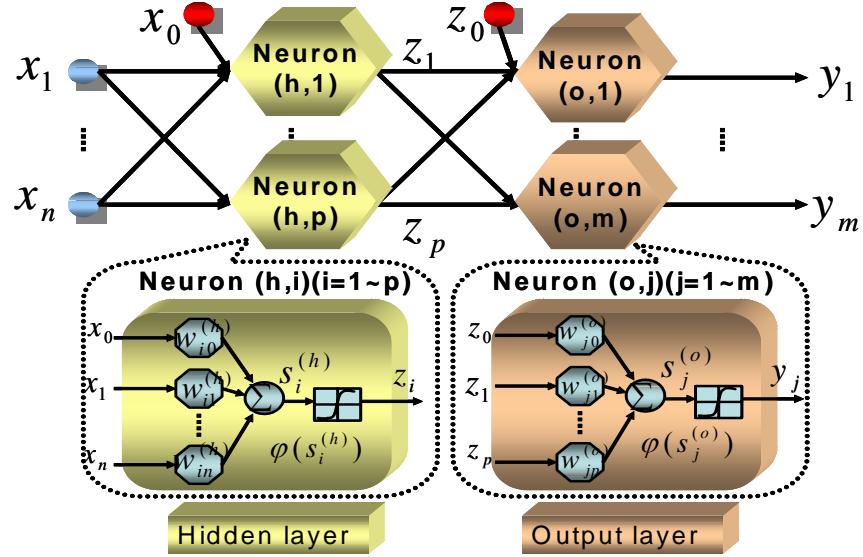


Figure 5.7: A two layered feed-forward network

The hidden neurons, neuron (h, i), ($i=1, 2 \dots p$), receive n input signals (x), and generate p temporary output signals (z), where

$$x = [x_1, x_2, \dots, x_n]^T \quad \text{and} \quad z = [z_1, z_2, \dots, z_p]^T \quad (5.9)$$

The temporary output signals are fed forward to the output neurons. As a result, output neurons, neuron (o, j), ($j=1, 2 \dots m$) generate m output signals (y) thus the network has an output vector

$$y = [y_1, y_2, \dots, y_m]^T \quad (5.10)$$

The synaptic weights associated with the hidden neurons and output neurons are defined as $w_{i1}^{(h)}, w_{i2}^{(h)}, \dots, w_{in}^{(h)}$, ($i=1 \dots p$) and $w_{j1}^{(o)}, w_{j2}^{(o)}, \dots, w_{jp}^{(o)}$, ($j=1 \dots m$). The vector expressions can be given as

$$w_i^{(h)} = [w_{i1}^{(h)} \quad \dots \quad w_{in}^{(h)}]^T, i=1,2...,p \quad (5.11)$$

$$w_j^{(o)} = [w_{j1}^{(o)} \quad \dots \quad w_{jp}^{(o)}]^T, j=1,2...,m \quad (5.12)$$

Consequently, the weight matrices can be given as

$$W^{(h)} = [w_1^{(h)} \dots w_p^{(h)}]^T = \begin{bmatrix} w_{11}^{(h)} & \dots & w_{1n}^{(h)} \\ \vdots & \ddots & \vdots \\ w_{p1}^{(h)} & \dots & w_{pn}^{(h)} \end{bmatrix} \quad (5.13)$$

$$W^{(o)} = [w_1^{(o)} \dots w_m^{(o)}]^T = \begin{bmatrix} w_{11}^{(o)} & \dots & w_{1p}^{(o)} \\ \vdots & \ddots & \vdots \\ w_{m1}^{(o)} & \dots & w_{mp}^{(o)} \end{bmatrix} \quad (5.14)$$

Where the i th row element of $W^{(h)}$ are obviously associated with neuron (h, i) in the hidden layer, whereas the j th row element of $W^{(o)}$ corresponds to neuron (o, j) in the output layer. The input-output equations can be written as:

$$\text{hidden} \left\{ \begin{array}{l} s_i^{(h)} = \sum_{k=1}^n w_{ik}^{(h)} x_k, i=1 \sim p \\ z_i = \varphi(s_i^{(h)}) \end{array} \right. \quad (5.15)$$

$$\text{output} \left\{ \begin{array}{l} s_j^{(o)} = \sum_{q=1}^p w_{jq}^{(o)} z_q, j=1 \sim m \\ y_j = \varphi(s_j^{(o)}) \end{array} \right. \quad (5.16)$$

According to Ham and Kostanic [1999] and Haykin [1999], the augmented expressions of the neural inputs and weights including the bias and its associated weights can be given as follows:

$$\begin{aligned} x_0, w_{i0}^{(h)} &= 1, \\ z_0, w_{j0}^{(o)} &= 1 \end{aligned} \quad (5.17)$$

As a result, the augmented version of the input and weight vectors of neuron (h, i) and neuron (o, j) can be given as:

$$\begin{aligned} x_a &= [x_0 \quad x_1 \quad \dots \quad x_n]^T \\ w_{ai}^{(h)} &= [w_{i0}^{(h)} \quad w_{i1}^{(h)} \quad \dots \quad w_{in}^{(h)}]^T, i = 1, 2, \dots, p \\ z_a &= [z_0 \quad z_1 \quad \dots \quad z_p]^T \\ w_{aj}^{(o)} &= [w_{j0}^{(o)} \quad w_{j1}^{(o)} \quad \dots \quad w_{jp}^{(o)}]^T, j = 1, 2, \dots, m \end{aligned} \quad (5.18)$$

Similarly, the augmented weights matrices can be written as:

$$W_a^{(h)} = [w_{a1}^{(h)} \dots w_{ap}^{(h)}]^T = \begin{bmatrix} w_{10}^{(h)} & \dots & w_{1n}^{(h)} \\ \vdots & \ddots & \vdots \\ w_{p0}^{(h)} & \dots & w_{pn}^{(h)} \end{bmatrix} \quad (5.19)$$

$$W_a^{(o)} = [w_{a1}^{(o)} \dots w_{am}^{(o)}]^T = \begin{bmatrix} w_{10}^{(o)} & \dots & w_{1p}^{(o)} \\ \vdots & \ddots & \vdots \\ w_{m0}^{(o)} & \dots & w_{mp}^{(o)} \end{bmatrix} \quad (5.20)$$

For an input signal $x \in R^n$, the MFNN produces a response signal $y \in R^m$ through complex nonlinear operations. In fact, the MFNN generates a nonlinear mapping process from the n-dimensional input signal space to the m-dimensional output signal space. The analytical expression of this can be given as follows:

$$s^{(h)} = \begin{bmatrix} s_1^{(h)} \\ \vdots \\ s_p^{(h)} \end{bmatrix}, \quad s^{(o)} = \begin{bmatrix} s_1^{(o)} \\ \vdots \\ s_m^{(o)} \end{bmatrix} \quad (5.21)$$

The augmented version of the input-output equations can be obtained

$$\text{hidden} \begin{cases} s^{(h)} = W_a^{(h)} x_a \\ z_a = \varphi(s^{(h)}) \end{cases} \quad (5.22)$$

$$output \begin{cases} s^{(o)} = W_a^{(o)} z_a \\ y = \varphi(s^{(o)}) \end{cases} \quad (5.23)$$

The $\varphi(\cdot)$ is a nonlinear activation function. Thus, the nonlinear mapping function is given as

$$y = \varphi(W_a^{(o)} \varphi(W_a^{(h)} x_a)) \quad (5.24)$$

Figure (5.8) illustrates the nonlinear input/output mapping using a two layered feed-forward network.

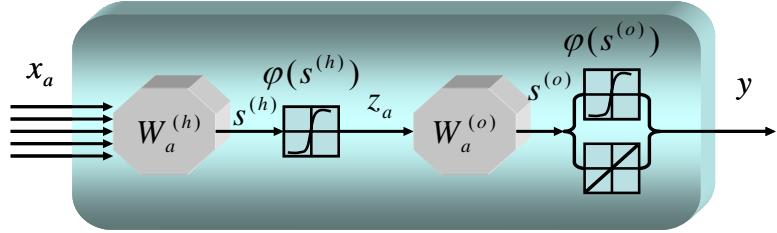


Figure 5.8: Nonlinear mapping

5.2.2 Standard Backpropagation Learning Algorithm

In the following derivation, the network inputs and outputs are considered in a discrete time domain. Given a desired response vector $d(k) = [d_1(k) \ d_2(k) \ \dots \ d_m(k)]$, Rumerhart and MaClelland [1986] proposed an adaptive weight learning algorithm that is characterized by the generalized delta rule. It performs an optimal tuning process of synaptic weights thus each output error vector, defined as the difference between the desired response, $d(k)$, and the actual network output, $y(k)$, is minimized. As indicated in Equation 5.25, an instantaneous error function for the network output is given as the sum of the squares of the errors (SSE) for all the output neurons [Haykin, 1999].

$$E_{SSE} = \frac{1}{2} \sum_{j=1}^m |d_j(k) - y_j(k)|^2 = \frac{1}{2} \sum_{j=1}^m e_j^2(k) \quad (5.25)$$

Such an error function can also be defined using the mean of the squares of the errors (MSE), which is given as follows [Anderson and McNeill, 1992].

$$E_{MSE} = \frac{1}{2m} \sum_{j=1}^m |d_j(k) - y_j(k)|^2 = \frac{1}{2m} \sum_{j=1}^m e_j^2(k) \quad (5.26)$$

The factor $\frac{1}{2}$ is used for convenience in calculating the derivatives. The error e_j indicates the error between the jth desired response and the jth output response at the neuron (o, j), and is given as

$$e_j = (d_j - y_j) \quad (5.27)$$

In fact, E can be regarded as the cost function and the index of the learning performance of the network. It is a function of all the synaptic weights and activation functions. Applying the gradient descent (steepest descent) algorithm for minimizing the cost function [Ham and Kostanic, 2001], the correction of the synaptic weights are made in the direction of decreasing error functions and are proportional to the negative gradients of the error function with respect to the weights;

$$\Delta w_{ai}^{(h)} = -\eta \nabla w_{ai}^{(h)}(E) = -\eta \frac{\partial E}{\partial w_{ai}^{(h)}}, i = 1, 2, \dots, p \quad (5.28)$$

$$\Delta w_{aj}^{(o)} = -\eta \nabla w_{aj}^{(o)}(E) = -\eta \frac{\partial E}{\partial w_{aj}^{(o)}}, j = 1, 2, \dots, m \quad (5.29)$$

Where $0 < \eta < 1$ is a learning rate parameter and the choice of such a parameter will affect the convergence speed of the updating process. Equations 5.28 and 5.29 indicate the magnitude, and the directions to the change of the synaptic weights are determined by the gradient of E. The partial derivatives of E with respect to the synaptic operation, δ_s , are defined as

$$\delta_i^{(h)} = -\frac{\partial E}{\partial s_i^{(h)}}, i = 1, 2, \dots, p, \quad [\text{neuron}(h, i)] \quad (5.30)$$

$$\delta_i^{(o)} = -\frac{\partial E}{\partial s_j^{(o)}}, j = 1, 2, \dots, m, \quad [\text{neuron}(o, j)] \quad (5.31)$$

Applying Equation 5.31, the delta variable of the output neuron, $\delta_j^{(o)}$, can be obtained by expanding the right hand side of Equation 5.31 by Equations 5.25 and 5.16

$$\delta_j^{(o)} = -\frac{1}{2} \frac{\partial \sum_{l=1}^m (d_l - y_l)^2}{\partial s_j^{(o)}} = -\frac{1}{2} \frac{\partial \sum_{l=1}^m (d_l - \varphi(s_l^{(o)}))^2}{\partial s_j^{(o)}} = -\frac{1}{2} \sum_{l=1}^m \frac{\partial (d_l - \varphi(s_l^{(o)}))^2}{\partial s_j^{(o)}} \quad (5.32)$$

$s_l^{(o)}$, $s_j^{(o)}$ and d_l are independent as $l \neq j$, thus

$$\delta_j^{(o)} = (d_j - \varphi(s_j^{(o)})) \frac{\partial \varphi(s_j^{(o)})}{\partial s_j^{(o)}} = e_j \varphi'(s_j^{(o)}) \equiv e_j^{(o)} \varphi'(s_j^{(o)}) \quad (5.33)$$

Equation 5.33 indicates that the delta associated with the output neuron, $\delta_j^{(o)}$, can be regarded as the product of the error at the output neuron and the differential signal of the nonlinear activation. The gradient of the cost function E with respect to the synaptic weight $w_{aj}^{(o)}$ is then evaluated to obtain the weight updating algorithm for output neurons.

$$\nabla w_{aj}^{(o)}(W) = \frac{\partial E}{\partial w_{aj}^{(o)}} \quad (5.34)$$

Applying the chain rule

$$\nabla w_{aj}^{(o)}(E) = \sum_{l=1}^m \frac{\partial E}{\partial s_l^{(o)}} \frac{\partial s_l^{(o)}}{\partial w_{aj}^{(o)}} \quad (5.35)$$

Since $s_l^{(o)} (l \neq j)$ is independent of $w_{aj}^{(o)}$, thus

$$\nabla_{w_{aj}^{(o)}}(E) = \frac{\partial E}{\partial s_j^{(o)}} \frac{\partial s_j^{(o)}}{\partial w_{aj}^{(o)}} \quad (5.36)$$

Since $s_j^{(o)} = (w_{aj}^{(o)})^T z_a$, and taking the partial derivatives of $s_l^{(o)}$ with respect to $w_{aj}^{(o)}$ yields

$$\frac{\partial s_j^{(o)}}{\partial w_{aj}^{(o)}} = z_a \quad (5.37)$$

Applying Equations 5.31, 5.33, 5.35 and 5.36 yields

$$\nabla_{w_{aj}^{(o)}}(E) = -\delta_j^{(o)} z_a = -e_j \varphi'(s_j^{(o)}) z_a \quad (5.38)$$

Applying the gradient descent algorithm, the weight updating algorithm for the nonlinear output neurons can be given as follows [Haykin, 1999]

$$\begin{aligned} w_{aj}^{(o)}(k+1) &= w_{aj}^{(o)}(k) - \eta \nabla_{w_{aj}^{(o)}}(E(k)) \\ &= w_{aj}^{(o)}(k) + \eta \delta_j^{(o)}(k) z_a(k) \\ &= w_{aj}^{(o)}(k) + \eta e_j(k) \varphi'(s_j^{(o)}(k)) z_a(k) \end{aligned} \quad (5.39)$$

Sometimes the linear activation function can be applied for the output neurons for nonlinear mappings as well, thus Equation 5.33 can be rewritten as

$$\delta_j^{(o)} = e_j \varphi'(s_j^{(o)}) \equiv e_j \quad (5.40)$$

As a result, the weight updating algorithm for the linear output neuron is given by simplifying Equation 5.39

$$w_{aj}^{(o)}(k+1) = w_{aj}^{(o)}(k) + \eta e_j(k) z_a(k) \quad (5.41)$$

Similarly, to obtain the weight updating algorithm for the hidden neurons, the delta associated with the hidden neuron, $\delta_j^{(h)}$ can be given using chain rule as well as Equations 5.30 and 5.31

$$\delta_i^{(h)} = -\sum_{l=1}^m \frac{\partial E}{\partial s_l^{(o)}} \frac{\partial s_l^{(o)}}{\partial s_i^{(h)}} = \sum_{l=1}^m \delta_l^{(o)} \frac{\partial s_l^{(o)}}{\partial s_i^{(h)}} \quad (5.42)$$

Applying Equations 5.15 and 5.16

$$\frac{\partial s_l^{(o)}}{\partial s_i^{(h)}} = \sum_{q=0}^p \frac{\partial (w_{lq}^{(o)} \varphi(s_q^{(h)}))}{\partial s_i^{(h)}} = w_{li}^{(o)} \varphi'(s_i^{(h)}) \quad (5.43)$$

Applying Equations 5.42 and 5.43 yields

$$\delta_i^{(h)} = \sum_{l=1}^m \delta_l^{(o)} w_{li}^{(o)} \varphi'(s_i^{(h)}) \quad (5.44)$$

The propagation error that represents the error in the hidden layer due to all the output error can be defined as

$$e_i^{(h)} = \sum_{l=1}^m \delta_l^{(o)} w_{li}^{(o)} \quad (5.45)$$

Thus

$$\delta_i^{(h)} = e_i^{(h)} \varphi'(s_i^{(h)}) \quad (5.46)$$

Equations 5.45 and 5.46 indicate that each output error generated at the output neurons has its influence on the $\delta_i^{(h)}$. In fact, the way that error signal propagates backward from the output neurons to the hidden neurons is similar to the way the input signal transmits forward from hidden neurons to the output neurons. Applying the chain rule, the gradient of E with respect to the hidden synaptic weights can be obtained

$$\nabla_{w_{ai}^{(h)}}(E) = \frac{\partial E}{\partial w_{ai}^{(h)}} = \sum_{j=1}^m \sum_{q=1}^p \frac{\partial E}{\partial s_j^{(o)}} \frac{\partial s_j^{(o)}}{\partial s_q^{(h)}} \frac{\partial s_q^{(h)}}{\partial w_{ai}^{(h)}} = \sum_{q=1}^p \frac{\partial E}{\partial s_q^{(h)}} \frac{\partial s_q^{(h)}}{\partial w_{ai}^{(h)}} \quad (5.47)$$

Applying Equation 5.30 yields

$$\nabla_{w_{ai}^{(h)}}(E) = - \sum_{q=1}^p \delta_q^{(h)} \frac{\partial s_q^{(h)}}{\partial w_{ai}^{(h)}} \quad (5.48)$$

Applying Equation 5.15, $s_q^{(h)} (q \neq j)$ is independent of $w_{ai}^{(h)}$, thus

$$\frac{\partial s_q^{(h)}}{\partial w_{ai}^{(h)}} = \begin{cases} 0, & q \neq i \\ x_a, & q = i \end{cases} \quad (5.49)$$

Consequently,

$$\nabla_{w_{ai}^{(h)}} = -\delta_i^{(h)} x_a \quad (5.50)$$

Applying Equations 5.46 and 5.50 yields

$$\nabla_{w_{ai}^{(h)}} = -e_i^{(h)} \varphi'(s_i^{(h)}) x_a \quad (5.51)$$

Again, using the gradient descent algorithm to obtain the weight updating algorithm for the hidden neurons [Haykin, 1999],

$$\begin{aligned} w_{ai}^{(h)}(k+1) &= w_{ai}^{(h)}(k) - \eta \nabla_{w_{ai}^{(h)}}(E(k)) \\ &= w_{ai}^{(h)}(k) + \eta \delta_i^{(h)}(k) x_a(k) \\ &= w_{ai}^{(h)}(k) + \eta e_i^{(h)}(k) \varphi'(s_i^{(h)}(k)) x_a(k) \end{aligned} \quad (5.52)$$

Both Equations 5.39 and 5.52 are called the **generalized delta rule** [Haykin, 1999 and Ham and Kostanic, 2001]. However, from the error signal propagation perspective, it is also known as the standard backpropagation algorithm. Figure (5.9) illustrates the learning flow of the backpropagation algorithm for a two layered feed-forward network.

The popularity of using gradient descent algorithms as the standard weight updating algorithm for backpropagation algorithm is based on its simplicity instead of its search power. In fact, the gradient descent algorithm is an inefficient procedure as it is sensitive to the selection of the learning rate parameter η . If η is too large, the algorithm may overshoot leading to divergent oscillations and result in a complete breakdown of the algorithm. On the contrary, if η is too small, the search for the minimum can be extremely slow [Bishop, 1995].

Furthermore, since the search of gradient descent is based only on the local gradient information it can become trapped in local minima as the goal is to search global minima. In terms of local curvature, a local minima and global minima are identical, thus the gradient descent will be trapped in any local concavity of the error surface. It will move very slowly, when the search traverses a flat region of the error surface because the weights are modified proportionally to the gradient. If the gradient is small, the weight updates will be small thus many iteration steps are required to move through the flat spot.

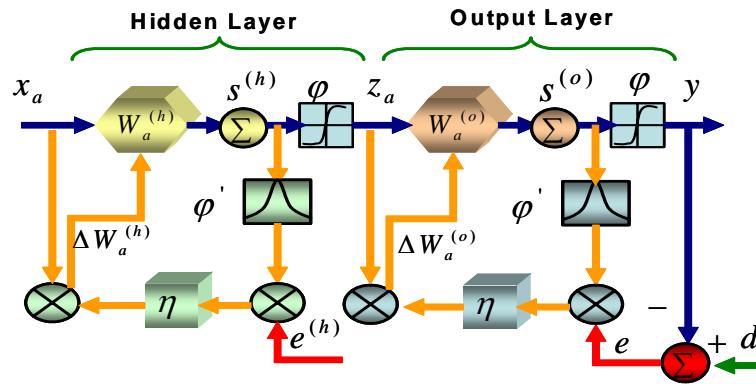


Figure 5.9: Learning flow of standard backpropagation algorithm

Consequently, the slow convergence speed of the standard backpropagation algorithm has encouraged research in faster algorithms for MFNN training [Anderson and McNeill, 1992]. In general, research on faster learning algorithms roughly fall into two categories. The first category consists of introducing the improvement to the standard backpropagation algorithm, such as the introduction of momentum to the gradient descent, which is given in this section. The second category involves the use of standard

numerical optimization techniques, such as second order optimization learning algorithms, which are given briefly in Appendix B.2.

In order to avoid oscillations due to large η for rapid learning and the acceleration of the learning speed, a modified version of the standard backpropagation is obtained by introducing an additional parameter α , $0 \leq \alpha \leq 1$, called the momentum constant, in the second term on the right side of Equations 5.39 and 5.52 [Haykin, 1999].

$$w_{aj}^{(o)}(k+1) = w_{aj}^{(o)}(k) + \eta \delta_j^{(o)}(k) z_a(k) + \alpha \Delta w_{aj}^{(0)}(k-1) \quad (5.53)$$

$$w_{ai}^{(h)}(k+1) = w_{ai}^{(h)}(k) + \eta \delta_i^{(h)}(k) x_a(k) + \alpha \Delta w_{ai}^{(h)}(k-1) \quad (5.54)$$

Equations 5.53 and 5.54 are derived for a two layered feed-forward network. Combing both of these equations, a general gradient descent plus momentum algorithm can be obtained

$$\begin{aligned} w_{aj}^{(i)}(k+1) &= w_{aj}^{(i)}(k) + \eta \delta_j^{(i)}(k) x_a^{(i-1)}(k) + \alpha \Delta w_{aj}^{(i)}(k-1) \\ \Delta w_{aj}^{(i)}(k) &= \eta \delta_j^{(i)}(k) x_a^{(i-1)}(k) + \alpha \Delta w_{aj}^{(i)}(k-1) \\ i &= 1, 2, \dots, M \\ j &= 1, 2, \dots, n_i \end{aligned} \quad (5.55)$$

Where M is the number of the hidden layers plus one output layer and the n is the total number of neuron in the i th layer. Thus, if the network is operating in the flat area of the error surface, the value of the gradient does not change substantially from each step. The second term of Equation 5.55 can be approximated as

$$\begin{aligned} \Delta w_{aj}^{(i)}(k) &\approx \eta \delta_j^{(i)}(k) x_a^{(i-1)}(k) + \eta \alpha \delta_j^{(i)}(k) x_a^{(i-1)}(k) + \eta \alpha^2 \delta_j^{(i)}(k) x_a^{(i-1)}(k) + \dots \\ &= \eta (1 + \alpha + \alpha^2 + \dots) \delta_j^{(i)}(k) x_a^{(i-1)}(k) \\ &\approx \frac{\eta}{1 - \alpha} \delta_j^{(i)}(k) x_a^{(i-1)}(k) \end{aligned} \quad (5.56)$$

As α is always smaller than unity, thus updating with momentum increases the effective learning rate to $\frac{\eta}{1-\alpha}$. Equation 5.55 indicates that the weights are updated proportionally to how much they were updated in the last iteration. Thus, if the search is going down the hill and finds a flat region, the weights are still changed, not because of the gradient (which is practically zero in the flat region) but because of the rate of the change in the weights. Likewise in a narrow valley, where the gradient tends to bounce back and fourth between hillsides, the momentum stabilizes the search as it tends to make the weights follow a smoother path, as shown in Figure (5.10).

As illustrated in Figure (5.10), imagining a ball (weight vector position) rolling down a hill (error surface), if the ball reaches a small flat part of the hill it will continue past this local minima because of its momentum. In contrast, a ball without momentum, however, will get stuck in this valley. As a result, momentum learning can provide faster and more stable performance than the basic gradient descent algorithm.

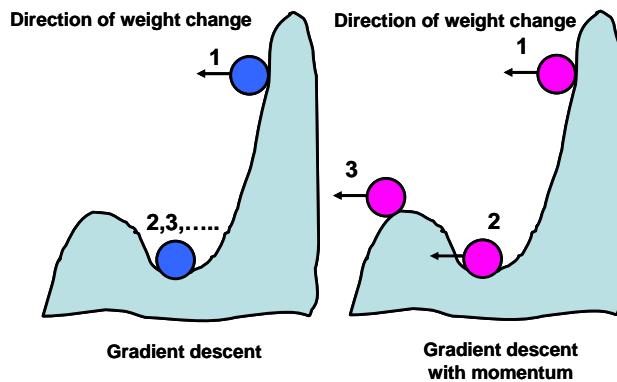


Figure 5.10: The impact of momentum learning (After Principe et al., 1999)

As mentioned previously, the standard backpropagation algorithm (gradient descent algorithm) and the modified standard backpropagation algorithm (gradient descent algorithm plus momentum algorithm) employ only the first order partial derivatives of the error function (gradient). Such first order learning algorithms have weaknesses when dealing with most of the classification and approximation problems. They usually require large numbers of iterations to optimally tune the synaptic weights thus it is not possible to

adapt them for on-line application, such as adaptive control. As a result, recent development of backpropagation algorithms utilizes the high order optimization algorithms to speed up the learning speed. Appendix B.2 provides more details on the advantages and disadvantages of different second order optimization algorithms. In addition, a simple example is given in Appendix B.2 to compare the performance of the gradient descent algorithm (1st order), Quasi-Newton algorithm (2nd order), conjugate gradient algorithm (2nd order), and Levenberg-Marquardt algorithm (2nd order).

As indicated in Appendix B.2, the Levenberg-Marquardt algorithm is the fastest algorithm among those candidates. However, it requires large storage space. Thus, the choice between the conjugate gradient algorithm and Levenberg-Marquardt algorithm is application dependent. If the computation storage space is a critical issue, then the conjugate gradient algorithm is an appropriate option, otherwise, the Levenberg-Marquardt algorithm is the best choice in terms of the speed of convergence.

In addition, the least square (LS) learning algorithm proposed by Douglas and Meng [1991] and an extended Kalman filter (EKF) based learning algorithm proposed by Singhal and Wu [1989] are given briefly in Appendix B.3. As indicated in Douglas and Meng [1991] and Rivals and personnaz [1998], Both LRLSL and EKF learning algorithms are able to provide better performance in terms of convergence speed. However, it comes with the price as the complexity of the computation model is more complicated and storage requirements are higher than any of those 1st and 2nd order learning algorithms mentioned in Appendix B.2. The weight updating requires a centralized computing facility, thus the advantage of the parallel computation provided by ANNs is not exploited. Practically, the EKF and LRLSL learning algorithms are not recommended due to their computation complexity and storage requirement [Haykin, 2001]. However, if the requirement of the computation speed is critical and the limitation of the storage space is not an issue, the EKF is then suggested.

To overcome the difficulties associated with the computation complexity and storage space, some decoupled versions of the EKF learning algorithms , known as the decoupled extended Kalman filter (DEKF) can be applied [Haykin,2001]. Since the

dimension of the synaptic weights is very high, the DEKF learning algorithm can be applied to reduce the computational complexity and storage requirement by decomposing the weight vector into several sub vectors. There are also more sophisticated algorithms such as the neuron-decoupled EKF (NDEKF) and weight decoupled EKF (WDEKF). The detailed description of the DEKF, NDEKF, and WDEKF is far beyond the scope of this research but can be obtained from Haykin [2001].

To give an example about the performance between the second order learning algorithms (i.e., conjugate gradient and Levenberg-Marquardt algorithms) and linearized recursive estimation learning algorithms (i.e., EKF learning algorithm), a simple linear regression problem is applied. The true function (f) is given as

$$f = \sin(2 * \pi * x), x = 0 \sim 1, \Delta x = 1/(n-1), n = 1000; \quad (5.57)$$

Where n is the total number of points. The desired output (t) for the training is generated by adding the random noise to the function,

$$t = \sin(2 * \pi * x) + noise \quad (5.58)$$

The objective is to evaluate the accuracy of the regression by comparing the prediction output of the trained MFNN (y) and ideal function output (f) and the required training time for the conjugate gradient (CG) learning algorithm, Levenberg-Marquardt (LM) learning algorithm and EKF learning algorithm. The training procedure for each algorithm was repeated for five trials. Figure (5.11) illustrates the function output, desired output and MFNN outputs through the use of different learning algorithms during one of the five trials. A two layered feedforward network with 10 hidden neurons was designed to solve the regression problem.

The desired output (t) for each trail was generated independently by adding different random noise to the function. Thus, the RMS errors that can be obtained by comparing the actual MFNN prediction output and the function output and required training time for

each learning algorithm during five different trials are listed in Table 5.6. As indicated in Table 5.6, the EKF learning algorithm was able to provide the best performance in the time domain. It converged 10 times and 100 times faster than the LM and CG algorithms, respectively. However, the performance of the EKF learning algorithm in the prediction accuracy domain was 5 times worse than the LM and CG learning algorithms.

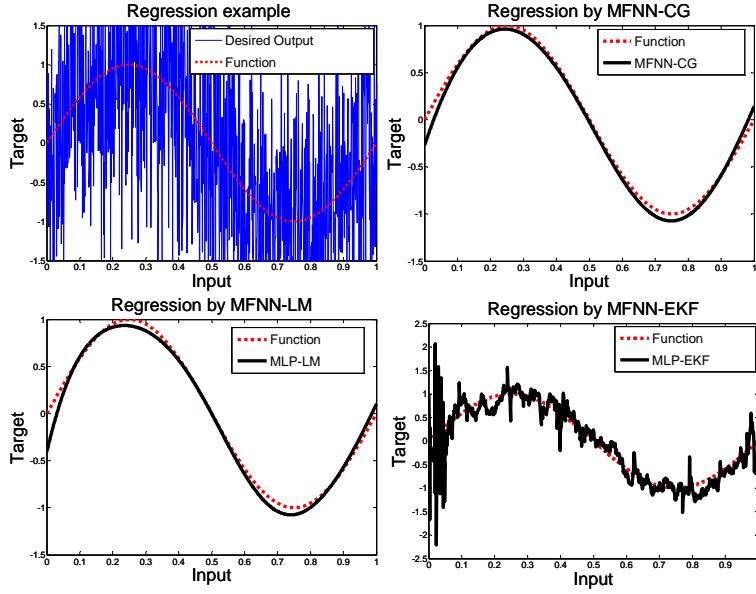


Figure 5.11: Regression problem solved by MFNN with different learning algorithms

Generally speaking, the choice between these learning algorithms is made based on the balance between the computation complexity, storage space requirement, accuracy and training speed. Indeed, the EKF learning algorithm is the fastest algorithm among those candidates, however, the prediction accuracy of the EKF trained MFNN is not as good as other learning algorithms. Thus the EKF learning is not recommended for the development of the conceptual intelligent navigator, On the contrary, the LM algorithm, although possessing the disadvantage of storage space requirement, is recommended due to its reasonable training speed and prediction accuracy.

Table 5.6: Performance summary

Learning algorithm	Index	1	2	3	4	5
MFNN-CG	Time(second)	<i>10.16</i>	<i>10.45</i>	<i>9.42</i>	<i>9.45</i>	<i>10.19</i>
	RMS	<i>0.046</i>	<i>0.045</i>	<i>0.046</i>	<i>0.044</i>	<i>0.046</i>
MFNN-LM	Time(second)	<i>3.95</i>	<i>3.93</i>	<i>3.91</i>	<i>3.89</i>	<i>3.84</i>
	RMS	<i>0.049</i>	<i>0.047</i>	<i>0.048</i>	<i>0.048</i>	<i>0.047</i>
MFNN-EKF	Time(second)	<i>0.27</i>	<i>0.25</i>	<i>0.23</i>	<i>0.20</i>	<i>0.20</i>
	RMS	<i>0.19</i>	<i>0.18</i>	<i>0.17</i>	<i>0.18</i>	<i>0.18</i>

5.3 Recurrent Neural Networks

As mentioned previously, a dynamic neural network can be defined as a neural network that consists of inter-layer feedback loops (i.e., from output layer to input layer) and intra-layer feedback loops (i.e., between different neurons within the same layer) or self-feedback loops. Recurrent neural networks can be characterized as dynamic neural networks as they usually have at least one of those feedback loops mentioned above [Principe et al., 1999].

From the computational perspective, a dynamic neural network that contains a feedback loop may provide more computational advantages than a static neural network, which only contains feed-forward architecture [Mandic and Chambers, 2001]. In fact, applications of ANNs in forecasting, signal processing, and control require the treatment of dynamics associated with the unknown model. Feed-forward networks for processing of dynamical system tend to capture the dynamics by including past input in the input vector. However, for dynamical modeling of complex systems, there is a need to involve feedback, to use recurrent networks. Thus, the nonlinear dynamic recurrent neural network architecture is particularly appropriate for system identification, control and filtering application because of its distributed information processing ability as in biological neural systems. In fact, various recurrent neural network architectures have been introduced for learning, information storing and using knowledge that might be found widely in the brain [Norgarrd, et al., 2001].

One of the earliest recurrent networks reported in the literature is the auto-associator independently described by Anderson et al., [1977] and Kohonen [1977]. It contains a

pool of neurons with connections between each unit i and $j, i \neq j$, as shown in Figure (5.12). Hopfield [1982] suggested the famous Hopfield networks that are a special kind of recurrent neural networks that can be used as associative memory. Associative memory is memory that is addressed through its contents. That is, if a pattern is presented to an associative memory, it returns whether this pattern coincides with a stored pattern. An associative memory may also return a stored pattern that is similar to the presented one, so that noisy input can also be recognized.

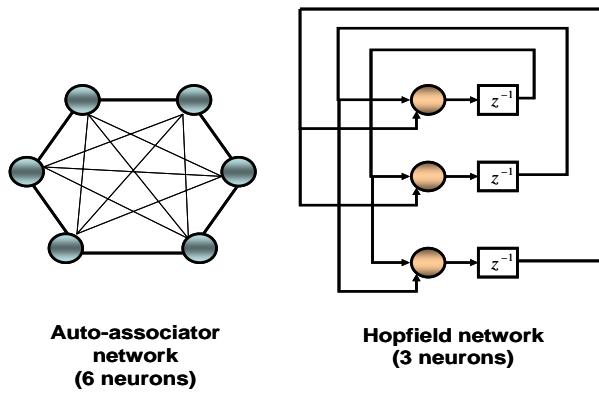


Figure 5.12: Auto-associator and Hopfield network

The Hopfield network consists of a set of neurons and a corresponding set of unit delays, forming a multi-loop feedback system, as shown in Figure (5.12). The number of feedback loops is equal to the number of neurons. Basically, the output of each neuron is fed back; via a unit delay, to each of the other neurons in the network. In other words, there is self feedback in the network. Hopfield networks are used as associative memory by exploiting the property that they possess stable states, one of which is reached by carrying out the normal computations of a Hopfield network. If the connection weights of the network are determined in such a way that the patterns to be stored become the stable states of the network, a Hopfield network produces for any input pattern a similar stored pattern. Thus noisy patterns can be corrected or distorted patterns can still be recognized.

Jordan [1986] suggested a recurrent network architecture, known as the Jordan network, for control of robots, as shown in the Figure (5.13). In the Jordan network, the network outputs are fed back into the input layer through a set of extra input units called the

context layer. There are as many units as the output neurons. The connections between output neuron and context units have a fixed weight of +1; learning takes place in the connection between the input and hidden neurons as well as hidden and output neurons. Elman [1990] proposed a Simple Recurrent Network (SRN), also known as Elman network, for problems in linguistics. In addition, Aguilar et al., [2003] applied Elman network for the prediction of pitch and head movements using a MEMS IMU and it concluded that the Elman network can do a very good job of learning the head motion, thus only the topology and associated learning algorithm of Elman networks is given in the next section.

Similar to a Jordan network, the Elman network also has a context layer, which contains extra input units that receive the outputs from the hidden neurons, as shown in the Figure (5.13). This is the main difference between Jordan and Elman networks. All the learning algorithms derived for the MFNN can be applied to train both Jordan and Elman networks [Principe et. al., 1999].

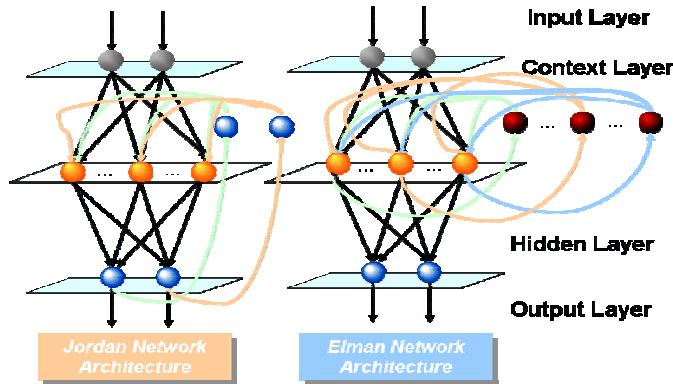


Figure 5.13: Jordan and Elman network

5.3.1 Elman Networks

As indicated in Figure (5.14), the architecture of the Elman network is very similar to the architecture of the MFNN with a single hidden layer. It consists of input nodes where the inputs from the environment are applied, hidden neurons where internal representations of the nonlinear input-output relationship are formed and the output neurons where the

outputs of the network are produced. In addition, the Elman network also consists of another layer of neurons, called context neurons that receive the output of hidden neurons at previous time instants. The input neurons and the context neurons are connected with feed-forward connections to the hidden neurons.

Hence, the hidden neurons of the Elman network do not only learn useful representations of the current input to the network; they also develop representations that are useful for encoding the temporal properties of the input patterns. Since the feedback synaptic weight links from the hidden neurons to the context neurons are fixed (normally taken to be equal to 1), the only adaptive weight links are feed-forward type. Hence, the three phases of the backpropagation learning algorithm (feed-forward phase, backpropagation phase and synaptic weights updating phase) can also be applied to the Elman network to teach the network to learn the optimal synaptic weights required to perform a desired task [Christodoulou and Georgopoulos, 2001].

5.3.2 Standard Backpropagation Learning algorithm.

For illustrating the standard backpropagation learning algorithm for the Elman network, a detailed architecture of the Elman network that contains n input neurons, p nonlinear hidden and context neurons and m linear output neurons is given as Figure (5.14).

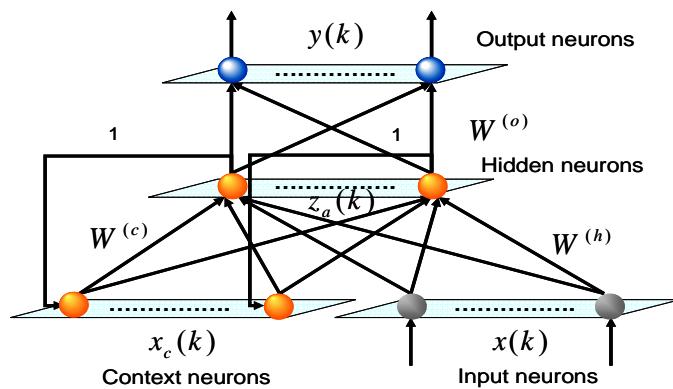


Figure 5.14: Elman network

The synaptic weight links between context layer and hidden layer, $W_a^{(c)}$, is a $p \times p+1$ matrix. Similarly, the synaptic weight between hidden layer and input layer, $W_a^{(h)}$, and the synaptic weight between hidden layer and output layer, $W_a^{(o)}$, are $p \times n+1$ and $m \times p+1$ matrices, respectively. Since the standard backpropagation algorithm developed for MFNN can be applied with only minor modification, the detailed derivation can be found at section 5.2.2 or See Shi et al., [2001] for a similar derivation of the backpropagation algorithm for the Elman network. The associated equations are given in brief according to the three phases of the standard backpropagation algorithm.

■ Feed-forward phase

Similar to the augmented version of the expression used in the previous section, the augmented expression for the feed-forward can be given as

$$\begin{aligned} x_a &= [x_0 \quad x_1 \quad \dots \quad x_n]^T, \\ w_{ai}^{(h)} &= [w_{i0}^{(h)} \quad w_{i1}^{(h)} \quad \dots \quad w_{in}^{(h)}]^T, i = 1, 2, \dots, p, \\ w_{ai}^{(c)} &= [w_{i0}^{(c)} \quad w_{i1}^{(c)} \quad \dots \quad w_{ip}^{(c)}]^T, i = 1, 2, \dots, p, \\ z_a &= [z_0 \quad z_1 \quad \dots \quad z_p]^T, \\ w_{aj}^{(o)} &= [w_{j0}^{(o)} \quad w_{j1}^{(o)} \quad \dots \quad w_{jp}^{(o)}]^T, j = 1, 2, \dots, m \end{aligned} \quad (5.59)$$

$$s^{(h)}(k) = W_a^{(c)}(k)x_c(k) + W_a^{(h)}(k)x_a(k) \quad (5.60)$$

$$z_a(k) = \varphi(s^{(h)}(k)), (\text{hidden layer}) \quad (5.61)$$

$$x_c(k) = z_a(k-1), (\text{context layer}) \quad (5.62)$$

$$y(k) = W_a^{(o)}(k)z_a(k), (\text{output layer}) \quad (5.63)$$

The activation functions φ of hidden neurons and output neurons are nonlinear (i.e., sigmoid function) and linear, respectively.

■ Backpropagation phase

Applying Equation 5.25, and taking the partial derivatives of E with respect to the augmented synaptic weights, $W_a^{(c)}$, $W_a^{(h)}$ and $W_a^{(o)}$ thus the delta for the output and hidden neurons can be given as

$$\delta_i^{(o)} = (d_i - y_i) = e_i^{(o)} = e_i, i = 1, 2, \dots, m \quad (5.64)$$

$$\delta_j^{(h)} = \sum_{i=1}^m (\delta_i^{(o)} w_{ij}^{(o)}) \varphi'(s_j^{(h)}) = e_j^{(h)} \varphi'(s_j^{(h)}), j = 1, 2, \dots, p \quad (5.65)$$

■ Synaptic weights updating phase

Applying the gradient descent algorithm yields the synaptic weights updating algorithm for the Elman network.

For output neurons:

$$w_{ij}^{(o)}(k+1) = w_{ij}^{(o)}(k) + \eta e_i(k) z_{aj}(k), i = 1, 2, \dots, m; j = 0, 1, \dots, p \quad (5.66)$$

For hidden neurons:

$$w_{jq}^{(h)}(k+1) = w_{jq}^{(h)}(k) + \eta e_j^{(h)}(k) \varphi'(s_j^{(h)}(k)) x_q(k), j = 1, 2, \dots, p; q = 0, 1, \dots, n \quad (5.67)$$

For context neurons:

$$w_{jl}^{(c)}(k+1) = w_{jl}^{(c)}(k) + \eta e_j^{(h)}(k) \varphi'(s_j^{(h)}(k)) z_{al}(k-1), j = 1, 2, \dots, p; l = 0, 1, \dots, p \quad (5.68)$$

Equations 5.66, 5.67 and 5.68 represent the gradient descent learning algorithm for the Elman network. In fact, the learning algorithms that were developed for MFNN in the B.2 and B.3 can be applied for Elman networks [Principe et al., 1999 and Christodoulou and Georgopoulos, 2001]. Similar to MFNN, the choice between these learning algorithms depends on the accuracy requirement, computation complexity, storage space requirement, and training speed.

5.4 Performance Analysis of a MFNN and SRN

To evaluate the difference in performance between the MFNN and SRN for INS/GPS integrated land vehicular navigation systems, two simple position update architectures (SPUA) that consisted of a two layered MFNN and a two layered SRN, respectively, were designed . The topologies of both architectures are shown in Figure (5.15) and Figure (5.16). The input vectors and output vectors of both architectures are exactly the same. The input vectors are the north position, east position, and current time epoch while the output vectors include the positioning errors of INS in the north and east directions. Figure (5.17) illustrates the learning strategy of the SPUA.

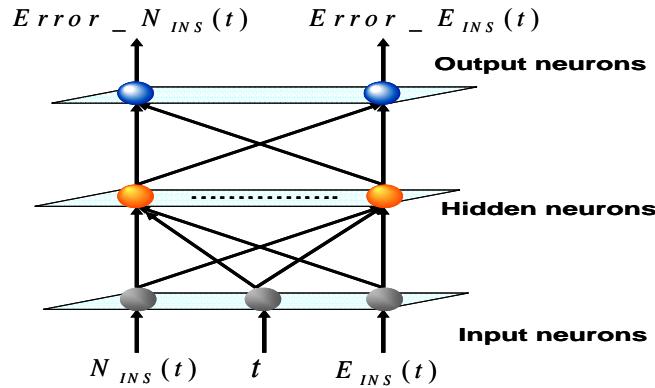


Figure 5.15: Topology of SPUA_MFNN

SPUA receives the position and time information from the INS mechanization and applies the position errors that are generated by comparing the INS position vectors and DGPS position vectors as the desired outputs. Thus, during the availability of DGPS, SPUA continues the training procedure to update the synaptic weights and adapt the dynamical model of the moving vehicle carrying both sensors. In contrast, it can then predict the vehicle's position errors generated by INS during a GPS signal blockage.

Two field test data were applied for the evaluation. The first field test was conducted by the Mobile Multi-sensor Systems (MMSS) research Group of the University of Calgary. The test was conducted in a land vehicle environment using the MMSS DGPS/INS integrated system utilizing a Litton LN 90-100 system, a strapdown navigation grade

IMU and two Ashtech Z12 receivers. The specifications of LN 90-100 system is given in Appendix C. The duration of the test was about 2500 seconds and no GPS signal outage periods were intentionally introduced as the purpose of this trajectory was to obtain the initial weights or stored weights for the estimation of a second field test. The first trajectory is shown in the Figure (5.18).

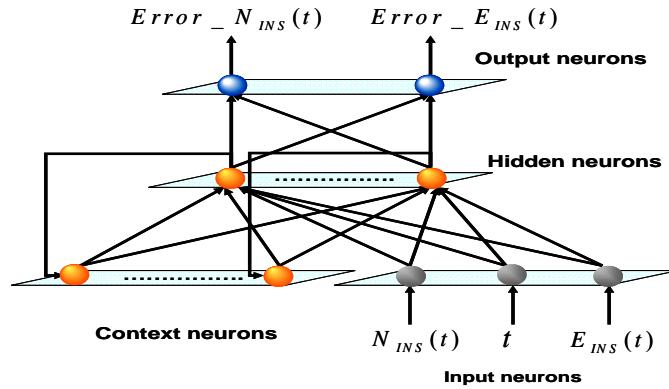


Figure 5.16: Topology of SPUA_SRN

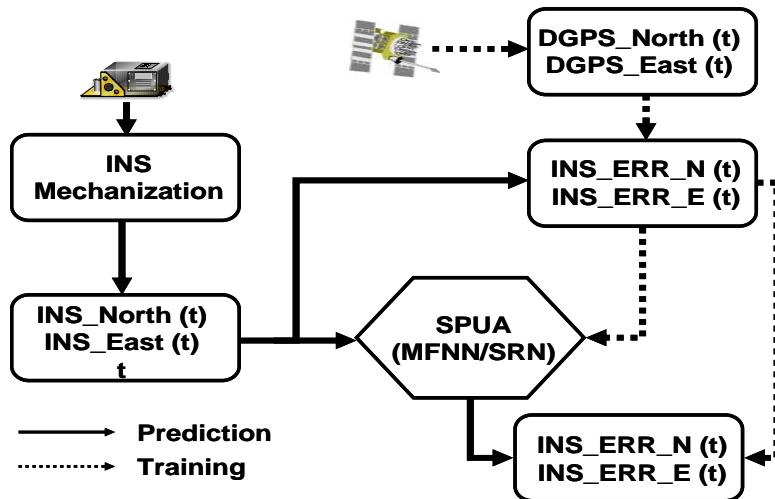


Figure 5.17: Learning strategy of SPUAs

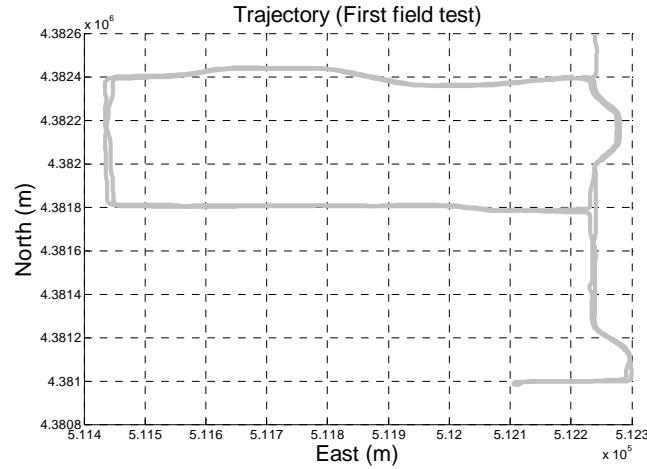


Figure 5.18: Trajectory of the first field test

The second field test was made available by the Institut fuer Astronomische und Physikalische Geodäsie, Technische Universität München. The test was conducted in a land vehicle environment using an integrated iMAR® iNAV-RQH system, a strapdown navigation grade IMU and two Trimble-4000SSI receivers. The specifications of iMAR® iNAV-RQH is given in Appendix C. The duration of the test was about 2100 seconds and six GPS signal outage periods were intentionally introduced with different motion dynamic variation, as shown in Figure (5.19). The durations of these outages were 30 seconds, 60 seconds, 30 seconds, 300 seconds 60 seconds and 120 seconds, respectively.

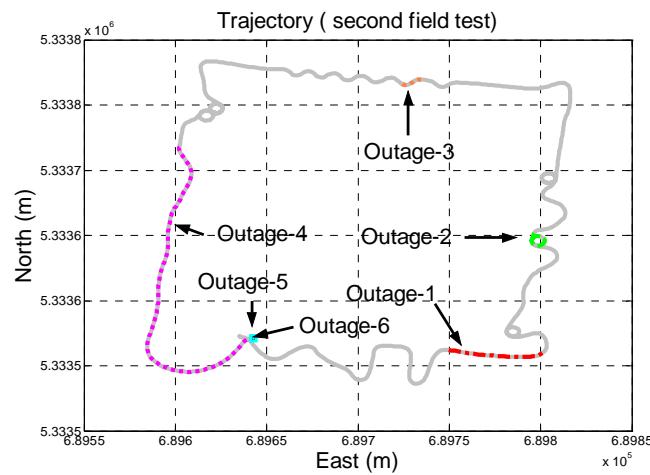


Figure 5.19: Trajectory of the second field test

During the availability of the DGPS signal, the stored weights of the SPUA obtained by the first field test data were updated using the window based weights updating strategy that is given in the next chapter. Thus, the latest updated weights were applied to predict the INS position errors and correct them to fill the gap between DGPS signal outages. In order to compare the SPUA performance against conventional integration techniques, the results obtained from the SPUA and Kalman filter were compared with a reference trajectory which was generated by DGPS with positioning error less than 10cm. The results of the Kalman filter were obtained from the University of Calgary Kalman filter DGPS/INS integration (KINGSPADTM - KINematic Geodetic System for Positions and Attitude Determination software, [Schwarz and El-Sheemy, 1999]).

Table 5.7 lists the RMS errors of each outage. The results demonstrated that SPUAs were able to provide 1~2 metre level positioning accuracy with navigation grade IMU during a 5 minutes GPS outage, and in contrast, the positioning accuracy of the Kalman Filter during the 5 minutes GPS outage was about 7 metres. The result clearly indicates that the SPUAs are capable of integrating navigation grade IMU measurements with DGPS measurements and providing positioning accuracy with 60~75% improvement in comparison with the results obtained from a conventional DGPS/INS integration technique with a Kalman filter.

Figure (5.20) and (5.21) show the position error of each outage. It can be seen from both figures that the performance of the Kalman filter was mainly related to the length of the GPS outage; in other words, it grew with time. The Kalman filter was capable of providing superior performance over SPUAs when the length of GPS outage was less than one minute; on the contrary, SPUAs outperformed the Kalman Filter when the length of the GPS outage was longer than one minute. The results demonstrated the positioning accuracy of SPUAs was mainly affected by the vehicle's motion dynamics, instead of time. It is a significant difference and improvement in comparison with time growing errors characteristic of the Kalman filter.

Table 5.7: RMS Errors

	KF (m)	SPUA_MFNN(m)	SPUA_SRN(m)	Vehicle motion
Outage_1 (30s)	0.08	0.39	0.51	<i>Straight line</i>
Outage_2 (60s)	0.09	2.48	5.2	<i>Circle</i>
Outage_3 (30s)	0.13	0.26	0.95	<i>Curve</i>
Outage_4 (300s)	6.32	1.45	2.65	<i>Straight line + Curve+ Sharp turn</i>
Outage_5 (60s)	0.06	0.04	0.29	<i>Static</i>
Outage_6 (120s)	0.12	0.07	0.14	<i>Static</i>

There is no strong indication in this case that the utilization of a dynamic neural network (Elman network) can provide any superior performance than a static neural network (MFNN) in the position domain. To evaluate the performance in the time domain, a segment of the first field test trajectory with a time period of 500 seconds was extracted as the training materials for both SPUAs. Both SPUAs had the same number of hidden neurons and utilized the same learning algorithm (i.e., Levenberg-Marquardt Algorithm). The weights were initialized randomly first then trained using the training materials for 10 independent trials with the same training goal (i.e., 10^{-8}). The required convergent time of SPUAs was then given as in Table 5.8.

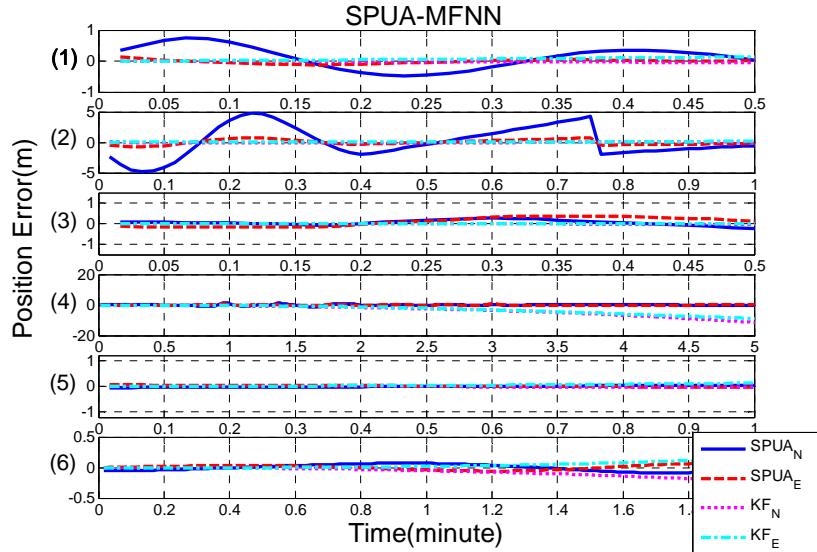


Figure 5.20: Position errors for the SPUA and KF during different GPS outages

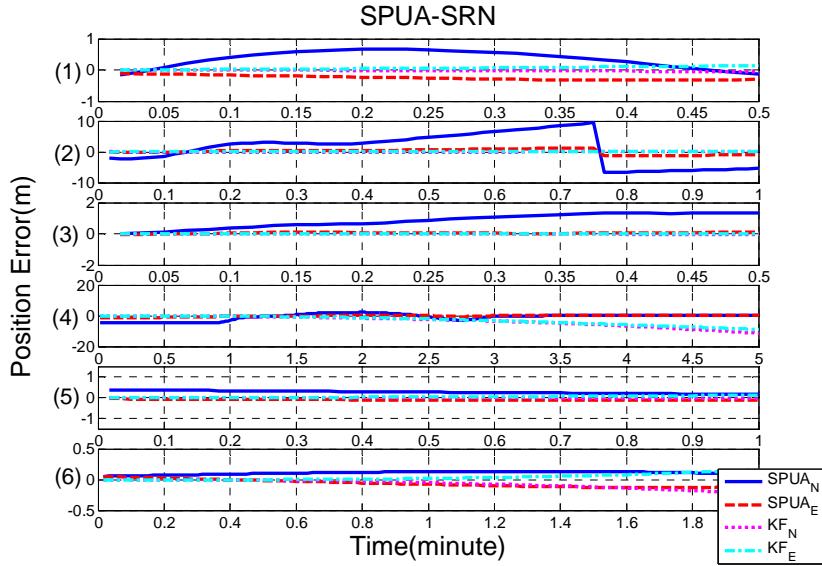


Figure 5.21: Position errors for the SPUA and KF during different GPS outages

Table 5.8: Training time required for 500 seconds data

Trails	1(epoch/time)	2(epoch/time)	3(epoch/time)	4(epoch/time)	5(epoch/time)
MFNN	6/2s	5/1s	8/3s	6/2s	8/3s
SRN	416/>5m	328/>5m	655/>5m	320/>5m	471/>5m
Trails	6(epoch/time)	7(epoch/time)	8(epoch/time)	9(epoch/time)	10(epoch/time)
MFNN	12/6s	7/2.5s	8/3s	8/3s	8/3s
SRN	627/>5m	385/>5m	848/>5m	803/>5m	534/>5m
Unit (s): second, (m):minute					

It can be seen from the above table that the learning speed of a dynamic neural network (Elman network) is much slower than that of a static neural network (MFNN). In other words, dynamic neural networks can not provide superior performance than a static neural network in terms of accuracy and learning speed in this INS/GPS integration case.

Thus, a MFNN is recommended as the core algorithm for the development of the conceptual intelligent navigator, which is given in the next chapter, due to its reasonable accuracy, storage requirement, learning speed and computation complexity.

CHAPTER 6

DEVELOPMENT OF THE CONCEPTUAL INTELLIGENT NAVIGATOR

As mentioned in previous chapters, Kalman filter approach has been widely applied as the core algorithm for INS/GPS integrated systems for many navigation applications. Although it represents one of the best estimation algorithms for INS/GPS integration applications, it has limitations in terms of model dependency, priori knowledge dependency, sensor dependency, and linearization dependency for general INS/GPS integrated navigation applications, see section 3.3.2 for more details. Consequently, in order to overcome or reduce the impact of these limitations, several research works have been conducted to investigate possible alternative algorithms for INS/GPS integrated navigation systems, see section 2.4.2.

Among these efforts, Chiang and El-Sheimy [2002], Chiang et al., [2003], El-Sheimy et al., [2003] Chiang [2003], and Chiang [2004] developed several INS/GPS integration architectures using ANNs technology. The preliminary results demonstrated that ANNs based INS/GPS integration algorithm could overcome or, at least reduce the impact of the limitations of the conventional Kalman filter. Thus, the optimal goal of this chapter is to propose a conceptual intelligent navigator that uses ANNs as the core algorithm for INS/GPS integration.

Based on the knowledge gained in the previous chapter, two layered feed-forward neural networks will be applied to build the proposed INS/GPS integration architectures. Such a conceptual intelligent navigator is expected to provide positioning information (N, E), velocity information (V_N, V_E) and azimuth information (φ) and to be able to overcome the limitations of the Kalman filtering algorithm for INS/GPS integrated land vehicular navigation applications in terms of positioning accuracy during GPS signal outages. Figure (6.1) illustrates a comparison between the system architecture of the conventional navigator and conceptual intelligent navigator for land vehicular applications.

The human brain has the ability to learn adaptively in response to knowledge, experience and environments by a network of interconnected adaptive information processing elements that transform inputs to desired outputs [Principe et al, 2000]. Thus, the conceptual intelligent navigator is expected to have the ability to learn and adapt.

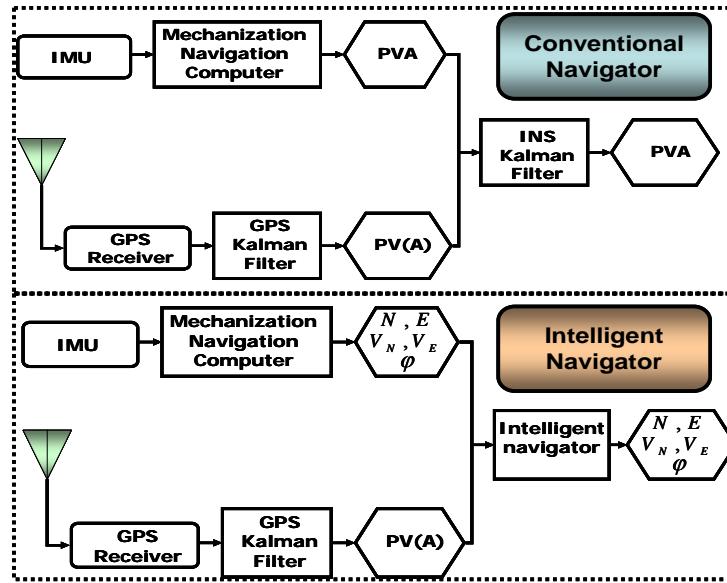


Figure 6.1: Comparison between conventional and conceptual intelligent navigator

Learning is defined as a process of acquiring and memorizing new information, knowledge and experience [Callatay, 1992]. The adaptation can be regarded as the ability of the information processing elements to change in a systematic manner and alter the nonlinear transformation between inputs and outputs. Thus, three function schemes of the conceptual navigator that fulfill the requirements of self-learning or adaptive learning are given as follows:

- **Generate navigation knowledge:** An intelligent navigator should be able to generate or acquire necessary navigation knowledge by itself from the information provided by the navigation sensors (i.e., INS, GPS, DR, CCD...etc).

Thus, the navigation knowledge can be applied to provide necessary navigation solutions.

- **Store navigation knowledge:** Humans have the ability to learn, memorize, and store enormous amounts of information. Furthermore, humans also have the ability to generalize what they have learnt based on the information stored in their memory [Callatay, 1992]. In addition, memory consists of time dependent processing of encoding and retrieval of the information stored in the brain as a result of sensory knowledge or experience. Therefore, an intelligent navigator should be able to store and generalize the navigation knowledge generated by previous function schemes.
- **Accumulate navigation knowledge:** The real learning process of humans is conducted through continuous accumulation of knowledge or experience then transforming such information in the form of long term memory. In other words, long term memory refers to the more or less permanent form of information in the brain. In general, it might take many repetitions or activations of sensory information to establish long-term memory by the process of continuous learning, memorizing and generalizing [Vidyasagar, 2002]. Thus, an intelligent navigator should possess a strategy to accumulate the navigation knowledge to mimic the way humans learn and accumulate knowledge.

Figure (6.2) illustrates the core components of the conceptual intelligent navigator. Several ANNs based INS/GPS integration architectures are given first in this chapter to generate the necessary navigation knowledge for the conceptual intelligent navigator. After that, the concept of a navigation information database is discussed to provide storage space of the navigation knowledge, and then a window based weights updating strategy is given as a tool for accumulating the navigation knowledge. Before leaving this chapter, the conceptual intelligent navigator is implemented and evaluated.

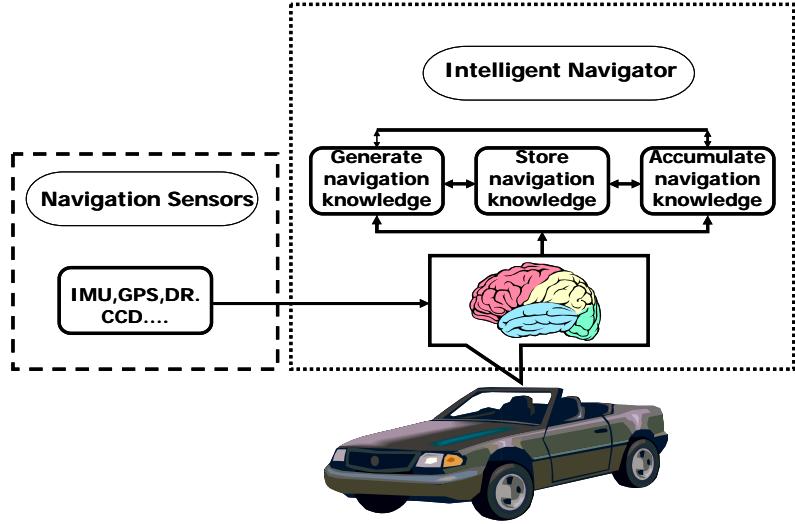


Figure 6.2: Core components of the conceptual intelligent navigator

6.1 ANNs Based INS/GPS Integration Architecture

Being inspired by Figure (6.2), the first step toward building the conceptual intelligent navigator is to provide the INS/GPS integration architectures that can be applied to generate and acquire necessary navigation knowledge. Consequently, three ANNs based INS/GPS integration architectures are designed and discussed in the following sections.

6.1.1 Position Update Architecture

The first INS/GPS integration architecture is called the Position Update Architecture (PUA) which consists of a two layered feed-forward neural network. It integrates the data from INS and DGPS and mimics the dynamical model of the vehicle to generate navigation knowledge. Thus the latest acquired navigation knowledge can be applied to predict the vehicle's position during GPS outages in real time [Chiang and El-Sheimy, 2002].

The topology of the first generation PUA is illustrated on the left side of Figure (6.3). The input neurons receive the velocity at current epoch ($V_{INS}(t)$), azimuth at current epoch

$(\phi_{INS}(t))$ from INS mechanization, and the two dimensional coordinates in the local level frame at previous epoch $(N_{GPS}(t-1), E_{GPS}(t-1))$ from GPS solutions. The number of hidden neurons is decided empirically; see Chiang et al., [2003] for more details. The output neurons generate two dimensional coordinates in the local level frame at the current epoch $(N_{PUA}(t), E_{PUA}(t))$. However, according to Chiang and El-Sheimy [2002], the 1st generation PUA can be further simplified, thus a 2nd generation PUA is given in Figure (6.3). The input neurons receive only the velocity at the current epoch $(V_{INS}(t))$, azimuth at the current epoch $(\phi_{INS}(t))$ and the output neurons generate the two dimensional coordinates difference between two consecutive epochs in the local level frame $(\delta N(t), \delta E(t))$.

The purpose of using the coordinate differences instead of the position component itself is to simplify the learning process. In fact, the differences can reduce the complexity of the input/output function relationship thus they provide a more efficient NN training and reduce the time required for the training procedure [Chiang and El-Sheimy, 2002].

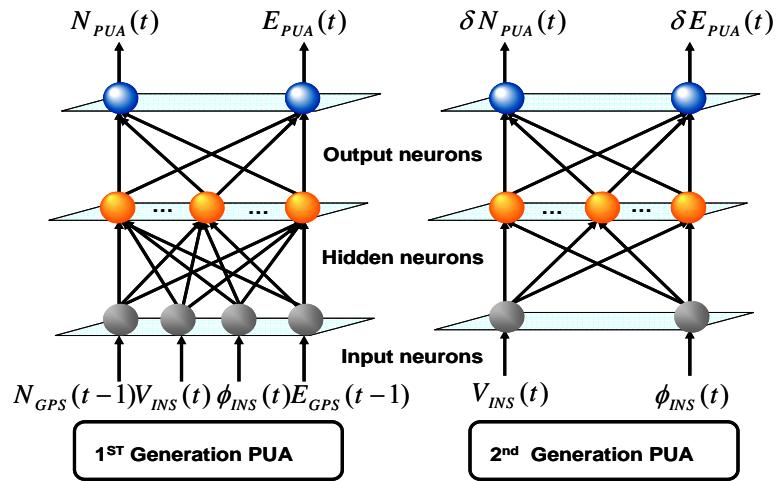


Figure 6.3: Topology of PUA

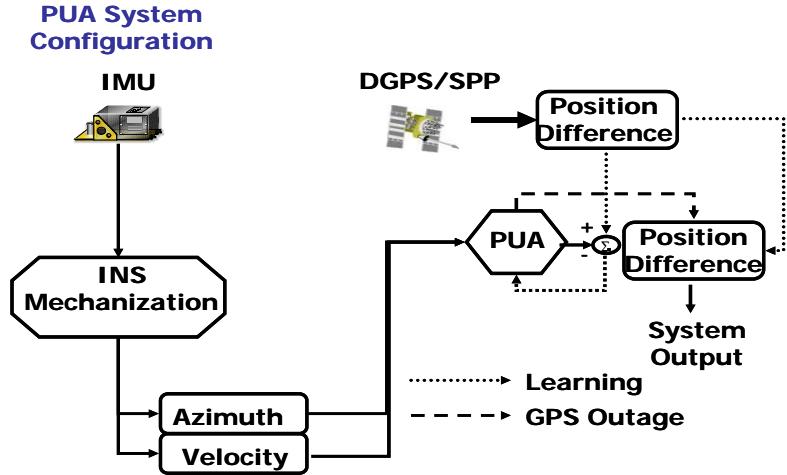


Figure 6.4: System configuration and learning strategy of PUA

Figure (6.4) illustrates the system configuration and learning strategy of the PUA. The system receives the outputs from INS mechanization at t epoch ($V_{INS}(t), \phi_{INS}(t)$) and is expected to generate the coordinate differences between two consecutive epochs ($\delta N_{PUA}(t), \delta_{PUA} E(t)$). The desired outputs ($\delta N_{GPS}(t), \delta_{GPS} E(t)$) are provided by GPS during signal availability in either DGPS or SPP mode of operation. As long as the GPS signals are available for more than 4 satellites, the learning process continues to reduce the estimation error in order to obtain optimal values of the NN parameters. In this case, the desired outputs ($\delta N_{GPS}(t), \delta_{GPS} E(t)$) are applied as the system output. Thus, the navigation knowledge can be learnt, stored and accumulated during the availability of the GPS signal. On the other hand, during GPS signal outages, the latest acquired navigation knowledge can be retrieved from the “brain” (navigation information database, see section 6.2 for more details) of the intelligent navigator to predict the position ($N_{PUA}(t), E_{PUA}(t)$) in real time.

6.1.2 Position and Velocity Update Architecture

The second INS/GPS integration architecture is named the Position and Velocity Update Architecture (PVUA) [El-Sheemy et al, 2003]. It consists of two different two layered

feed-forward neural networks that work in parallel. Similar to PUA, PVUA is applied to generate navigation knowledge which can be used to provide, in real time, the vehicle's position and velocity during GPS signal outages. The topologies of PVUA are illustrated in Figure (6.5). In fact, PVUA is the combination of a Velocity Update Architecture (VUA) with a modified version of the PUA.

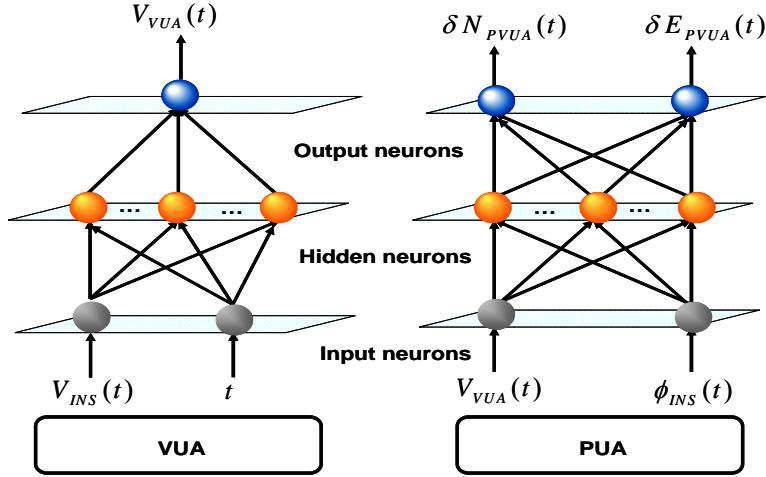


Figure 6.5: Topologies of PVUA

The input neurons of the VUA receive the velocity from the INS mechanization ($V_{INS}(t)$) and time epoch (t). The output neuron generates VUA estimated velocity ($V_{VUA}(t)$) as the velocity input for the modified PUA. Thus, instead of using velocity information from INS mechanization directly in the original PUA, the modified PUA receives the velocity from the VUA and combines it with the azimuth obtained from the INS ($\phi_{INS}(t)$). The final results are the two dimensional coordinates difference between two consecutive epochs in the local level frame ($\delta N_{PVUA}(t), \delta E_{PVUA}(t)$). Figure (6.6) illustrates the system configuration and learning strategy of the PVUA. The system utilizes the outputs from INS mechanization at t epoch ($V_{INS}(t), \phi_{INS}(t)$) along with the time information (t) and is expected to generate the coordinate differences between two consecutive epochs ($\delta N_{PUA}(t), \delta E_{PUA}(t)$). The desired outputs for the VUA and modified PUA are ($V_{DGPS}(t)$)

and $(\delta N_{DGPS}(t), \delta E_{DGPS}(t))$, respectively. They are provided by GPS as long as the GPS signals are available from at least 4 satellites.

During GPS outage, the modified PUA receives the VUA predicted velocity ($V_{VUA}(t)$) and the azimuth ($\phi_{INS}(t)$) obtained from INS to generate coordinate differences between two consecutive epochs $(\delta N_{PVUA}(t), \delta E_{PVUA}(t))$. Thus, similar to PUA, the latest acquired navigation knowledge obtained through use of the PVUA can be retrieved from the “brain” of the intelligent navigator to predict the positions $(N_{PVUA}(t), E_{PVUA}(t))$ in real time to fill the gaps during GPS outages.

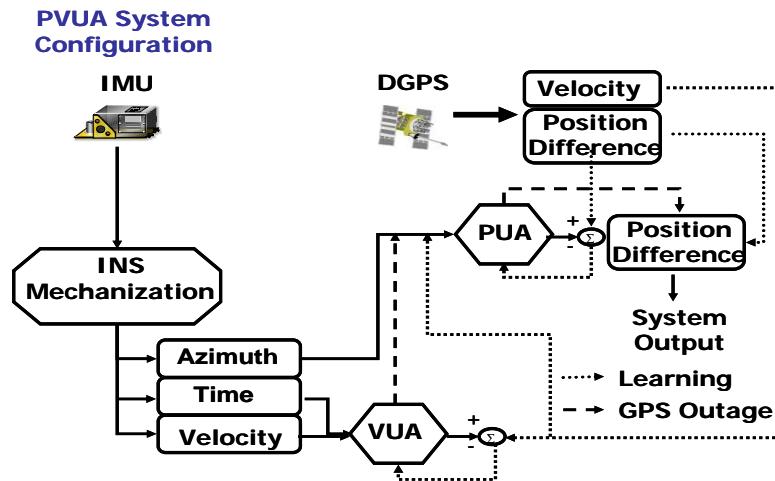


Figure 6.6: System configuration and learning strategy of PVUA

6.1.3 Position, Velocity, and Azimuth Update Architecture

The third INS/GPS integration architecture is called the Position, Velocity and Azimuth Update Architecture (PVAUA) [Chiang, 2004 and Chaing and El-Sheimy, 2004b]. It consists of three different two layered feed-forward neural networks that work in parallel. Similarly, the PVAUA can be applied to generate navigation knowledge which can be used to provide the vehicle’s position, velocity and azimuth to fill the gap between GPS outages in real time. In fact, it can be regarded as the combination of an Azimuth Update

Architecture (AUA) with a modified PUA. Instead of receiving the velocity and azimuth obtained from INS mechanization at time epoch t ($V_{INS}(t), \phi_{INS}(t)$), the modified PUA receives the velocity ($V_{VUA}(t)$) and azimuth ($\phi_{AUA}(t)$) generated by the VUA and AUA, respectively. Thus the modified PUA generates two dimensional coordinate differences between two consecutive epochs in the local level frame ($\delta N_{PVAAU}(t), \delta E_{PVAAU}(t)$).

Similarly, the desired outputs required for training the VUA and the modified PUA networks can be obtained directly from DGPS solutions (i.e. $V_{DGPS}(t), \delta N_{DGPS}(t)$ and $\delta E_{DGPS}(t)$). As for the AUA, the desired output can be provided by a “physical” measurement (i.e., magnetic compass) or a “pseudo” measurement (i.e., DGPS derived azimuth or multi-antenna derived azimuth). Considering its simplicity, cost, and integrity, raw DGPS derived azimuth ($\phi_{DGPS-(V,P)}(t)$) is preferred as the “pseudo” measurement to provide the desired output for training AUA. Mathematically, the DGPS derived azimuth at t epoch ($\phi_{DGPS-(V,P)}(t)$) can be calculated as follows:

$$\phi_{DGPS-(V)}(t) = \tan^{-1} \frac{V_E(t)}{V_N(t)} \quad \text{or} \quad \phi_{DGPS-(P)}(t) = \tan^{-1} \frac{\delta E(t)}{\delta N(t)} \quad (6.1)$$

However, the azimuth derived from DGPS velocities ($\phi_{DGPS-(V)}(t)$) or DGPS position ($\phi_{DGPS-(P)}(t)$) becomes unstable when the denominator approaches zero (e.g. during ZUPT or for low V_N and small $\delta N(t)$), as shown in Figure (6.7). In other words, they suffer a numerical problem during low dynamics (low V_N and small $\delta N(t)$) or ZUPT due to the nature of the inverse tangent algorithm. Therefore, an azimuth constraint algorithm that can be applied to reduce such limitation and provide stable DGPS derived azimuth for the utilization of PVAAU is given as follows [Chiang, 2004 and Chaing and El-Sheimy, 2004b];

- (1) Calculate GPS derived azimuth, $\phi_{DGPS-(V,P)}(t)$, using Equation 6.1
- (2) Using Equation 6.2 to obtain constrained GPS derived azimuth, $\bar{\phi}_{DGPS-(V,P)}$, during ZUPT, where t is the beginning time epoch of ZUPT

$$\bar{\phi}_{DGPS-(V,P)}(t) = \phi_{DGPS-(V,P)}(t-1) \quad (6.2)$$

- (3) Using Equation 6.3 to obtain $\bar{\phi}_{DGPS-(V,P)}$, during low V_N or small $\delta N(t)$, where t is the beginning time epoch of low dynamics along the north direction and ϕ_{INS} is the azimuth provided by the INS mechanization

$$\bar{\phi}_{DGPS-(V,P)}(t) = \bar{\phi}_{DGPS-(V,P)}(t-1) + \phi_{INS}(t) - \phi_{INS}(t-1), \quad (6.3)$$

- (4) The whole process can be summarized as shown in Figure (6.8)

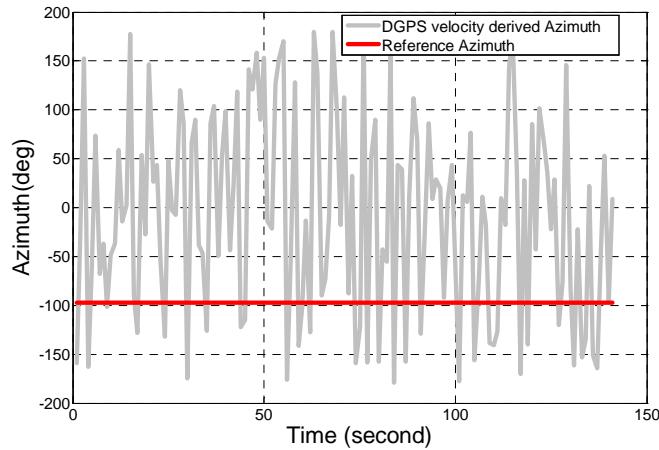


Figure 6.7: Instability of $\phi_{DGPS-V}(t)$

To gain more appreciation to the azimuth constrain algorithm, the field test data applied in chapter 4 is used to evaluate the performance of constrained GPS derived azimuth. Figure (6.9) illustrates the outputs of raw DGPS velocity derived azimuth, IMU azimuth, constrained DGPS velocity derived azimuth and reference azimuth during several ZUPTs and low dynamics (i.e., low V_N and small $\delta N(t)$). As mentioned previously, raw DGPS velocity derived azimuth demonstrated oscillatory behavior during ZUPTs and low dynamics scenarios while IMU azimuth drifted with time. In contrast, constrained DGPS velocity derived azimuth demonstrated its stability and reduced the impact of the numerical issue. Table 6.1 lists the RMS error of different azimuth outputs. More results

about the performance of constrained DGPS velocity and position derived azimuths are given in Chapter 7.

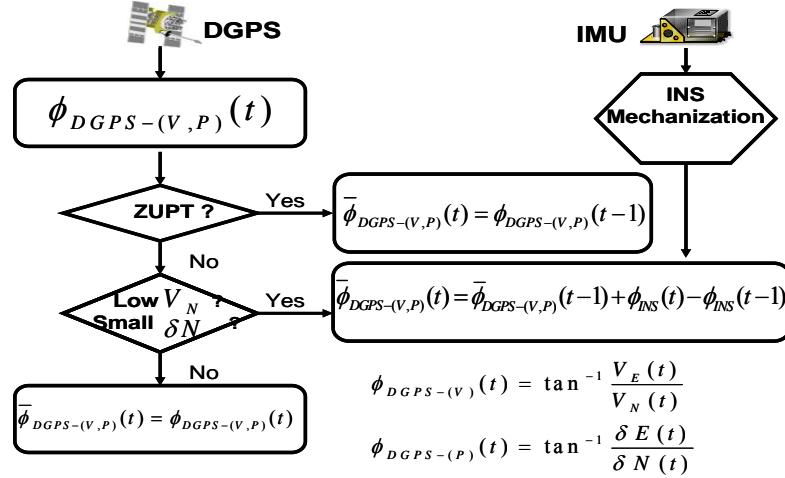


Figure 6.8: Azimuth constrain algorithm

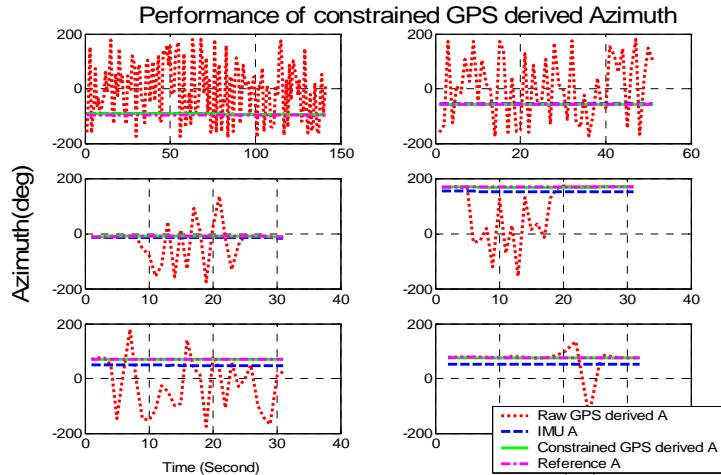


Figure 6.9: Comparison between different azimuth outputs

The results demonstrated that the raw GPS velocity derived azimuth was unreliable; therefore, it could not be applied directly with the utilization of the AUA. On the other hand, although IMU azimuth drifted with time in stand alone mode, the azimuth change between two consecutive epochs was quite accurate during short periods. Therefore, it can be applied to improve the quality of raw DGPS velocity derived azimuth. As a result,

constrained DGPS velocity derived azimuth ($\bar{\phi}_{DGPS-(V)}$) demonstrated its superior performance than raw DGPS velocity derived ($\phi_{DGPS-(V)}$) and IMU azimuth. The topologies of PVAUA are given in Figure (6.10). The PVAUA consists of an AUA, a VUA and a modified PUA. The topology of VUA used by PVAUA can be found in Figure (6.5).

Table 6.1: Performance Summary (RMSE)

	1	2	3	4	5	6
Raw DGPS derived Azimuth (deg)	136.41	132.38	70.62	120.60	130.80	55.17
IMU Azimuth (deg)	4.80	1.95	4.63	17.01	21.91	23.71
Constrained DGPS derived Azimuth (deg)	4.80	1.02	0.66	1.59	0.27	0.60
GPSTime-150000 (sec)	410:550	640:690	990:1020	1490:1520	1600:1630	1690:1705

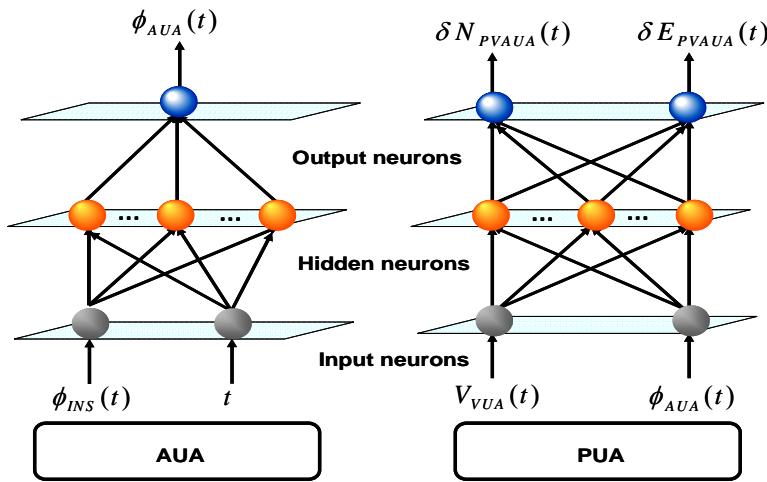


Figure 6.10: Topologies of PVAUA

The input neurons of the AUA receive the azimuth from the INS mechanization ($\phi_{INS}(t)$) along with the time epoch (t). The output neuron generate AUA estimated azimuth ($\phi_{AUA}(t)$) as the input azimuth for the modified PVA. Thus, instead of using velocity and

azimuth information from the INS mechanization directly in the original PUA, the modified PVA receives the velocity ($V_{VUA}(t)$) from VUA and combines it with the azimuth obtained from AUA ($\phi_{AUA}(t)$) to generate two dimensional coordinate differences between two consecutive epochs in the local level frame ($(\delta N_{PVUA}(t), \delta E_{PVUA}(t))$). Figure (6.11) illustrates the system configuration and learning strategy of the PVAUA.

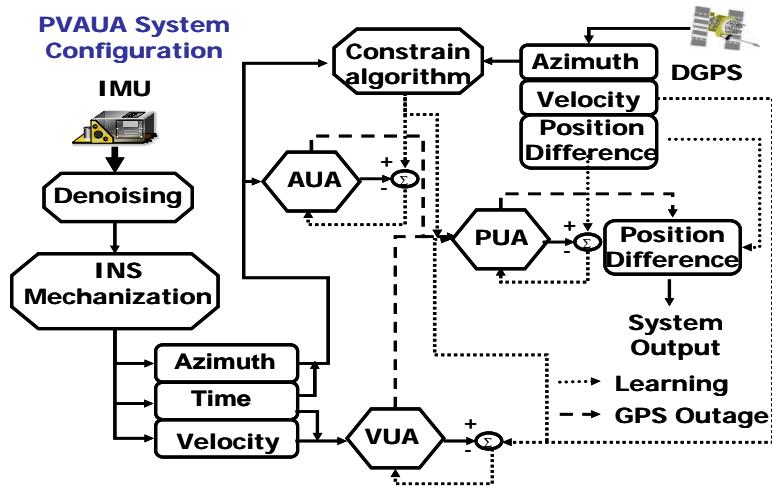


Figure 6.11: System configuration and learning strategy of PAVUA

The system utilizes the outputs from the INS mechanization at t epoch ($V_{INS}(t), \phi_{INS}(t)$) along with the time information (t) and is expected to generate the coordinate differences between two consecutive epochs ($(\delta N_{PUA}(t), \delta E_{PUA}(t))$). The desired outputs for the AUA, VUA and modified PUA are ($\bar{\phi}_{DGPS}(t)$), ($V_{DGPS}(t)$) and ($(\delta N_{DGPS}(t), \delta E_{DGPS}(t))$), respectively. They are provided by GPS during signal availability in DGPS mode. During GPS outages, the modified PUA receives the VUA predicted velocity ($V_{VUA}(t)$) and the AUA predicted azimuth ($\phi_{AUA}(t)$) to generate coordinate differences between two consecutive epochs ($(\delta N_{PVUA}(t), \delta E_{PVUA}(t))$). Thus, similar to PUA and PVUA, the latest acquired navigation knowledge obtained through the use of PVAUA can be retrieved

from the “brain” of the intelligent navigator to predict the positions ($N_{PVAUA}(t), E_{PVAUA}(t)$) in real time.

Using the above presented INS/GPS integration architectures, the navigation knowledge can be acquired and learnt, however, a “brain” is required for the intelligent navigator to store, accumulate and generalize what it has learnt. Although the architecture that can be applied to predict the height of the vehicle is not implemented in research because the scope of this research is limited to the horizontal navigation solutions, It is worth mentioning that such architecture can be implemented easily by replacing the velocity with height in Figure (6.5) and uses GPS information to provide navigation knowledge associated with the height.

6.2 Navigation Information Database

As mentioned previously, learning can be defined as the acquisition of new knowledge, experience and information. Thus, learning is a process of memorizing the learnt knowledge or experience in the human brain. Thus, the second step towards building the conceptual intelligent navigator is to store the learnt navigation knowledge provided by INS/GPS integration architectures presented in the previous section. As a result, a navigation information database (NAVi) that contains the acquired and learnt navigation knowledge can serve as the “brain” of the conceptual intelligent navigator. Therefore, several issues regarding the NAVi are addressed as follows [Chiang and El-Sheemy, 2004c];

- **Content of NAVi:** The database consists of the training samples (input vectors and desired output vectors) and estimated synaptic weights during the availability of the GPS signal. Thus, these components can be regarded as the navigation knowledge. In other words, the content of the database varies with the topologies of different INS/GPS integration architectures. Figure (6.12) illustrates the NAVi of the PUA. It can be modified to fit the requirement of PVUA and PVAUA using Figure (6.5) and Figure (6.10), respectively.

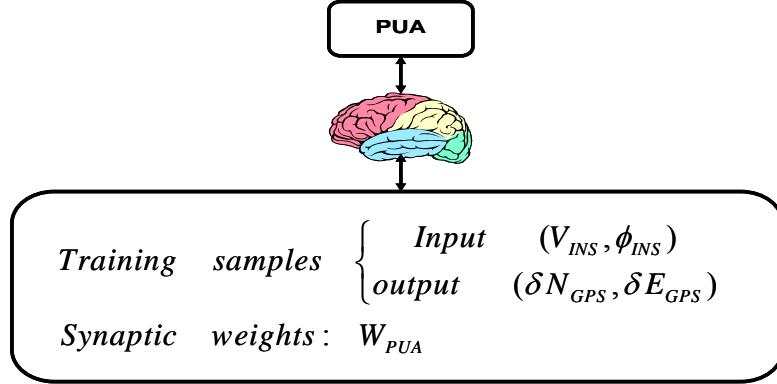


Figure 6.12: NAVi of PUA

- **Distributed navigation knowledge storage:** The content of NAVi becomes more complicated with complicated INS/GPS integration architectures, such as PVAUA. Therefore, considering the efficiency of database maintenance and retrieval, the navigation knowledge learnt by each sub-component should be stored individually in a distributed way, as shown in Figure (6.13).

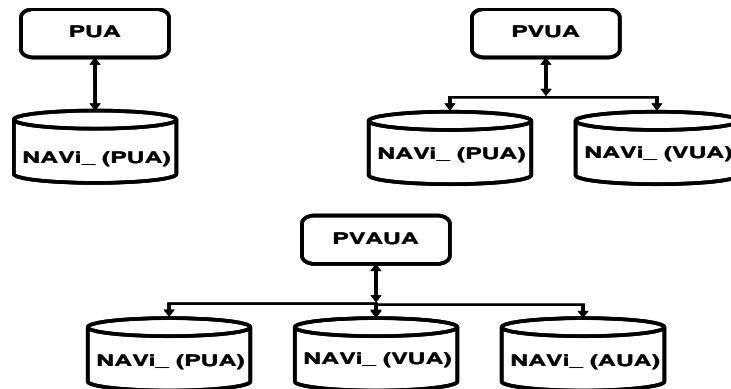


Figure 6.13: Distributed navigation knowledge storage

- **Off-line database maintenance:** As the goal of NAVi is to store the learnt and acquired navigation knowledge to provide long term memory to the conceptual intelligent navigator, therefore, the capacity of long term memory increases when the learning process increases (i.e. .more demand for storage space). Since the

goal of such an intelligent navigator is to mimic the way humans learn, accumulate, memorize, and generalize new knowledge or experience, the storage space requirement is the price to pay. However, the storage space is not unlimited thus it is necessary to employ either on-line (during navigation) or off-line (before or after) database maintenance methods to reduce the impact of storage requirement. As on-line maintenance might cause delay in providing navigation in real time, thus off-line database maintenance methods are recommended. The simplest way to reduce the storage requirement is to remove any redundant training samples that include inputs and their corresponding desired outputs. As for the synaptic weights, they should be kept without any change as they are the core component of the navigation knowledge. Using PUA as an example, a simple procedure that can be applied prior to navigation (i.e., during alignment, ZUPT...etc) or after navigation before shutting down the system, is given below.

- Regroup the training samples: Using one of the training inputs (i.e., $V_{INS}(t)$ or $\phi_{INS}(t)$) as the index; the training inputs can be regrouped to increase the efficiency for maintenance.
- Locate redundant navigation knowledge: Although it is difficult to locate a pair of training samples that are exactly the same, searching the most similar pairs of training samples using threshold values then deciding if they are redundant or not is possible, as shown in Figure (6.14). $\varepsilon_V, \varepsilon_\phi, \varepsilon_{\delta N}$ and $\varepsilon_{\delta E}$ are the threshold values of the velocity, azimuth and coordinate differences, respectively.
- Remove the redundant navigation knowledge.

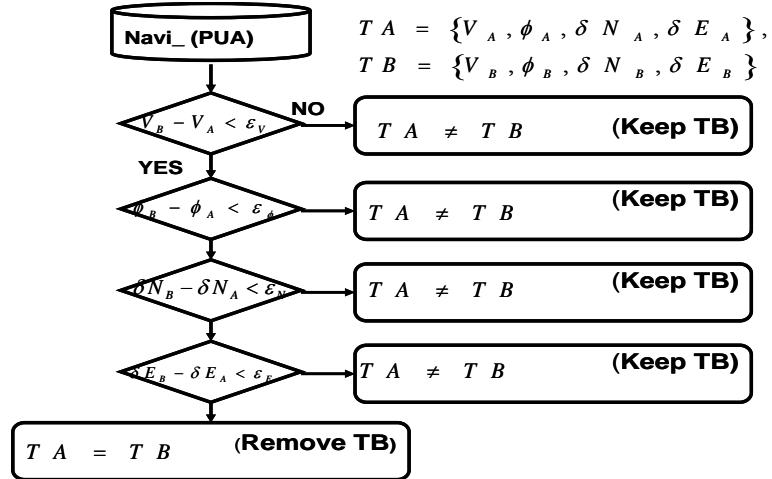


Figure 6.14: Locate redundant navigation knowledge

Up to now, the conceptual intelligent navigator has been given the ability to generate, and learn navigation knowledge and it also has been given the “space” to store navigation knowledge. However, there is still one thing missing. It requires a way to accumulate the acquired and learnt navigation knowledge and store them for further retrieving or generalization.

6.3 Window Based Weights Updating Strategy

As the synaptic weights are the core components of the navigation knowledge, the final step towards building the conceptual intelligent navigator is to develop a strategy to accumulate the acquired navigation knowledge by updating the synaptic weights whenever the GPS signal is available.

In most of their applications, ANNs are trained using some known training data set (input/desired output) to obtain the optimal values of the synaptic weights via off-line training. For any other set of inputs, different from those used in training, the synaptic weights can then be applied to provide prediction of the network outputs. It is worth mentioning that ANNs weights are frozen after completing the training procedure and no

further modification will be made during the prediction process [Ham and Kostanic, 2001].

In fact, off-line training can work well in case of slowly changing time sequences (e.g., weather forecasting, stock market prediction, robot trajectory control...etc.) [Saad, 1998]. In the case of INS/GPS integration for navigation applications, it is required to track direction changes and mimic the motion dynamics utilizing the latest available INS and GPS data. In other words, the synaptic weights should be updated during the navigation process to adapt the network to the latest INS sensor errors and the latest dynamics condition whenever the GPS signal is available.

To implement such criterion, a window-based weights updating strategy, which utilizes the synaptic weights obtained during the conventional off-line training procedure (or probably from previous navigation missions) is stored in the NAVi and is developed in this research [Chiang et al., 2004a]. This criterion utilizes the latest available navigation information provided by the GPS signal window to adapt the stored synaptic weights so that they can be applied to mimic the latest motion dynamic. The window-updated synaptic weights are stored after each training stage. They are then used as initial values for the weights to be estimated during the next training window or for prediction during GPS outages. Prior to looking into the details of the window based weights updating strategy, several aspects of traditional weights updating strategies are given in the next session.

6.3.1 Limitations of Traditional Weights Updating Methods

Traditional synaptic weights updating methods can be classified as: (1) sample-by-sample training, also known as on-line or sequential training, that modifies the weights for each input record after computing the weights updates; (2) batch training, which computes the synaptic weight updates for each sample and stores these values (without changing the weights). At the end of the whole training procedure, all the synaptic weight updates are added together and then the weights are modified with the accumulated synaptic weight

updates [Saad, 1998]. One major difference between sequential training and batch training is that batch mode keeps the synaptic weights constant while computing the error associated with each input sample. On the contrary, sequential mode is constantly updating its synaptic weights.

From an online operational point of view, the sequential mode of training is preferred over the batch mode since less local storage is required. In addition, the random presentation of the pattern makes it less likely for the standard backpropagation algorithm to be trapped in a local minimum if the sequential mode of training is utilized [Fine, 1999]. In contrast, the use of batch mode provides a more accurate estimate of the gradient vector, thus giving more accurate estimation of the weights.

Another major advantage of sequential training over batch training arises if there is a high degree of redundancy in the data [Fine, 1999]. For example, suppose that a vehicle is moving along a circular trajectory with constant velocity under an ideal condition for ten runs. As a result, the whole training data is ten times larger than a single run but contains a high level of redundancy. As a result, batch training takes ten times longer than sequential training to get the optimal values of the network weights [Haykin, 1999].

On the other hand, the sequential training updates the weights after receiving each record of the input samples. Therefore, it will not be affected by such highly redundant data [Fine, 1999]. However, during batch training, the network can learn more general relationships as it utilizes most of the available training data at the same time instead of sample by sample. Both generalization and training efficiency are very critical for INS/GPS integration applications, therefore, developing a special weights updating strategy that can preserve the generalization ability without losing too much training efficiency is very important.

6.3.2 Development of Window Based Weights Updating Strategy

The window-based weights updating strategy considers the previously stored weights as “long term memory”. Although the stored weights might not be able to provide accurate prediction during all GPS outages, it can be applied as the initial weights at the beginning of a new navigation mission. The GPS window signal concept is then applied to introduce new navigation knowledge to modify stored synaptic weights during navigation. In fact, this method combines the advantages of both sequential mode and batch mode of training in order to make the training procedure suitable for real-time processes. The proposed method tries to enlarge the sample size of sequential training to fit certain window lengths. In addition, the weights of each window are then updated via batch training mode. In other words, the weights of each window are updated sequentially. As depicted in Figure (6.15), the procedure of the window-based weights updating method is given [Chiang et al., 2004a]:

- **Weights initialization:** The initial weights can be obtained using previously stored weights that are stored in NAVi or random initialization. In this research, the initial weights were obtained using random initialization. After that, the weights were stored in NAVi after each navigation mission and could be applied as the initial weights for the next mission. Accurate initial weights may significantly reduce training time.
- **GPS signal reception:** Within the first GPS window ($i=1$), GPS (i), the synaptic weights are not updated, thus the stored synaptic weights are still the initial synaptic weights $W(i-1)$ (i.e., $W(0)$).
- **GPS signal reception:** At the next GPS window, GPS ($i+1$), the stored weights, $W(i-1)$, are updated utilizing the previous available GPS information (GPS (i)). These weights are stored as $W(i)$ after training is completed. Steps 2 and 3 are repeated until GPS signal blockage is detected.
- **GPS Outage:** As depicted in Figure (6.15), in case of a GPS outage (after GPS (i)), $W(i-1)$ is first applied for real time prediction and then $W(i)$ is utilized to

replace $W(i-1)$ and carry on real time prediction during the GPS outage.

The prediction using $W(i-1)$ and the training of $W(i)$ can be operated in parallel as the NAVi contains an exact duplication of the NN applied for the training of $W(i)$ with initial weight $W(i-1)$. In other words, one NN updates $W(i-1)$ using the latest acquired GPS (i) to acquire $W(i)$ and the duplicated one provides prediction using $W(i-1)$. For simplification, the update procedure during GPS outage can be paused thus $W(i-1)$ is applied to provide prediction during entire GPS outages and it can be updated after the reception of next available GPS signal window.

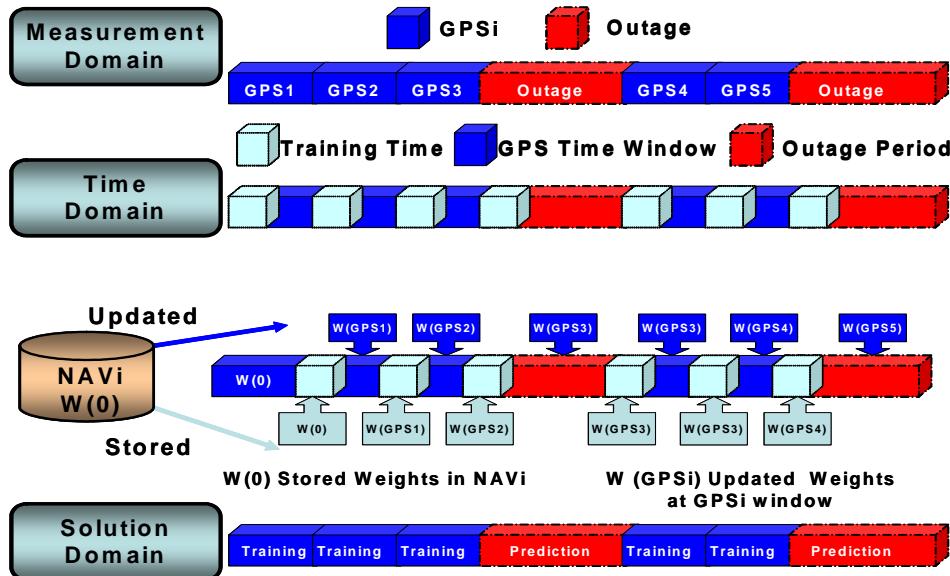


Figure 6.15: Window based weights updating strategy (After Chiang et al., 2004a)

Since the ANNs training procedure takes time, updating the synaptic weights immediately at the latest available sample of a GPS signal before outage is difficult. However, the utilization of the proposed method can still provide reasonable prediction accuracy during GPS outages since it provides the latest updated weights instead of real time updated weights for real time prediction. Therefore, failure in providing real time updated synaptic weights doesn't mean the intelligent navigator is not able to provide real

time prediction. On the contrary, it can utilize the latest acquired and learnt navigation knowledge to provide real time solutions. Combining the latest GPS window signals, stored weights can be adaptively updated to follow the latest motion dynamics and INS errors, thus improving the prediction accuracy during GPS outages.

The training samples acquired for the window based weights updating strategy can be arranged through using the following procedure:

- **One step training procedure:** As shown in Figure (6.16), the training samples acquired for each GPS window during navigation are the combination of stored training samples ($T_{(NAVi)}$) and available training samples obtained at the end of each GPS window ($\sum_{i=1}^n TW(i)$, n is the nth window). In other words, as the size of training samples increases during navigation, the size of $NAVi$ grows during navigation as well.

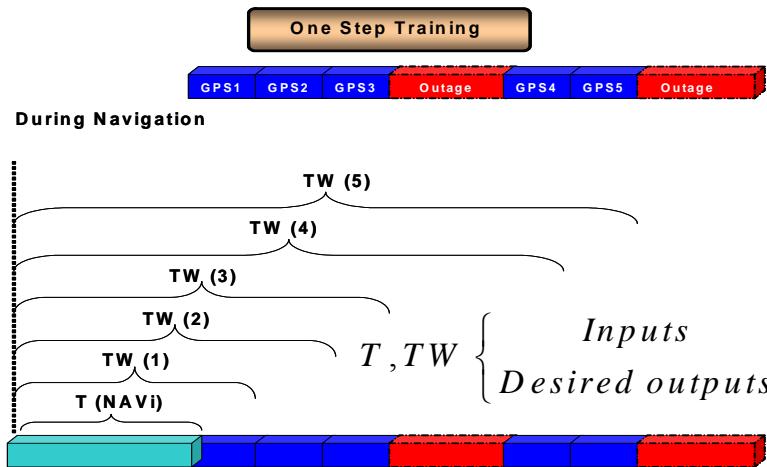


Figure 6.16: One step training procedure

The advantage of the one step training is that it can provide better generalization of the navigation knowledge by incorporating stored and previous training samples during navigation. The one step training procedure is recommended at

the early stage for building the intelligent navigator as the navigation knowledge acquired by the navigator at this moment might not be enough to provide acceptable accuracy during GPS outages. As the size of NAVi is quite small, the incorporation of stored training samples doesn't slow down the learning process during each window; actually, it can provide better generalization of the navigation knowledge.

- **Two steps training procedure:** As shown in Figure (6.17), the training samples acquired for each GPS window during navigation are obtained at the end of each GPS window (TW (i)). After navigation, all the training samples acquired during the navigation are recalled and combined with the stored training samples (T (NAVi)) then fed into the navigator to improve the generalization of navigation knowledge using a conventional off-line batch training method. This procedure is recommended for the regular operational stage for building the intelligent navigator. After several field tests, the navigator might accumulate enough navigation knowledge to provide navigation solutions during navigation without incorporating stored training samples. In other words, the size of training samples is the same as the GPS window. Therefore, the training speed during each window is expected to be faster than the previous procedure.

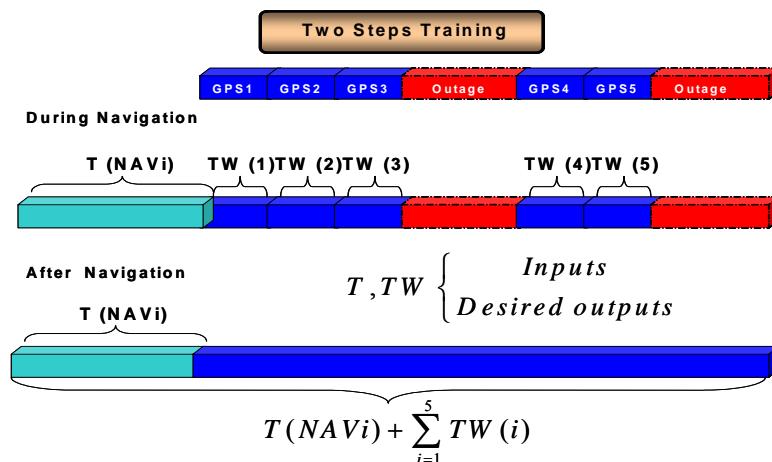


Figure 6.17: Two steps training procedure

After navigation, all the training samples acquired during the current navigation are recalled and combined with the stored training samples first to remove redundant navigation knowledge, and are then re-trained to improve the generalization of the navigation knowledge for future navigation missions.

As mentioned earlier, the development of the conceptual intelligent navigator is inspired by the learning process of human beings. The key factor that can accelerate the learning is the generalization of navigation knowledge. The perfect solution is to obtain the most generalized navigation knowledge that can then be fed into the navigator in one field test. However, that is not the case for real life applications. Therefore, the navigator must have the ability to evolve during each navigation mission to provide generalized navigation knowledge for future missions. Thus, using the proposed INS/GPS integration architectures, NAVi database, and window based weights updating strategy, the conceptual intelligent navigator has the ability to generate and accumulate the navigation knowledge which can be stored in its “brain”. In other words, it can learn and evolve continually to provide updated navigation knowledge and fill the gap between GPS outages.

6.4 Performance Analysis of the Conceptual Intelligent Navigator

In order to evaluate the performance of the conceptual intelligent navigator, two field tests were conducted in August 2002 using a land vehicle and the NovAtel BDS[®] GPS/IMU system (IMU-Honeywell HG1700, tactical grade IMU). The specifications of HG1700 are given in Appendix C. GPS signals with a minimum of 7 satellites were available throughout the test periods. The objective was to evaluate the performance of ANN based INS/GPS integration architectures, NAVi, and the window based weight updating strategy, as they are the core components of the conceptual intelligent navigator.

■ Prediction by NAVi

The duration of the first test was about 5800 seconds and no GPS signal outage periods were intentionally introduced for the purpose of using trajectory to obtain the stored

weights for the estimation of the second field test. The first trajectory is shown in the Figure (6.18). The duration of the second test was about 2500 seconds and a long GPS signal outage period was intentionally introduced for the whole navigation period (i.e., 2500 second). The second trajectory is shown in Figure (6.19).

PUA and PVUA were implemented as the INS/GPS integration architectures for the intelligent navigator. Both IMU and GPS measurements obtained during the 1st field test were fed into the PUA and PVUA to obtain navigation knowledge (stored synaptic weights) for the 2nd field test. Thus, the NAVi was applied to fill the gap between DGPS signal outages during the second field test. In order to compare the performance of the conceptual intelligent navigator against conventional integration techniques, the results obtained from the PUA, PVUA and Kalman filter were compared with a reference trajectory generated by DGPS (with accuracy $\approx 10\text{cm}$). The results of the Kalman filter were obtained from the University of Calgary Kalman filter DGPS/INS integration software (KINGSPADTM - KINematic Geodetic System for Positions and Attitude Determination). GPS outages with only the last 900 seconds were introduced for the Kalman filter INS/DGPS integration scheme which meant that the DGPS signal was available during the first 2000 seconds for the Kalman filter computation.

It can be noticed from Figure (6.20) that the position errors in a single direction (either North or East) at the output of the Kalman filter INS/DGPS integration scheme accumulated up to 400 metres within 900 seconds of GPS outage. In contrast, both PUA and PVUA provided stable solutions along the whole test of 2900 seconds with RMS errors of 31 metres and 21 metres, respectively, as shown in Table 6.2.

PUA showed adequate performance for the first 2000 seconds and then the position errors increased with time during the last 900 seconds. Uncompensated gyroscopes and accelerometer measurements dominated the time growing position errors along both the North and East directions. Although the vehicle maneuvering or dynamic conditions resulted in several error peaks or oscillations, both PUA and PVUA demonstrated the ability to reduce the impact of time growing errors in the long term. Table 6.2 also shows that PVUA demonstrated a 35% improvement in terms of RMS position errors when

compared to PUA. The above results demonstrated the prediction ability of proposed architectures. Similar to the conclusion obtained in section 5.4, the performance of the Kalman filter was mainly related to the length of the GPS outage. On the other hand, the results demonstrated the positioning accuracy of the conceptual intelligent navigator was mainly affected by the vehicle's motion dynamics and not on time from the beginning of the GPS outage. It is a significant difference and improvement in comparison with time growing error characteristics of the Kalman filter.

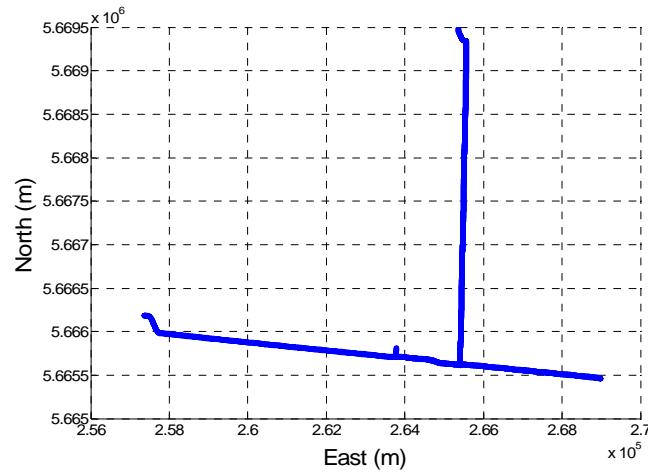


Figure 6.18: Trajectory of 1st field test

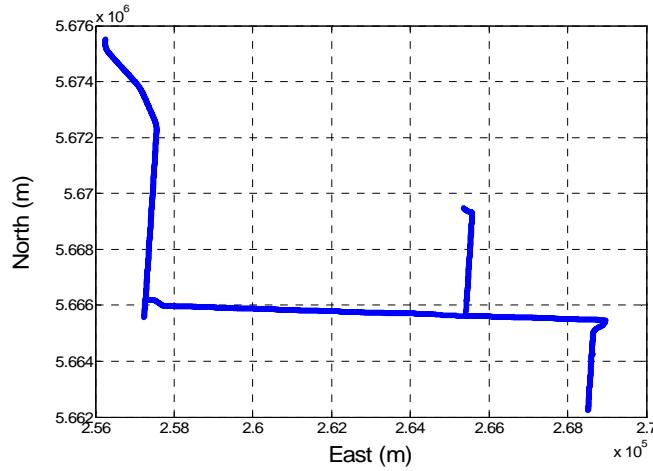


Figure 6.19: Trajectory of 2nd field test

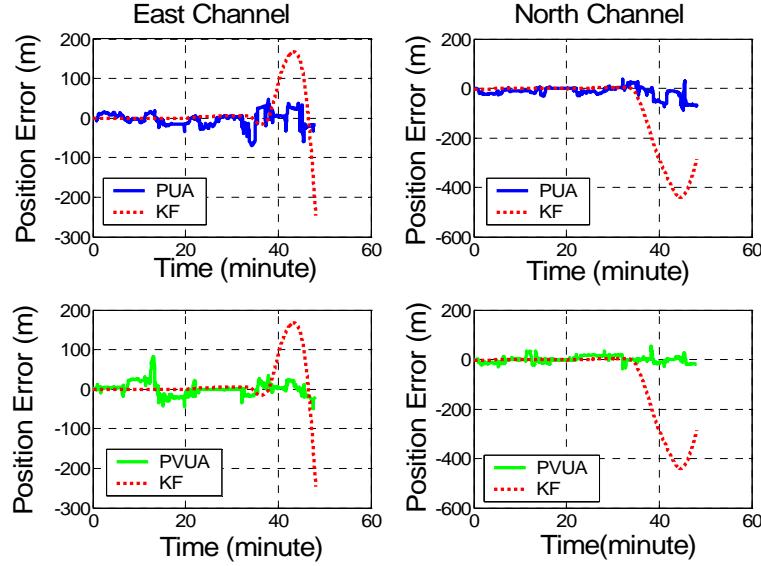


Figure 6.20: Position Error of 2nd field test

Table 6.2: RMSE (2nd field test)

	North(m)	East(m)	Overall(m)
PUA	25.10	18.77	31.33
PVUA	12.14	16.46	20.45
KF	300.10	102.09	316.99

■ Performance of the window based weight updating strategy

To evaluate the performance of the window based weight updating strategy, four different window sizes were considered (10s, 20s, 40s and 60s). In addition, 10 windows were implemented with stored weights obtained via the 1st field test. Data were then updated utilizing the proposed method and 2nd field test data with the availability of a GPS signal. In case of a GPS outage, the latest updated weights were applied to provide real time prediction, as shown in Figure (6.21). Since there were four different sizes of windows, the simulated GPS outage for each scenario was different, as shown in Table 6.3. One and two step training procedures were applied for the PUA and PVUA, respectively. The results demonstrated that the utilization of the proposed method to update the previously stored weights can improve the prediction accuracy of the 2nd field

test by 60% for PUA and 50% for PVUA. In general, the positioning accuracy improved with longer GPS signal windows (e.g. 20% between 10s and 60s using PVUA).

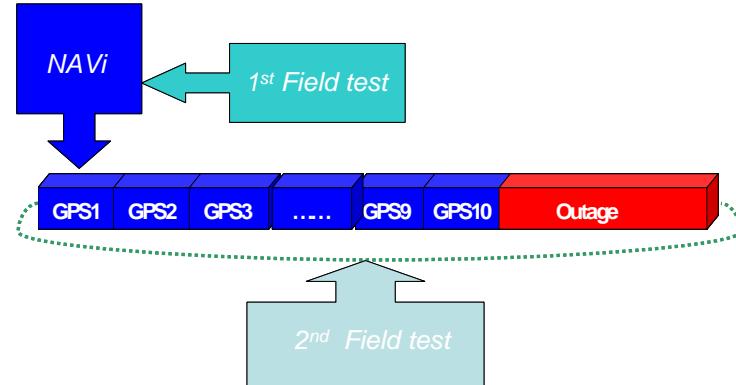


Figure 6.21: Window-based weights updating strategy

Table 6.3: RMSE (2nd field test)

	North(m)	East(m)	Overall(m)	GPS outage
KF	300.10	102.09	316.99	900s
PUA	25.10	18.77	31.33	2886s
PVUA	12.14	16.46	20.45	2886s
10s/PUA	5.56	9.32	10.85	2786s
20s/PUA	5.40	5.94	8.02	2686s
40s/PUA	10.46	3.60	11.07	2486s
60s/PUA	3.17	8.04	8.67	2286s
10s/PVUA	4.35	6.22	7.64	2786s
20s/PVUA	3.38	6.93	7.71	2686s
40s/PVUA	3.59	5.26	6.36	2486s
60s/PVUA	3.35	5.13	6.12	2286s

In order to evaluate the relationship between training time and window size, training time required in each of the previous groups is illustrated in Figure (6.22) for PUA and Figure (6.23) for PVUA. As shown in these two figures, the worst-case scenario consumed half the window size. In fact, most of the training time was much smaller than the window size. In addition, the training time required by one and two step training procedures was similar as the size of NAVi was small.

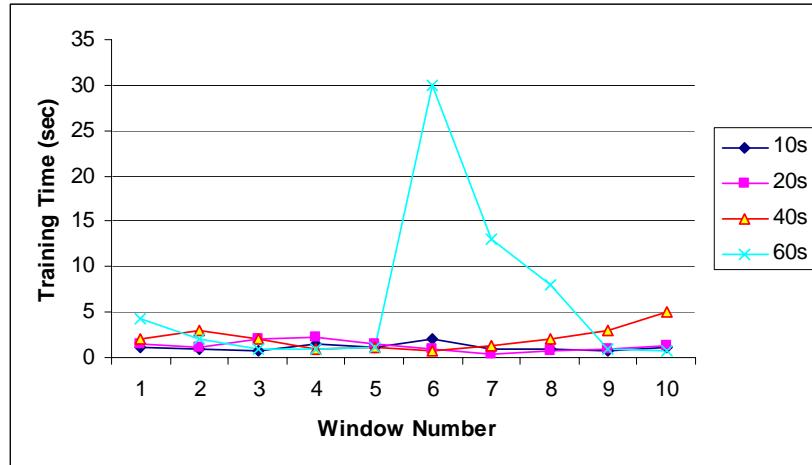


Figure 6.22: Training time for each window (PUA)

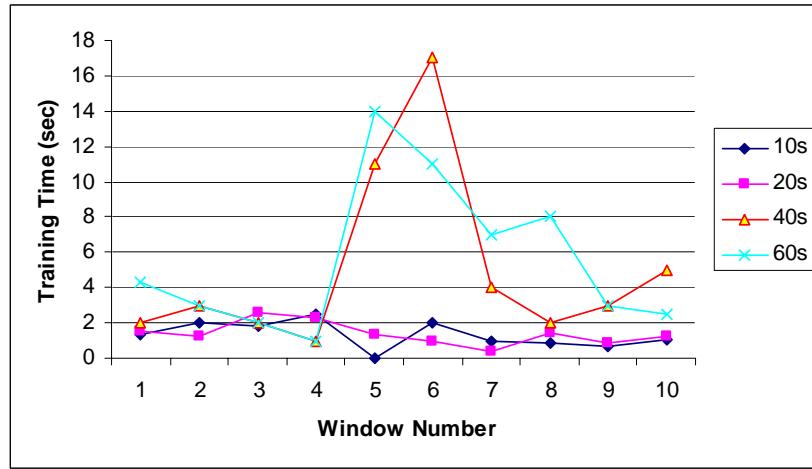


Figure 6.23: Training time for each window (PVUA)

Thus, the window based weights updating strategy with stored weights can provide alternative weight updating algorithms for INS/GPS integration and can improve the position accuracy during GPS outages. The preliminarily results demonstrated the potential to incorporate the conceptual intelligent navigator as the alternative navigation algorithm for next generation land vehicular navigation systems as it can overcome or reduce the limitations of conventional Kalman filtering.. Several issues such as the impact of IMU quality on the conceptual intelligent navigator and the performance of the conceptual intelligent navigator that uses INS/SPP integrated system will also be investigated in Chapter 7.

CHAPTER 7

RESULTS AND DISCUSSIONS

7.1 Introduction

This chapter describes the tests and analysis of the methods proposed in the previous chapters. Three field tests were conducted on October 2003 by the Mobile Multi-sensor Systems (MMSS) research Group of the University of Calgary. The tests were conducted in land vehicle environments using three different INS/GPS integrated systems consisting of a navigation grade IMU (Honeywell CIMU), a tactical grade IMU (Litton LN-200), a MEMS IMU (Crossbow AHRS-400 CC, XBOW) and two NovAtel OEM-4 receivers. The performance of the cascade denoising algorithm and conceptual intelligent navigator were evaluated in terms of IMUs' quality (i.e., accuracy level). The specifications of those IMUs are given in appendix C. Figure (7.1) shows the test van provided by Novatel Inc. and the set up of these IMU systems. Table 7.1 summarizes the different tests in terms of shape of trajectory, dynamics, and satellite availability.



Figure 7.1: Picture of test van

Table 7.1: The Summary of different field tests

Field test	Trajectory	Motion dynamic	Satellite availability
1	<ul style="list-style-type: none"> • Shape: L-shape • The duration: 3800 seconds • Baseline: 4km • Travel distance: 32km • Data rate: 1 Hz 	<ul style="list-style-type: none"> • Acceleration and deceleration along north-south and west-east directions and several U-turns. • Rich information about velocity variation 	No natural GPS signal blockage or obstruction
2	<ul style="list-style-type: none"> • Shape: Circle / Rectangular • The duration: 1900 seconds • Baseline: 3km • Travel distance: 8.5km • Data rate: 1 Hz 	<ul style="list-style-type: none"> • Acceleration and deceleration along a rectangle in clockwise and counterclockwise direction • Several straight line segments, a big U-turn and two circles in clockwise and counterclockwise direction. • Significant information associated with velocity variation and azimuth variation 	No natural GPS signal blockage or obstruction
3	<ul style="list-style-type: none"> • Shape: Complex trajectory • The duration: 1200 seconds • Baseline: 15km • Distance traveled: 15km • Data rate: 1Hz 	<ul style="list-style-type: none"> • The motion dynamic variation of the vehicle was not complicated in comparison with previous field test • Real life scenario 	<ul style="list-style-type: none"> • Six natural GPS outages that includes the impact from intermittent signal reception (urban canyon) or no signal reception (underground) • The length of those GPS outages was four seconds, six seconds, five seconds, nine seconds, ten seconds and fourteen seconds, respectively.

The detailed spatial relationship (lever-arm) between the IMUs and GPS antenna is illustrated in Figure (7.2a). In addition, Figure (7.2b), Figure (7.2c) and Figure (7.2d) illustrate the trajectories of these field tests. The blue solid lines in these figures illustrate the trajectory generated by CIMU/DGPS integrated solutions and the red dot lines show

the trajectory generated by the DGPS solutions using carrier phase measurements. The GPS measurements were processed using an independent GPS post-processing software GrafNav™ 7.0 (Waypoint Consulting Inc.) in DGPS and SPP mode. After that, the GPS navigation solutions were then fed into a decentralized Kalman filter (INS filter) implemented in the INS Toolbox [Shin and El-Sheimy, 2004] or KINGSPAD™ [Schwarz and El-Sheimy, 1999] developed by the MMSS research group to obtain INS/GPS integrated solutions for further analysis. In the mean time, the IMU measurements were processed in INS/GPS integrated mode and stand-alone mode for further analysis. The reference trajectories were generated by the CIMU/DGPS integrated system.

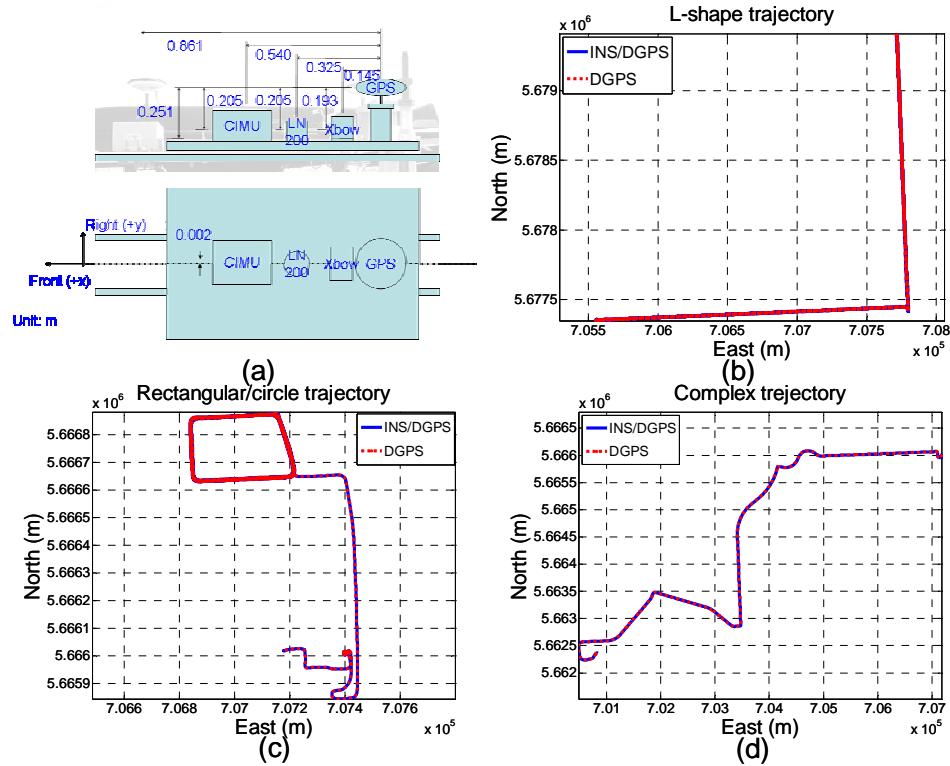


Figure 7.2: The trajectories of different field tests

In this chapter, the performance and impact of the proposed cascade denoising algorithm (Chapter 4) on the different INS/GPS integrated systems are evaluated in terms of the position errors during several GPS outages. Then, the performance of the proposed

conceptual intelligent navigator is evaluated using the denoised INS data and the integrated systems in terms of the position errors during GPS outages. Before leaving this chapter, an additional low cost MEMS IMU/GPS integrated system that includes a MEMS IMU sensor triad developed by the MMSS research group is applied to enhance the potential of implementing the proposed conceptual intelligent navigator as an alternate to the tradition Kalman filter solutions for developing next generation land vehicular navigation systems.

7.2 Performance Analysis of the Cascade Denoising Algorithm

To evaluate the performance of the cascade denoising algorithm in terms of the position errors during GPS outages, the first field test was applied. As mentioned previously, there was no natural GPS outage in this trajectory; thus, eight simulated GPS outages were generated by removing the GPS solutions being fed into the INS Kalman filter during the simulated GPS outages, as shown in Table 7.2 and Figure (7.3). This meant the INS Kalman filter operated in prediction mode during simulated GPS outages. Table 7.2 also summarizes the length and the motion dynamic variation of the simulated GPS outages.

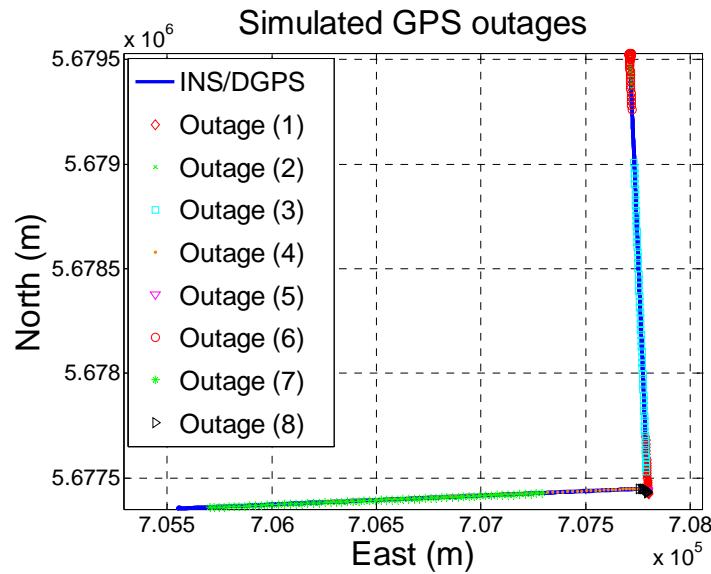


Figure 7.3: Simulated GPS outages

Table 7.2: Background information of simulated GPS outages

	1	2	3	4	5	6	7	8
Length (second)	30	30	60	60	60	30	60	120
Motion	<i>Straight line (north)</i>	<i>U- turn</i>	<i>Straight line (north)</i>	<i>Straight line (west)</i>	<i>static</i>	<i>Sharp turn</i>	<i>Straight line (east)</i>	<i>Sharp turn and static</i>
Distance (m)	612	123	1225	1230	0	352	1212	25

The navigation solutions obtained using the raw IMU measurements and denoised IMU measurements were then compared with the reference trajectories. In addition, the INS/DGPS and INS/SPP integrated mode were also investigated for each INS/GPS integrated system, respectively.

7.2.1 Performance Summary of Different Integrated Systems

Figure (7.4) and Table 7.3 illustrate the performance summary of the positional errors generated by comparing the raw and denoised INS (Honeywell CIMU)/DGPS integrated navigation solutions with the reference trajectory during each GPS outage period. It should be mentioned here that the maximum of the absolute position error along the North and East directions (MAX_N and MAX_E) always happen at the end of the blockage periods.

Table 7.3: Performance summary of positional errors (CIMU/DGPS)

Blockage no.	Blockage Length(s)	RAW			RMSE Total(m)	Cascade Denoising _DGPS		
		MAX_N (m)	MAX_E (m)	RMSE Total(m)		MAX_N (m)	MAX_E (m)	RMSE Total(m)
1	30	0.1	1.13	0.43	0.02	0.05	0.03	
2	30	1.30	0.12	0.44	0.28	0.02	0.17	
3	60	0.45	1.06	0.55	0.37	1.34	0.68	
4	60	2.39	0.85	1.18	0.67	0.75	0.48	
5	60	0.84	0.59	0.50	0.56	0.10	0.37	
6	30	0.24	0.15	0.14	0.38	0.06	0.19	
7	60	1.72	1.48	1.06	0.52	0.39	0.44	
8	120	2.28	3.08	1.62	1.07	0.41	0.48	

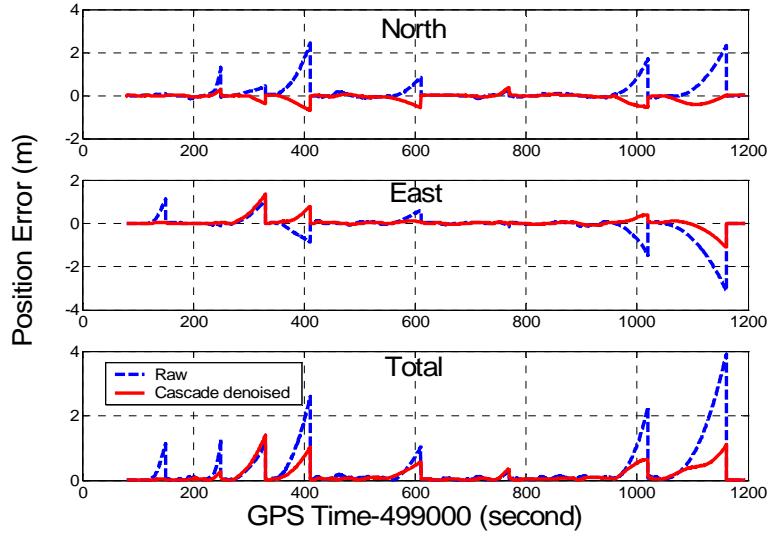


Figure 7.4: Position errors (CIMU/DGPS)

It can be seen from Figure (7.4) that the utilization of the cascade denoising algorithm provided visible improvements during several simulated GPS outage periods. As indicated in Table 7.3, the position errors of six GPS outage periods were successfully reduced using denoised CIMU measurements. The rate of improvement was 75% (6/8) which is similar to the results presented in the chapter 4.

In addition, the magnitude of improvement ranged from 20 centimetres to 1 metre and the percentage of improvement ranged from 20% to 90%. In contrast, the remaining two GPS outage periods that were not improved by the cascade denoising algorithm, were not degraded significantly. The magnitude of degradation ranged from 5 centimetres to 20 centimetres and the percentage of degradation ranged from 20% to 35%.

Figure (7.5) and Table 7.4 illustrate the performance summary of the positional errors generated by comparing the raw and denoised INS (Honeywell CIMU)/SPP integrated navigation solutions with the reference trajectory during each GPS outage period respectively, where MAX_N and MAX_E represent the maximum of absolute position error along North and East directions, respectively. Being limited by positioning accuracy

of SPP, the positional errors generated by the INS/SPP integrated navigation solution ranged from 1 to 7 metres, as indicated in Figure (7.5).

Table 7.4: Performance summary of positional errors (CIMU/SPP)

Blockage no.	RAW				Cascade Denoising _SPP		
	Blockage Length(s)	MAX_N (m)	MAX_E (m)	RMSE Total(m)	MAX_N (m)	MAX_E (m)	RMSE Total(m)
1	30	5.21	1.84	5.20	4.95	2.28	5.12
2	30	6.04	2.37	5.50	5.61	2.02	5.69
3	60	20.14	3.99	13.08	6.74	2.95	7.32
4	60	7.09	5.06	7.42	5.45	2.98	6.03
5	60	2.27	2.17	2.64	1.55	2.11	2.22
6	30	1.16	1.28	1.8	2.20	1.60	1.88
7	60	3.73	2.99	3.89	2.68	2.59	3.37
8	120	5.33	5.07	4.69	3.30	2.02	3.50

Figure (7.5) indicates that the improvements introduced by using the cascade denoising algorithm during several GPS outage periods were significant enough to be visualized. Similarly, Table 7.4 indicates that the positional errors of six GPS outage periods were successfully reduced by denoising the CIMU measurements. The rate of improvement was 75% (6/8).

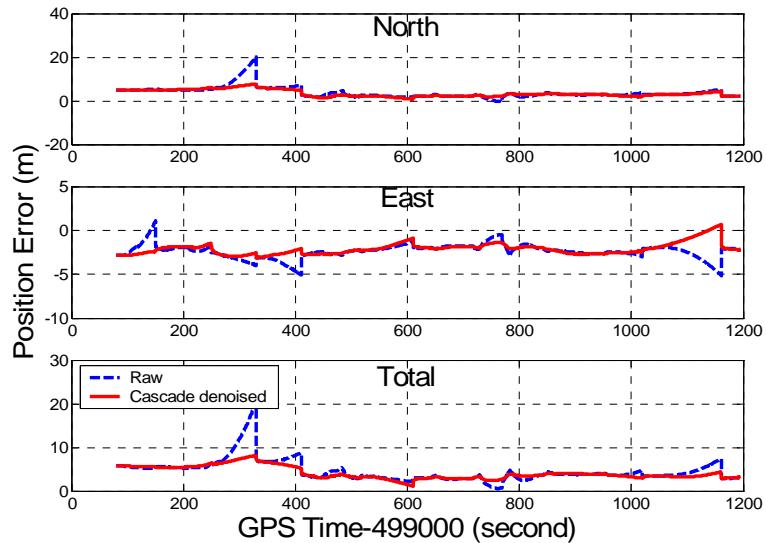


Figure 7.5: Position errors (CIMU/SPP)

In addition, the magnitude of improvement ranged from 8 centimetres to 7 metres and the percentage of improvement ranged from 20% to 90%. In contrast, the magnitude of degradation associated with the remaining two GPS outage periods ranged from 19 centimetres to 60 centimetres and the percentage of degradation ranged from 20% to 35%. Similar analyses were conducted to investigate the impact of the cascade denoising algorithm on positional errors for the LN200 and XBOW IMUs for the two cases GPS operations (i.e., DGPS and SPP). For more details about the results, see Appendix A.5 and A.6, respectively. Tables 7.5 and Table 7.6 summarize the performance summary of the positional errors for the four cases for each system respectively (IMU Raw with DGPS and SPP; IMU denoised with DGPS and SPP).

Figure (7.6) compares the impact of the proposed cascade denoising algorithm on the INS/DGPS and INS/SPP integrated systems in terms of the percentage of improvement or degradation during each GPS outage period using a CIMU/GPS integrated system. In general, the INS/DGPS integrated system provided more accurate prediction accuracy than the INS/SPP integrated system. Similar impact of the proposed cascade denoising algorithm on the LN200/GPS and XBOW/GPS integrated systems are given in Figure (A.10) and Figure (A.13), respectively.

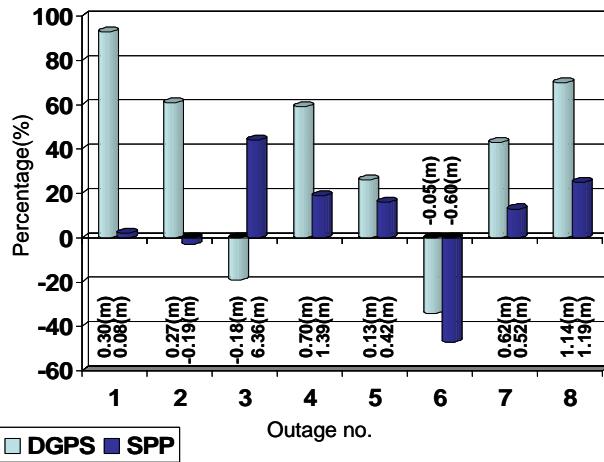


Figure 7.6: The impact of denoising on the integrated systems (CIMU/GPS)

Table 7.5: Performance summary of positional errors (LN200/GPS)

LN200		RMSE (m) DGPS		RMSE (m) SPP	
Blockage no.	Blockage Length(s)	Raw IMU	denoised IMU	Raw IMU	denoised IMU
1	30	6.12	6.85	15.96	15.08
2	30	4.83	4.90	5.51	7.32
3	60	2.84	1.92	27.01	18.10
4	60	12.95	13.56	24.23	20.97
5	60	2.99	2.77	22.78	7.10
6	30	2.20	1.68	5.58	7.04
7	60	14.28	3.92	13.42	10.52
8	120	18.28	8.91	45.39	31.64

Table 7.6: Performance summary positional errors (XBOW/GPS)

XBOW		RMSE (m) DGPS		RMSE (m) SPP	
Blockage no.	Blockage Length(s)	Raw IMU	denoised IMU	Raw IMU	denoised IMU
1	30	12.44	15.12	22.78	23.35
2	30	19.06	8.31	18.53	6.81
3	60	86.90	15.50	103.70	28.21
4	60	15.36	19.24	61.63	62.84
5	60	45.64	44.33	128.87	65.93
6	30	11.62	8.13	64.03	8.36
7	60	42.59	21.49	27.11	27.13
8	120	172.47	137.28	220.08	163.18

7.2.2 Performance Analysis Index (PAI) for Different IMUs

To provide more specific descriptions associated with the performance of the cascade denoising algorithm, a performance analysis index (PAI) was defined as

$$PAI(N, E) = 1 - \frac{\sum_{i=1}^{TP} |\delta \hat{N}_i, \delta \hat{E}_i|}{\sum_{i=1}^{TP} |\delta N_i, \delta E_i|}, \quad TP = t * F_s, t = 450s$$

Where TP is the total number of accumulated point during all GPS outages (t is the total length of all GPS outage and Fs is the sampling rate of IMU), $|\delta\hat{N}_i, \delta\hat{E}_i|$ and $|\delta N_i, \delta E_i|$ are the accumulated absolute magnitude of position errors along North and East directions during each GPS outage period after and before applying the cascade denoising algorithm, respectively. Table 7.7 illustrates the PAIs for both CIMU/DGPS and CIMU/SPP integrated systems.

As indicated in Table 7.7, the PAIs of both integrated systems demonstrated improvements after using the proposed algorithm. In addition, the absolute position error accumulations in both cases were reduced after applying the cascade denoising algorithm. Thus, despite the minor degradation during some outages, the cascade denoising algorithm was effective in improving the positioning accuracy of an INS/GPS integrated system for a navigational grade IMU in both DGPS mode and SPP mode during several GPS outages of different lengths and motion dynamics.

Table 7.7: PAIs (CIMU/GPS)

CIMU	RAW		Cascade denoising			
	$\sum \delta E (\text{m})$	$\sum \delta N (\text{m})$	$\sum \delta\hat{E} (\text{m})$	$\sum \delta\hat{N} (\text{m})$	PAI(E)	PAI(N)
INS/DGPS	7.5410E4	6.2211E4	2.9063E4	2.8992E4	62%	54%
INS/SPP	5.3149E5	5.2345E5	1.728E5	3.4422E5	67%	34%

Similarly, Tables 7.8 and 7.9 list the PAIs for the LN200 and Xbow cases, respectively. Both tables demonstrate the improvements after using the proposed algorithm. In addition, the absolute position error accumulations in both cases were reduced after applying the cascade denoising algorithm. Thus, despite the minor degradation during some outages, the cascade denoising algorithm was effective at improving the positioning accuracy of an INS/GPS integrated system for a tactical grade or a MEMS IMU in both DGPS mode and SPP mode during several GPS outages of different lengths and motion dynamics.

Table 7.8: PAIs (LN200/GPS)

LN200	RAW		Cascade denoising			
	$\sum \delta E (\text{m})$	$\sum \delta N (\text{m})$	$\sum \delta \hat{E} (\text{m})$	$\sum \delta \hat{N} (\text{m})$	PAI(E)	PAI(N)
INS/DGPS	3.3596E5	5.7804E5	2.7863E5	3.1698E5	17%	45%
INS/SPP	1.1984E6	1.7095E6	7.8848E5	1.0514E6	34%	38%

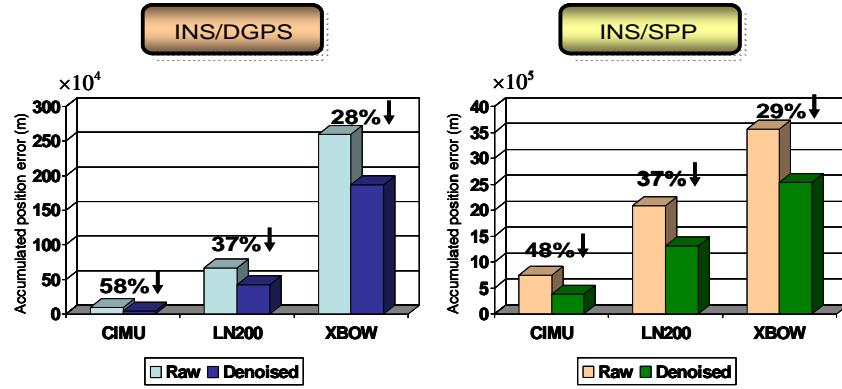
Table 7.9: PAIs (XBOW/GPS)

XBOW	RAW		Cascade denoising			
	$\sum \delta E (\text{m})$	$\sum \delta N (\text{m})$	$\sum \delta \hat{E} (\text{m})$	$\sum \delta \hat{N} (\text{m})$	PAI(E)	PAI(N)
INS/DGPS	1.6997E6	1.9663E6	7.3049E5	1.7197E6	57%	13%
INS/SPP	1.7821E6	3.0888E6	7.4292E5	2.4271E6	58%	21%

To evaluate the impact of the cascade denoising algorithm on different INS/GPS integrated systems, the two dimensional accumulated positional errors for the three IMU systems are illustrated in Figure (7.7). It compares the two dimensional accumulated position errors obtained through the use of raw and denoised IMU measurements generated by different INS/GPS integrated systems. The cascade denoising algorithm provided the most significant improvement for the CIMU/GPS integrated system in DGPS and SPP mode. The LN200/GPS integrated system provided lesser improvement; however, the percentage of improvement approached 40% in DGPS and SPP mode. The XBOW/GPS integrated system provided the least amount of improvement among these the three systems; however, the percentage of improvement reached 30% in DGPS and SPP mode.

The percentage of improvement indices shown in Figure (7.7) can be transformed to a common scale (e.g. for the CIMU $0.47 = 0.58 / (0.58+0.37+0.28)$ for the DGPS and $0.42=0.48 / (0.48+0.37+0.29)$ for the SPP). Thus, the impact of the cascade denoising algorithm on different INS/GPS integrated systems can be quantized, as shown Figure (7.8). This describes the portion or the percentage of position error that can be reduced by the cascade denoising algorithm. For example, given a 1 metre positional error for raw IMU data, the cascade denoising algorithm will be able to remove 47 centimetres, 30 centimetres and 23 centimetres with respect to CIMU/DGPS, LN200/DGPS and

XBOW/DGPS integrated systems. In other words, the benefit of the cascade denoising algorithm decreases with an increasing quality of IMU.



$$\sum |\delta P| = \sqrt{(\sum |\delta E|)^2 + (\sum |\delta N|)^2}$$

Figure 7.7: The impact of cascade denoising algorithm

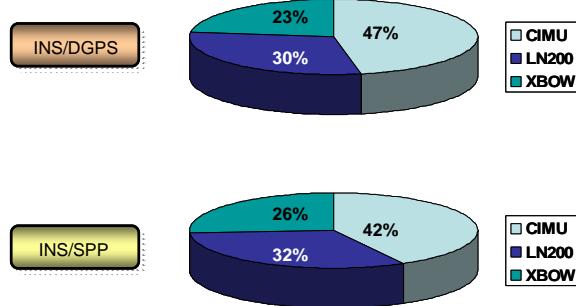


Figure 7.8: The quantized impact of cascade denoising algorithm

For a navigation grade IMU (e.g., CIMU), the long term errors (i.e., bias, drifts) are stable and well behaved, thus the remaining short term errors of the IMU account for most of the residual position error during a GPS outage. However, for a MEMS IMU, due to its poor stability of long term errors, the impact of short term errors is not as significant. In addition, most of the low cost MEMS IMUs have analogous low pass filters whose cutoff frequencies range from 30 Hz to 40 Hz to limit the bandwidth of their raw outputs and reduce the impact of unwanted high frequency noise. Consequently, the

improvement of the cascade denoising algorithm is limited. The long term errors dominate the residual position errors for current MEMS IMUs, however, the use of the cascade denoising algorithm is still able to improve the positioning accuracy during GPS outages by reducing the impact of short term errors.

7.3 Performance Analysis of the Conceptual Intelligent Navigator

Having the quality of IMU measurements improved by using the cascade denoising algorithm, the next step is to evaluate the performance of the conceptual intelligent navigator that consists of different INS/GPS integration architectures. The IMU and GPS measurements obtained through the first and second field tests using the above mentioned INS/GPS integrated systems were applied to generate the stored navigation knowledge. After that, the third field test was used as the test trajectory. To enhance the difference between the conceptual intelligent navigator and the traditional navigator that consisted of a 15 state extended Kalman filter, a long GPS signal outage of 20 minutes was simulated during the third filed test. In other words, both navigators received the IMU measurements and operated in prediction mode without any GPS information updates during the test trajectory. Then, the results predicted by both navigators were compared with the reference trajectory for further analysis.

To integrate IMU and DGPS measurements, a PVAUA and PVUA were implemented as the INS/DGPS architecture of the conceptual intelligent navigator for each INS/GPS integrated system, respectively. In contrast, a PUA was implemented to integrate IMU measurements and GPS measurements in either DGPS or SPP mode. In other words, the conceptual intelligent navigator and traditional navigator were evaluated using different INS/GPS integrated systems. The performance analysis associated with each navigator is presented in subsequent sections.

7.3.1 CIMU/GPS Integrated System

Table 7.10, Figure (7.9a) and Figure (7.9b) compare the accuracy of different azimuth measurements obtained through the first and second field test. Where V, P, INS, CV and CP represent raw DGPS velocity derived azimuth, raw DGPS position derived azimuth, INS generated azimuth in stand-alone mode, constrained DGPS velocity derived azimuth, and constrained DGPS position derived azimuth, respectively. Due to the superior quality of the CIMU, it was capable of providing the most accurate azimuth measurements even when it was operated in INS stand-alone mode for 1 hour (1st field test) and half an hour (2nd field test).

Table 7.10: RMSE of different azimuth measurements (CIMU/DGPS)

	V (deg)	P (deg)	INS (deg)	CV(deg)	CP(deg)
1 st	80.67	74.72	0.13	5.36	3.96
2 nd	59.78	57.70	0.33	5.93	2.24

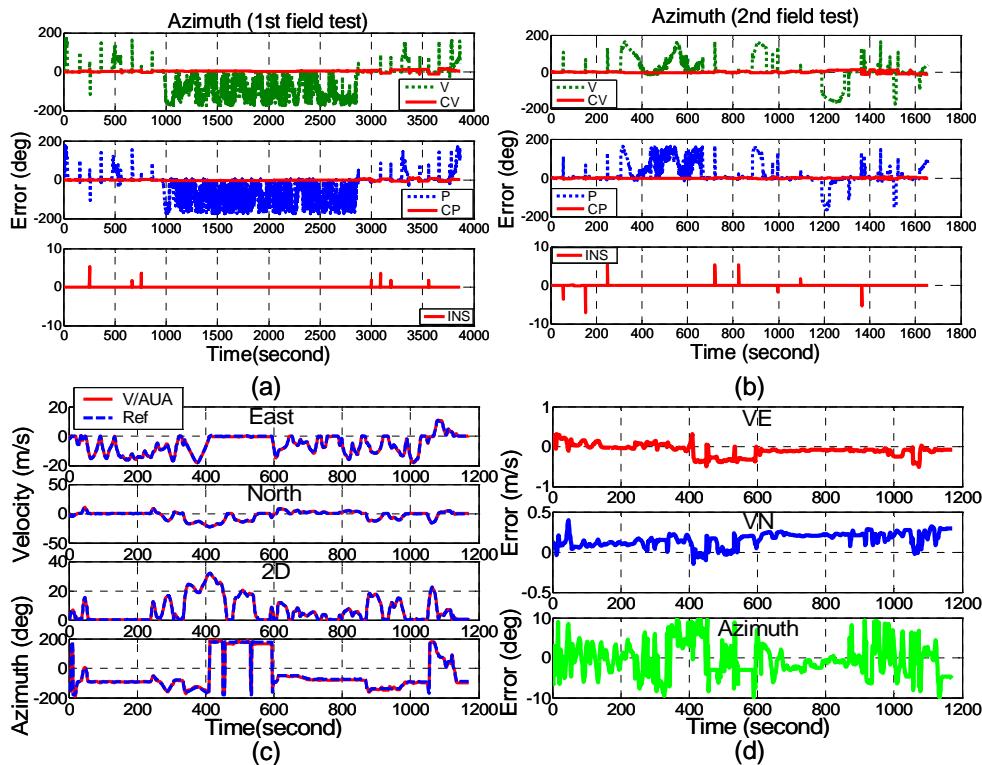


Figure 7.9: Performance of CP, VUA and AUA (CIMU/DGPS)

Comparing the raw DGPS derived azimuth measurements with constrained DGPS derived azimuth measurements, the constrained DGPS position derived azimuth measurements (CP) were the most accurate among the different azimuths generated by a “pseudo” azimuth sensor. Thus, it was applied as the desired outputs for the AUA. However, as indicated in Table 7.10, the constrained DGPS position derived azimuth measurements obtained during the 1st and 2nd field test were less accurate than the inputs of the AUA, the INS generated azimuth. Thus, the conceptual intelligent navigator using a PVAUA with a navigation grade IMU might not be able to provide superior performance than the intelligent navigator using a PVUA or a PUA. The impact of PVAUA using less accurate azimuth measurements on position error is presented later in this section.

After being trained using the IMU and GPS measurements obtained during the 1st and 2nd field tests, the conceptual intelligent navigators were then applied to provide real time prediction using the stored navigation knowledge without retrieving newly updated navigation knowledge during the 3rd field test. Table 7.11, Figure (7.9c) and Figure (7.9d) demonstrate the performance summary of the VUA and AUA outputs. In general, VUA was capable of providing 0.17 m/s and 0.18 m/s velocity RMS error along East and North directions for 20 minutes in prediction mode. In contrast, the velocity RMS errors generated by INS were 0.29 m/s and 0.23 m/s, respectively. As shown in Figure (7.9c), the stored navigation knowledge in the PVAUA was capable of compensating the time growing velocity errors and providing stable velocity outputs using a navigation grade IMU. In addition, the AUA was capable of providing 3.88 degrees azimuth RMS error in real time prediction mode for 20 minutes. In contrast, the RMS error of the INS generated azimuth measurement was 0.3 degrees. Therefore, the PVAUA was not capable of providing superior azimuth output using a navigation grade IMU operated in stand-alone mode due to the limitations of the constrained DGPS position derived azimuth. However, it was able to provide more accurate velocity outputs than a navigation grade IMU operated in stand-alone mode for 20 minutes.

Table 7.11: Performance summary of VUA and AUA (test trajectory, CIMU/DGPS)

	ER_VE_VUA	ER_VN_VUA	ER_Azimuth_AUA
RMS	0.17 m/s	0.18 m/s	3.88 deg
Max	0.52 m/s	0.41 m/s	9.98 deg

Table 7.12 and Figure (7.10) illustrate the performance summary of the positional errors generated by each navigator. Based on the field test data applied in this research, the traditional Kalman Filter navigator that used a navigation grade IMU was capable of providing 138 metres total position error in real time prediction mode for 20 minutes. In addition, it was capable of providing superior performance than some of the intelligent navigators when the GPS outage period was less than 5 minutes. As illustrated in Figure (7.10), the major characteristic of the traditional navigator was that its positional errors grew with time.

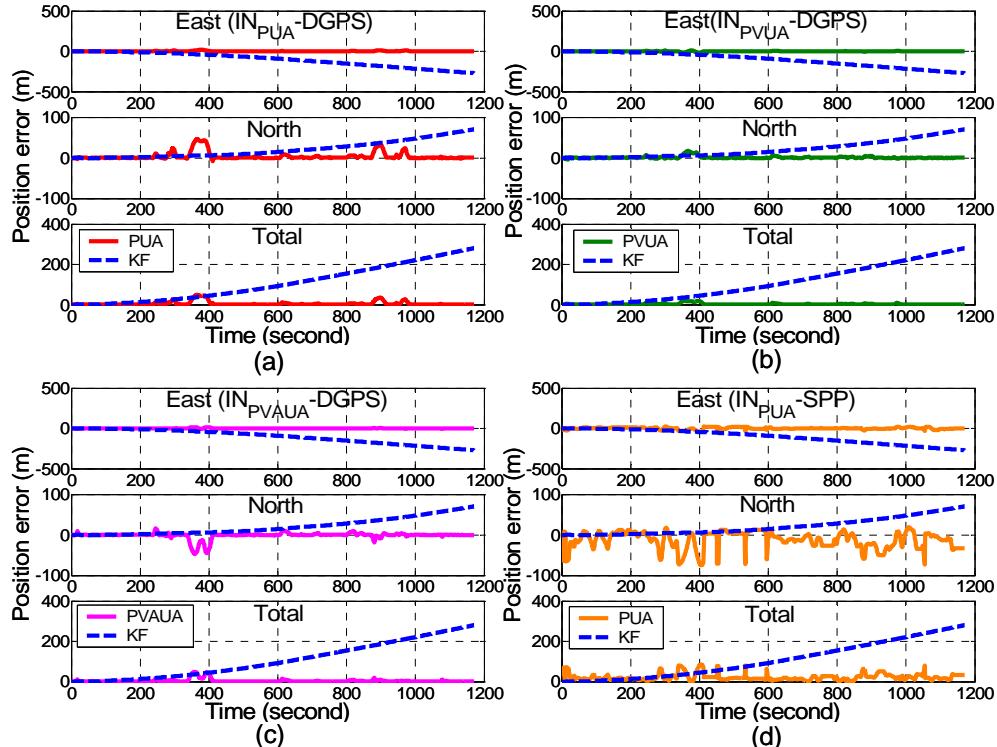


Figure 7.10: Position errors (CIMU/GPS)

Table 7.12: Performance summary of different navigators (test trajectory, CIMU/GPS)

	Length	Max_E	RMS_E	Max_N	RMS_N	Total
IN_PUA	1172s	17.65	3.87	46.20	10.17	10.88
IN_PVUA	1172s	16.88	3.20	17.35	2.96	4.40
IN_PVAUA	1172s	16.11	2.94	46.72	8.33	8.83
IN_PUA_SPP	1172s	41.88	12.57	75.10	24.75	27.76
KF	1172s	271.85	134.36	70.55	29.21	137.50

Figures (7.10a), (7.10b) and (7.10c) illustrate the positional errors generated by the conceptual intelligent navigators using a PUA, PVUA and PVAUA as INS/DGPS integration architectures, respectively. Based on the field test data applied in this research, the conceptual intelligent navigators were capable of providing 5, 9 and 11 metres total position error in real time prediction mode for 20 minutes, respectively. As mentioned previously, due to the accuracy limitation of the constrained DGPS position derived azimuth measurements, the conceptual intelligent navigator using a PVAUA was unable to provide superior positioning accuracy in comparison with the conceptual intelligent navigator using a PVUA. Recalling Chapter 6, the constrained DGPS position derived azimuth measurements and INS generated azimuth measurements were applied as the input for the PUA implemented in a PVAUA and PVUA, respectively. Given the fact that the accuracy of the INS generated azimuth measurements using a navigation grade IMU were superior to that of the constrained DGPS position derived azimuth measurements, the AUA degraded the accuracy of the INS generated azimuth by using constrained DGPS derived azimuth measurements as the desired output to generate stored navigation knowledge. However, the conceptual intelligent navigator using a PVAUA was capable of providing superior performance to the conceptual intelligent navigator using a PUA and traditional navigator.

Thus, a PVUA is recommended as the INS/DGPS integration architecture to achieve 5 metres level positioning accuracy requirement using a navigation grade IMU for real time prediction that lasts more than 15 minutes. In contrast, a PUA is recommended in terms of the system complexity with the degraded position accuracy requirement (10 metres). In general, the conceptual intelligent navigators discussed so far were able to provide

superior long term positioning accuracy based on the field test data applied in this research (i.e., GPS outage period >5 minutes).

Figure (7.10d) illustrates the position errors generated by the conceptual intelligent navigator using a PUA as the INS/SPP integration architecture. As illustrated in Table (7.12), the conceptual intelligent navigator was capable of providing 30 metres total position error in real time prediction mode for 20 minutes. The difference between Figure (7.10a) and (7.10d) is that the former used DGPS measurements and the later used SPP measurements to obtain the stored navigation knowledge. Thus, being limited by the positioning accuracy of GPS measurements in SPP mode (i.e., 10 metres), the position error generated by the conceptual intelligent navigator using SPP was 3 times larger than that of the conceptual intelligent navigator using the same INS/GPS integration architecture with DGPS measurements. However, it was capable of providing superior long term positioning accuracy in comparison with the traditional navigator (i.e., GPS outage period >5 minutes). Figure (7.11) illustrates the trajectories generated by different navigators.

The trajectory generated by the traditional navigator drifted gradually from the reference trajectory toward the end of the navigation. In contrast, all the trajectories generated by the conceptual intelligent navigators were almost overlapped with the reference trajectory except for some discrepancy segments that were indicated by arrows. In general, those discrepancy segments took place during significant motion dynamic variations (i.e., U-turns or sharp turns). As indicated in Figure (7.11), the number of arrows increased when the positioning accuracy provided by the conceptual intelligent navigator decreased.

Thus, the conceptual intelligent navigators using the stored navigation knowledge obtained by the 1st and 2nd field tests were able to provide real time prediction with superior long term positioning accuracy in comparison with the traditional navigators. In addition, the position errors generated by the conceptual intelligent navigator were mainly affected by the motion dynamic variations in comparison to the time growing error of traditional navigators during GPS outages.

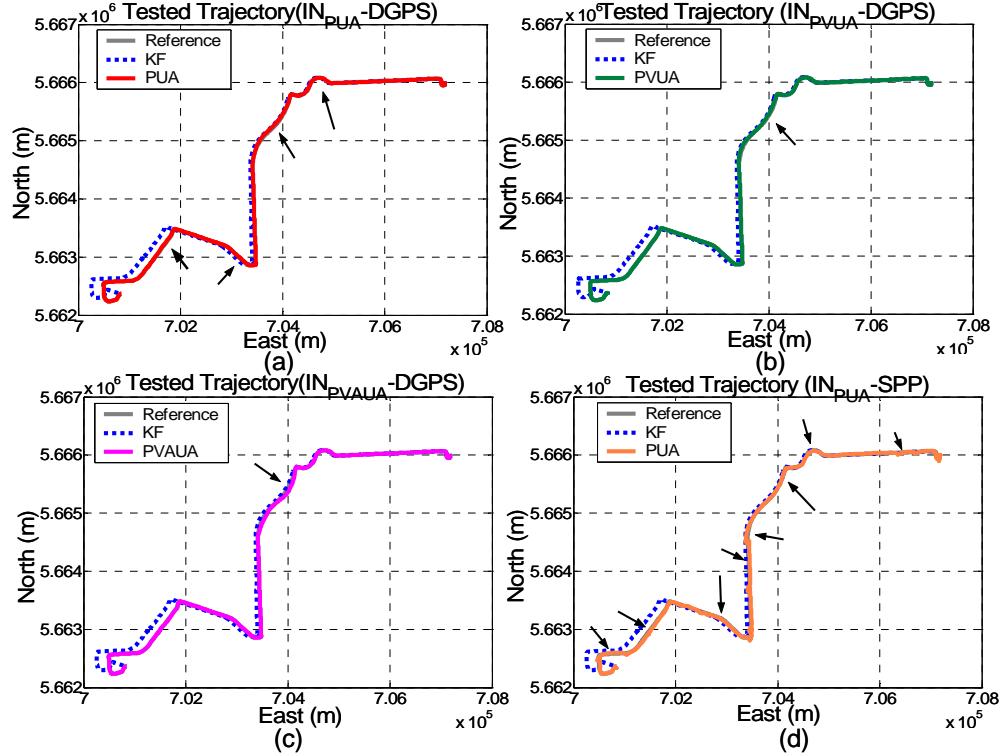


Figure 7.11: Trajectories generated by different navigators (CIMU/GPS)

7.3.2 LN200/GPS Integrated System

Similar analysis, about the accuracy of different azimuth measurements obtained during the first and second field test using a LN200/GPS integrated system, is given as Appendix B.4. As indicated in Appendix B.4, the constrained DGPS position derived azimuth measurements obtained by the 1st and 2nd field tests were more accurate than the input of the AUA, the INS generated azimuth. In fact, the stored navigation knowledge was capable of compensating time growing velocity errors generated by a tactical grade IMU operated in stand-alone mode to provide stable velocity outputs. Consequently, the PVAUA was capable of providing superior velocity and azimuth outputs in comparison with a tactical grade IMU (LN200) operated in stand-alone mode for test trajectory. Therefore, the conceptual intelligent navigator that uses a PVAUA with a tactical grade IMU is expected to be able to provide superior performance than the navigator that uses either a PVUA or a PUA.

Table 7.13 and Figure (7.12) illustrate the performance summary associated with the position errors generated by each navigator. The traditional Kalman Filter navigator that used a tactical grade IMU was capable of providing 2.6 kilometres total position error in real time prediction mode for 20 minutes. Similar to the previous section, it was capable of providing superior performance than some of the intelligent navigators when the GPS outage period was less than 3 minutes. As illustrated in Figure (7.12), the position errors generated by the traditional navigator grew with time.

Table 7.13: Performance summary of different navigator (test trajectory, LN200/GPS)

	Length	Max_E	RMS_E	Max_N	RMS_N	Total
IN_PUA	1178s	75.20	15.02	108.60	28.58	32.29
IN_PVUA	1178s	65.57	21.00	59.81	16.23	26.70
IN_PVAUA	1178s	30.71	8.75	42.13	13.88	16.41
IN_PUA_SPP	1178s	196.23	53.71	374.45	73.12	90.73
KF	1178s	2527.8	1246.2	6181.1	2352.8	2662.43

Figure (7.12a), Figure (7.12b) and Figure (7.12c) illustrate the position errors generated by the conceptual intelligent navigators using a PUA, PVUA and PVAUA as INS/DGPS integration architectures, respectively. As indicated in Table (7.13), the conceptual intelligent navigators were capable of providing 36, 27 and 16 metres total position error in real time prediction mode for 20 minutes, respectively. Unlike the performance observed in the previous section, the conceptual intelligent navigator that used a PVAUA was able to provide the most accurate positioning accuracy in comparison with the conceptual intelligent navigator using a PVUA or a PUA since the constrained DGPS position derived azimuth measurements were more accurate than the INS generated azimuth measurements during the test trajectory.

Consequently, a PVAUA is recommended as the INS/DGPS integration architecture to achieve the 10~15 metre level positioning accuracy requirement using a tactical grade IMU for real time prediction that lasts longer than 15 minutes. In contrast, a PUA is recommended in terms of system simplicity and with a degraded positioning accuracy requirement (30~40 metres). In general, the conceptual intelligent navigators discussed so far were able to provide superior long term positioning accuracy (i.e., GPS outage

period>3 minutes) based on the field test data applied in this research. Figure (7.12d) depicts the position errors generated by the conceptual intelligent navigator using a PUA as the INS/SPP integration architecture.

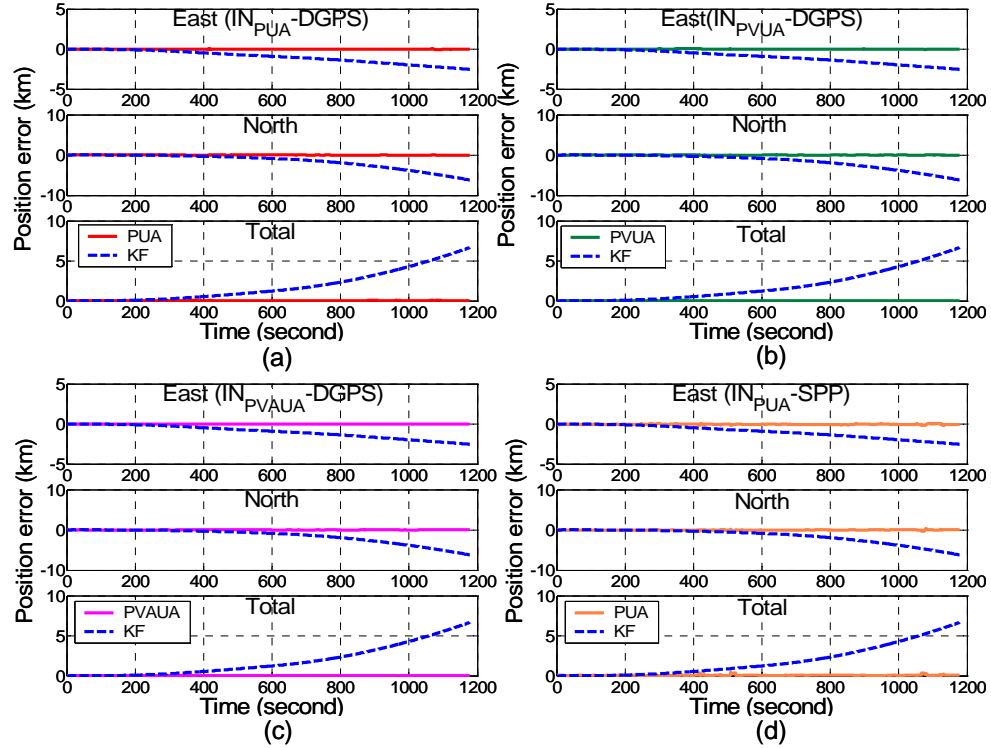


Figure 7.12: Position error (LN200/GPS)

As illustrated in Table 7.13, the conceptual intelligent navigator was capable of providing 90 metres total positioning error in terms of the RMS error for real time prediction that lasted for 20 minutes. Similar to the performance observed in the previous section, the position error generated by the conceptual intelligent navigator that used SPP was 3 times larger than that of the conceptual intelligent navigator that used the same INS/GPS integration architecture with DGPS measurements due to the limitation of the positional accuracy of SPP (i.e., 10 metres). However, it was capable of providing superior long term positioning accuracy in comparison with the traditional navigator (i.e. GPS outage period>3 minutes). Figure (7.13) illustrates the trajectories generated by different navigators.

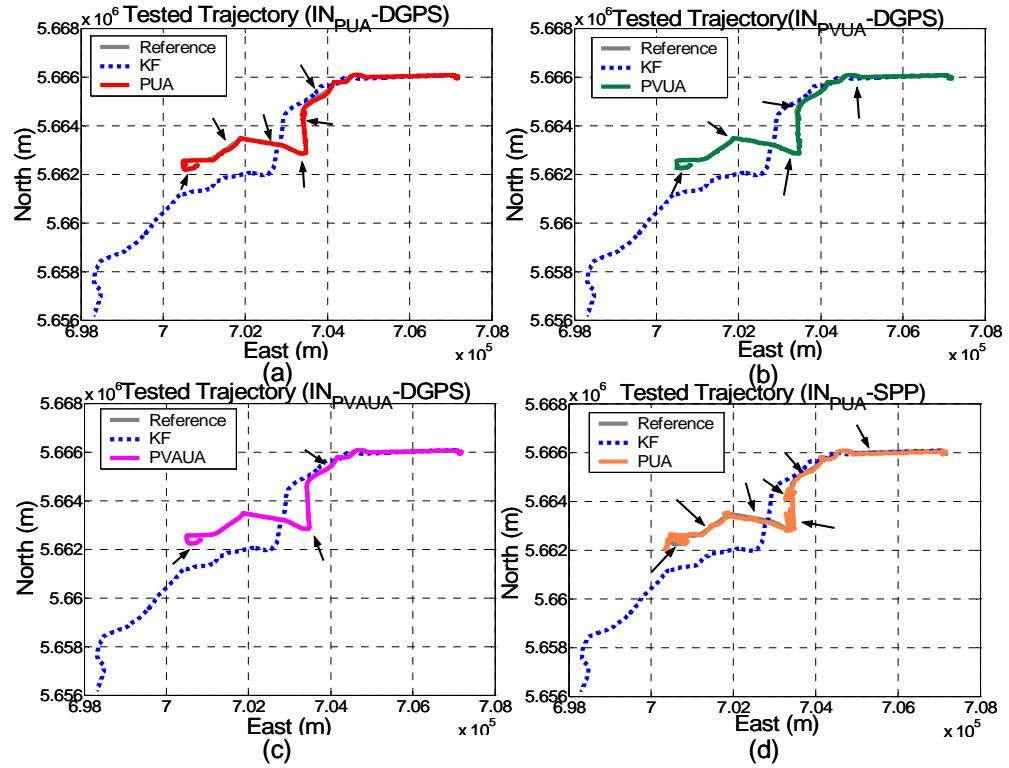


Figure 7.13: Trajectories generated by different navigators (LN200/GPS)

The trajectory generated by the traditional navigator drifted away from the reference trajectory after 3 minutes since the beginning of navigation. In contrast, all the trajectories generated by conceptual intelligent navigator were almost overlapped to the reference trajectory except for some discrepancy segments which are indicated by arrows. In general, those discrepancy segments with visible position errors took place during significant motion dynamic variation (i.e. U-turns or sharp turns). Thus, the conceptual intelligent navigators using the stored navigation knowledge obtained by the 1st and 2nd field tests were able to provide real time prediction with superior long term positioning accuracy compared to the traditional navigator. In addition, the position errors generated by the conceptual intelligent navigator were mainly affected by the motion dynamic variation in comparison to the time growing errors of the traditional navigators during GPS signal blockages.

7.3.3 XBOW/GPS Integrated System

Similar analysis about the accuracy of different azimuth measurements obtained during the first and second field test using an XBOW/GPS integrated system is given as Appendix B.5. As indicated in Appendix B.5, the constrained DGPS derived azimuth measurements were more accurate than the INS generated azimuth measurements when it was operated in INS stand-alone mode for 15 minutes (1st field test) and 8 minutes (2nd field test). The XBOW is a low cost MEMS IMU (\$1500 USD), thus the constrained DGPS velocity derived azimuth measurements and the constrained DGPS position derived azimuth measurements were able to provide improvements in comparison with the INS generated azimuth measurements.

As indicated in Appendix B.5, the constrained DGPS position derived azimuth measurements were the most accurate among those azimuth measurements generated by “pseudo” azimuth sensor and “physical” azimuth sensor (XBOW). Therefore, it was applied as the desired output for the AUA. The conceptual intelligent navigator that used a PVAUA with a MEMS IMU is expected to be able to provide superior performance than other intelligent navigators as it was capable of compensating the error of INS generated azimuth and velocity measurements and providing superior azimuth and velocity outputs in comparison with a stand-alone INS using a MEMS IMU (XBOW).

Table 7.14 and Figure (7.14) illustrate the performance summary associated with the position errors generated by each navigator. The scale of Y-axis in Figure (7.14) was adjusted to provide better visualized comparison between the position errors generated by the conceptual intelligent navigators and traditional navigator. The traditional Kalman Filter navigator provided 100 kilometres total position error in terms of RMS error in real time prediction for 20 minutes. Its position errors along the North and East directions grew beyond the scope of the adjusted scale after 5~8 minutes from the start of navigation. It was, however, capable of providing superior performance than some of the intelligent navigators when the GPS outage period was less than 1 minute. As illustrated in Figure (7.14), the time growing impact on the position errors generated by the

traditional navigator using a MEMS IMU was far more significant than the position errors generated by the traditional navigator using a tactical grade IMU. Due to its poor sensor error stability (i.e., large gyro drifts and accelerometer biases) and noisy behavior, a MEMS IMU was only able to provide accurate navigation solutions within 1 minute of starting navigation without acquiring any updated information from GPS.

Table 7.14: Performance summary of different navigator (test trajectory, Xbow/GPS)

	Length	Max_E	RMS_E	Max_N	RMS_N	Total
IN_PUA	1170s	1438.01	434.78	2817.7	796.86	907.75
IN_PVUA	1170s	1226.7	378.81	466.80	117.56	396.63
IN_PVAUA	1170s	887.24	202.02	588.82	148.29	250.60
IN_PUA_SPP	1170s	3411.4	1352.1	2042.5	700.91	1522.89
KF	1170s	268680.3	85070.5	117420.3	51376.2	99380.17

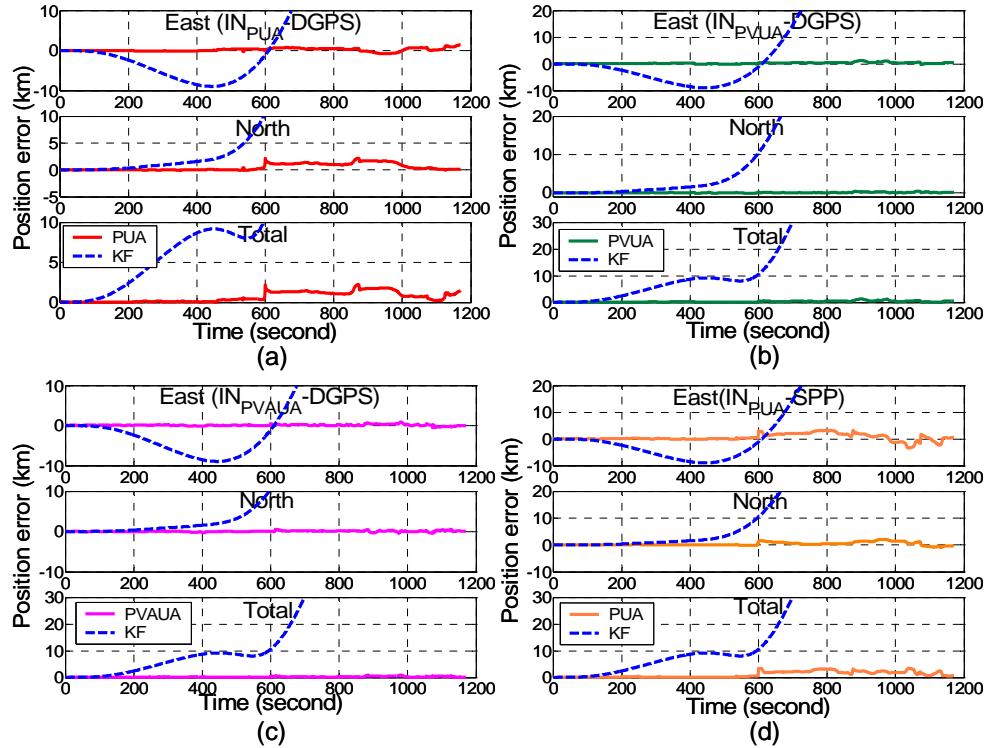


Figure 7.14: Position errors (Xbow/GPS)

Figure (7.14a), Figure (7.14b) and Figure (7.14c) illustrate the position errors generated by the conceptual intelligent navigators using a PUA, PVUA and PVAUA as the

INS/DGPS integration architectures, respectively. As indicated in Table 7.14, the conceptual intelligent navigators were capable of providing 900, 400 and 250 metres total position error in real time mode for 20 minutes, respectively. Similar to the previous section, the conceptual intelligent navigator using a PVAUA was able to provide the most accurate position accuracy compared to the conceptual intelligent navigator using a PVUA or a PUA since the constrained DGPS position derived azimuth measurements and VUA predicted velocity measurements were more accurate than the INS generated azimuth and velocity measurements in stand-alone mode.

Thus, a PVAUA is recommended as the INS/DGPS integration architecture to achieve the 200~300 metres level positioning accuracy requirement using a low cost MEMS IMU for real time prediction that lasts more than 15 minutes. In contrast, a PUA is recommended for system simplicity and if degraded positioning accuracy is acceptable (1 kilometre). In general, the conceptual intelligent navigators discussed so far were able to provide superior long term positioning accuracy (i.e., GPS outage period>1 minutes) based on the field test data applied in this research.

Figure (7.14d) depicts the position errors generated by the conceptual intelligent navigator that used a PUA as the INS/SPP integration architecture. As illustrated in Table 7.14, the conceptual intelligent navigator was capable of providing 1.5 kilometres total position error for real time prediction that lasted 20 minutes. Similar to the performance observed in the previous section, the position error generated by the conceptual intelligent navigator that used SPP was 1.5 times larger than that of the conceptual intelligent navigator that used the same INS/GPS integration architecture with DGPS due to the limitation of the position accuracy of SPP (i.e., 10 metres). However, it was capable of providing superior long term positioning accuracy in comparison with the traditional navigator.

Figure (7.15) illustrates the trajectories generated by different navigators. The trajectory generated by the traditional navigator drifted away from the reference trajectory soon after navigation start (i.e., 1 minute). Thus, only small portion of the KF generated

trajectory was displayed in Figure (7.15) to provide a better visualized effect. All the trajectories generated by the conceptual intelligent navigator using a MEMS IMU were not as good as the trajectories obtained in the previous sections. In fact, visible position errors were observed along each of the trajectories generated by the conceptual intelligent navigators. However, the conceptual navigators were able to maintain the course of the vehicle's maneuvering well in comparison with the traditional navigator, with their trajectories resembling those of the reference trajectory. In fact, the trajectories generated by the conceptual intelligent navigators were accurate during the first 8~10 minutes after the beginning of navigation. During the first 10 minutes, the test vehicle moved westward along a straight line segment then turned south smoothly and moved southward along another straight line segment., Thus, the conceptual intelligent navigators that used a MEMS IMU/GPS integrated system were able to provide acceptable positioning accuracy during the first 8~10 minutes.

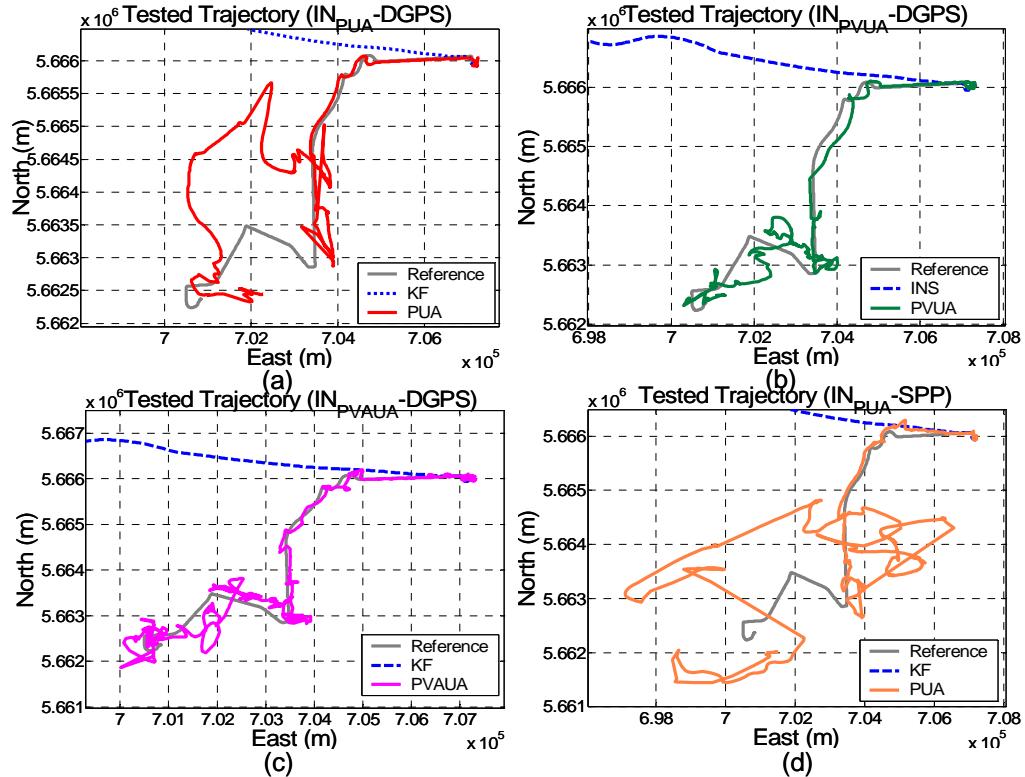


Figure 7.15: Trajectories generated by different navigators (XBOW/GPS)

Using a low cost MEMS IMU/GPS integrated system, the conceptual intelligent navigators was able to provide real time prediction with superior long term positioning accuracy during the first 8~10 minutes using the 3rd field test as the test trajectory. Considering the quality of the IMU, the improvements provided by the conceptual intelligent navigator were significant. In addition, the position errors generated by the conceptual intelligent navigators were mainly affected by the motion dynamic variation in comparison with the time growing errors generated by the traditional navigator during GPS outages.

7.4 Performance Analysis of a Low Cost MEMS/GPS Integrated System

To gain more appreciation of the performance of the conceptual intelligent navigator and traditional navigator, a low cost MEMS IMU/GPS integrated system developed by the MMSS research group will be used in this section. Three additional field tests were conducted on March, 2004 using a navigation grade IMU (CIMU, Honeywell), a MEMS Sensor Triad (MST) IMU (MST sensor triad, MMSS research group) and two NovATel OEM 4 receivers. The MST sensor triad [El-Sheimy, 2004b] consists of three ADI single chip accelerometers (ADXL105, Analog Device) and three ADI single chip gyros (ADXRS150, Analog Device). Figure (7.16a) shows the spatial relationship (lever-arm) between these IMUs and the GPS receiver. The specifications of the ADI sensor triad are given in Appendix C. Similar to section 7.1, the reference trajectories were generated by the INS/DGPS integrated navigation solutions using a navigation grade IMU (CIMU) and a 15 state extended Kalman filter. The data rate of the GPS navigation solutions was 1 Hz.

As illustrated in Figure (7.16b), the first field test was composed of a large loop. The duration of this field test was 1200 seconds and the baseline was 5 kilometres. The second field test consisted of a straight line segment and a large loop, as illustrated in Figure (7.16c). The duration of this field test was 1000 seconds and the baseline was 5 kilometres. As indicated in Figure (7.16d), the third field test consisted of six straight line

segments and eight sharp turns. The duration of this field test was 1800 seconds and the baseline was 15 kilometres. For these tests, the DGPS navigation solutions were not able to provide continuous position and velocity update due to the influence of several GPS signal outages. In the third test, the van drove through residential areas, short tunnels, overpasses and forested areas; thus, eight natural GPS outages were recorded along this trajectory.

The length of these natural GPS outages and the motion dynamic variations experienced by the vehicle are described in Table 7.15. This field test was selected as the test trajectory to evaluate the performance of the conceptual intelligent navigators and traditional navigators using the MST/GPS integrated system.

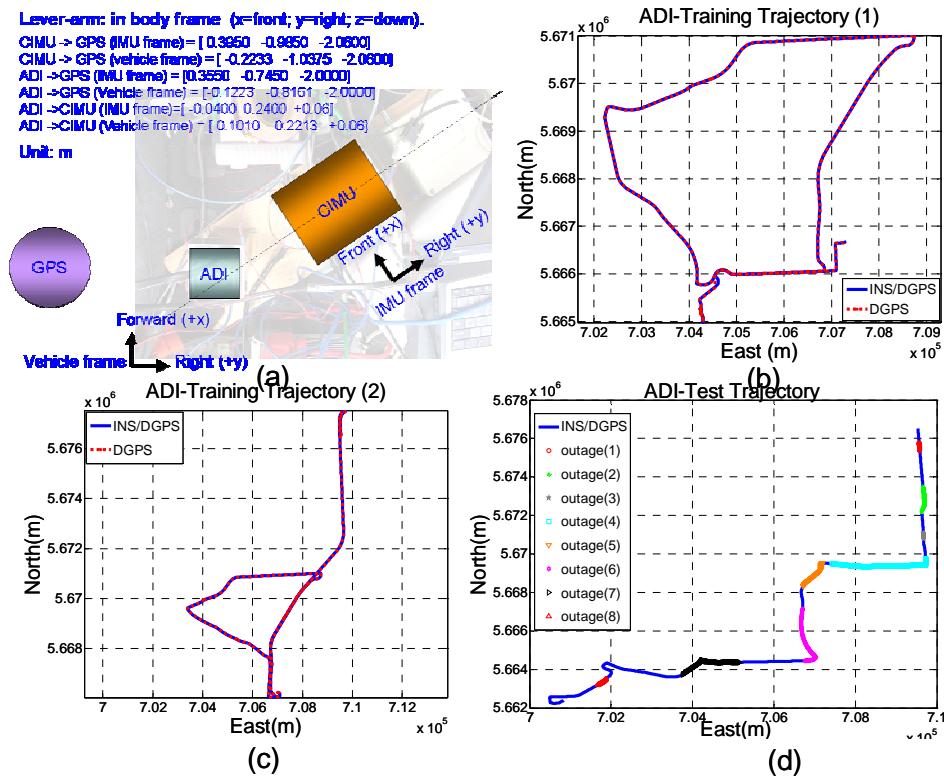


Figure 7.16: System setup and field test trajectories

Table 7.15: Background information of GPS outages

	1	2	3	4	5	6	7	8
Length (second)	30	60	30	120	60	120	120	60
Motion	<i>Straight line (south)</i>	<i>Straight line (south)</i>	<i>Straight line (south)</i>	<i>Sharp turn and Straight line (west)</i>	<i>Smooth turn</i>	<i>Straight line (south) and sharp turn</i>	<i>Straight line and sharp turn (west)</i>	<i>Straight line (south-west)</i>
Distance (m)	612	1260	585	1800	720	1750	1835	1220

7.4.1 Frequent GPS Outages Test

To evaluate the performance of the conceptual intelligent navigator in a more realistic environment, the 3rd field test that contained 8 GPS outage periods was applied. A PUA was implemented as the INS/SPP integration architecture for the conceptual intelligent navigator. Similarly, the traditional navigator was implemented in INS/SPP integration mode. Thus, the prediction results acquired by both navigators during GPS outages were then compared with the reference trajectory for further analysis, respectively.

The stored navigation knowledge was acquired using the IMU and GPS measurements obtained from the XBOW/GPS integrated system (3 field tests, see section 7.1) and MST/GPS integrated system (2 field tests, see Figure (7.16b) and Figure (7.16c)). In addition, the window based weights updating strategy was applied to update the navigation knowledge during the availability of the GPS signal. The window size was set to 60 seconds; however, the conceptual intelligent navigator could be switched to prediction mode using the latest updated navigation knowledge acquired using previous GPS window information whenever a GPS outage took place. In addition, a two steps training procedure was implemented to update the navigation knowledge acquired during each GPS window. See section 6.3.2 for more details about the implementation of the window based weights updating strategy. During the test trajectory, the stored navigation knowledge obtained through the previous 5 field tests was updated during the availability of the GPS signal. Thus, the conceptual intelligent navigator could be adapted to the latest sensor error characteristic and vehicle's motion dynamic variation. The latest

updated navigation knowledge was applied to provide real time prediction whenever a GPS outage took place.

Table 7.16 and Figure (7.17) illustrate the performance of the conceptual intelligent navigator and traditional navigator. In general, the traditional navigator (KF) was able to provide superior performance to the conceptual intelligent navigator when the length of the GPS outage was less than 1 minute. However, its position errors grew with time. Given the same length of GPS outage (i.e., blockage no. 2 and 5), the motion dynamic variations (i.e., azimuth change priori to blockage no.5) helped to reduce the position error in comparison with stable motion dynamic (i.e., straight line priori to blockage no.2).

Table 7.16: Performance summary (MST/GPS)

Blockage no.	Blockage Length(s)	KF			IN_PUA_SPP		
		MAX_N (m)	MAX_E (m)	RMSE Total(m)	MAX_N (m)	MAX_E (m)	RMSE Total(m)
1	30	6.34	25.48	13.76	58.89	10.19	39.96
2*	60	246.85	25.83	110.54	139.88	19.59	84.01
3	30	53.48	14.26	22.23	126.31	19.60	98.55
4*	120	511.57	1016.6	490.80	68.23	80.95	61.13
5	60	85.66	102.17	57.08	148.73	27.06	118.33
6*	120	1406.6	597.60	559.43	75.62	47.31	44.45
7*	120	1491.3	1698.1	878.02	19.80	46.47	19.95
8*	60	39.15	247.66	106.08	31.97	13.78	15.85

In contrast, the conceptual intelligent navigator was able to provide superior performance to the traditional navigator when the length of the GPS outage was longer than 1 minute. Its position errors did not significantly grow with time. Given the same length of GPS outage (i.e., blockage no. 2 and 5), the motion dynamic variation (i.e., azimuth change, blockage no.5) degraded the performance of the conceptual intelligent navigator in comparison with a relatively stable motion dynamic (i.e., straight line, blockage no.2). In other words, the conceptual intelligent navigator was sensitive to the motion dynamic variation of the vehicle. As indicated in section 7.3.3, the impact of dynamic variation could be reduced by using a PVAUA as the INS/DGPS integration architecture when a low cost MEMS IMU/GPS integrated system was applied. However, it requires DGPS

measurements and more sophisticated INS/GPS integration architecture. Thus, only PUA was implemented to integrate the measurements generated by a low cost IMU and GPS in SPP mode due to its simplicity. The trajectories generated by different navigators are shown in Figure (7.18).

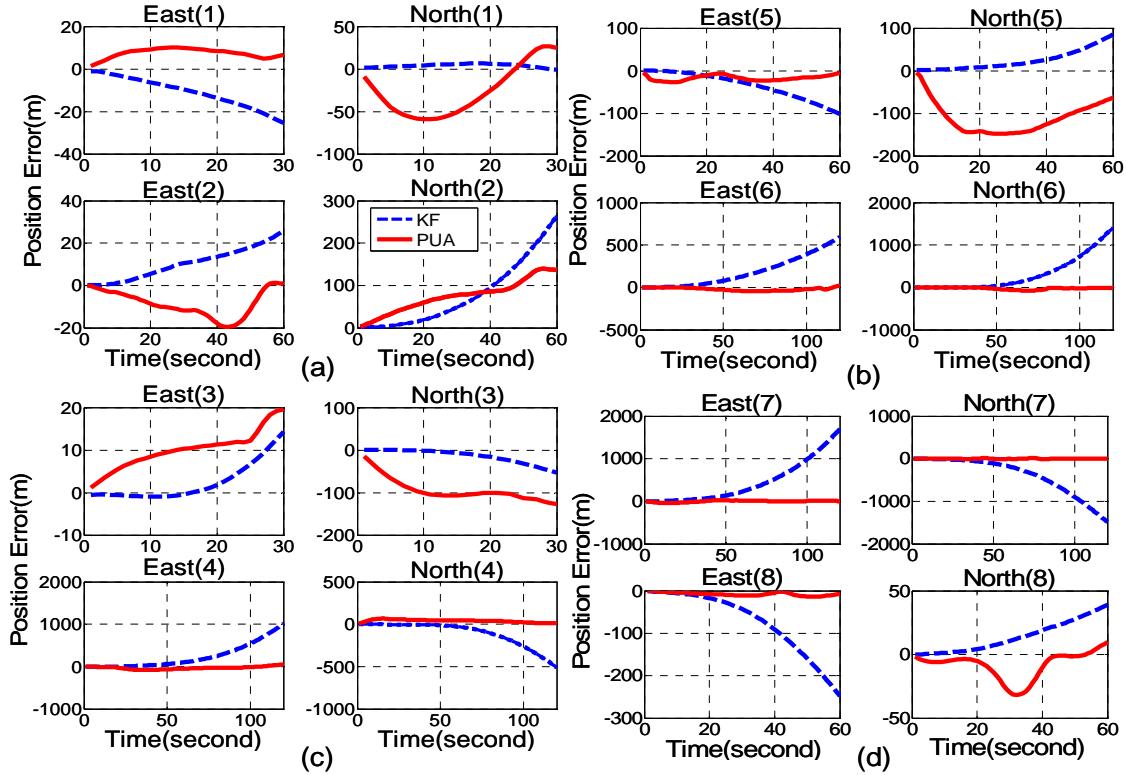


Figure 7.17: Position errors (MST/GPS)

The traditional way of evaluating the performance of different INS/GPS integration methods is usually through examining the positional errors. However, the use of positional errors as the only performance analysis index may not provide any information about the variation of position errors with respect to motion dynamic variations, or time. In fact, it is very crucial to analyze the impact of these variables on the accuracy of MEMS-based navigation systems. This is mainly due to the poor stability of the sensor biases, poor signal to noise ratio and the lack of knowledge about the error characteristics of MEMS IMUs. Therefore, the motivation behind developing the performance analysis index is to provide a way to look into the sources of position errors through post-mission

analysis. As a result, the characteristics of different MEMS IMU/GPS data fusion methods can be further studied.

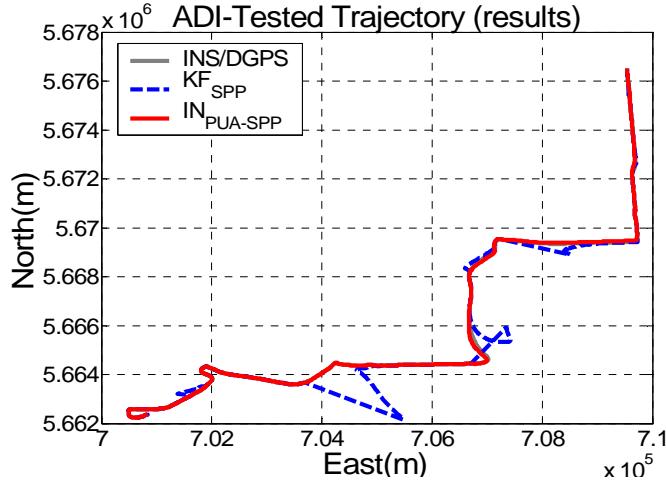


Figure 7.18: Trajectories generated by different navigator (MST/GPS)

Although being simple in theory and implementation, the proposed performance analysis index has potential practical considerations and can be applied in real-time. It treats position errors along different directions and all the candidate variables (velocity, attitude, time, temperature ... etc.) provided by the reference system as different time series and then calculates the correlation coefficients between the position error time series with each of those candidate variables. Thus, the impact of different variables on position error can be quantified. The whole procedure starts by normalizing all candidate time series before calculating the correlation coefficient, $\rho_{Q(i),P(j)}$ as follows:

$$\rho_{i,j} = \frac{E(Q(i) P(j)) - E(Q(i)) E(P(j))}{\sigma_{Q(i)} \sigma_{P(j)}}$$

where $Q(i)$ is a time series of one of the variables affecting the position error and the variable i takes values from 1 to n , where n is the number of variables affecting the position errors. $P(j)$ is the position error along the North position component if $j=1$ and along the East position component if $j=2$. The absolute values of the correlation

coefficient, $|\rho_{i,j}|$, can be regarded as the impact of one of the candidate variables on either the North or East position error component. However, in order to standardize the impact of the different candidate variables affecting the position errors, the correlation coefficient, $|\rho_{i,j}|$, has to be transformed to a common scale by normalizing $|\rho_{i,j}|$ with $\sum_{i=1}^n |\rho_{i,j}|$, thus obtaining the performance analysis index, $R_{i,j}$, as follows:

$$R_{i,j} = \frac{|\rho_{i,j}|}{\sum_{i=1}^n |\rho_{i,j}|}$$

Therefore, $R_{i,j}$ is the transformed version of $|\rho_{i,j}|$ using a common scale. In addition, the magnitude of $R_{i,j}$ can reflect the quantized impact of certain variables on position errors. In other words, it can be applied to interpret the most dominant error sources producing the position errors. Table 7.17 illustrates the performance analysis indices of the position errors generated by the traditional navigator and conceptual intelligent navigator, respectively. In the following tables and figures, the N/EPE represent the normalized position errors time series along North and East directions generated by the traditional navigator and conceptual intelligent navigator, respectively, as shown in Figure (7.19) and Figure (7.20). In addition VN, VE, AZ, R, P and T represent the normalized time series of the velocity in the North direction, velocity in East direction, azimuth, roll, pitch and time provided by the reference system, respectively, as shown in Figures (7.19) and (7.20).

As shown in Figures (7.19) and (7.20), the pie charts were given to display the performance analysis indices illustrated in Table 7.17 to provide better appreciation to the quantized impact of candidate variables on position errors. It can be seen from Figures (7.19) and (7.20) that the position errors generated by the traditional navigator (KF) were mainly related to the time impact (the square of time). The averaged performance

analysis index of time impact was 45%. In fact, it was the only significant factor that affected the position errors generated by the traditional navigator in comparison with the other factors.

In contrast, the performance analysis index indicated that the position errors generated by the conceptual intelligent navigator were mainly affected by the motion dynamic variations. Recalling Table 7.15, the vehicle drove southward during the 1st GPS outage, thus the velocity variation in the North direction contributed 47% to the position errors along the North direction. Similar impact could be found during the 5th GPS outage. As the vehicle drove south-west during this outage, the velocity variation in the North direction contributed 47% to the position errors along the North direction.

During the 2nd GPS outage, the vehicle first drove southward, then experienced a sharp turn toward East, then turned southward again. Thus, the velocity variation in the North direction contributed 30% to the position errors along the North direction. In addition, the azimuth variation contributed 19% and 30 % to the position errors along the North and East directions. It reflected the true motion dynamic variations of the test vehicle. Another interesting example is the performance analysis indices acquired during the 6th GPS outage. The vehicle experienced a sharp turn during this GPS outage, thus the azimuth variation contributed 56% and 46% to the position errors along the North and East directions, respectively.

In addition, the performance analysis indices of roll and pitch obtained by the traditional navigator and conceptual intelligent navigator indicated that the variations in roll and pitch had minor effect on the position errors along the North and East directions, respectively. In general, the variations in roll and pitch contributed 8% and 10% in average to the position errors along the North and East directions generated by the traditional navigator, respectively. Similarly, the variations in roll and pitch contributed 12% and 8% in average to the position errors along the North and East direction generated by the conceptual intelligent navigator, respectively. Therefore, the

performance analysis index can be applied to provide the quantized impact of the vehicle's motion dynamic variation on the positional errors.

Table 7.17: Performance analysis index

Blockage No.	KF	VN (i=1)	VE (i=2)	AZ (i=3)	R (i=4)	P (i=5)	T*T (i=6)
1	NPE(j=1) $\rho_{i,1} , R_{i,1}$	0.43	0.25	0.39	0.04	0.11	0.78
		0.22	0.13	0.20	0.02	0.05	0.38
	EPE(j=2) $\rho_{i,2} , R_{i,2}$	0.32	0.20	0.24	0.20	0.21	0.99
		0.15	0.09	0.11	0.09	0.10	0.46
2	NPE(j=1) $\rho_{i,1} , R_{i,1}$	0.39	0.38	0.44	0.14	0.11	1.0
		0.16	0.15	0.18	0.06	0.04	0.41
	EPE(j=2) $\rho_{i,2} , R_{i,2}$	0.51	0.36	0.43	0.1	0.12	0.98
		0.20	0.14	0.17	0.04	0.05	0.40
5	NPE(j=1) $\rho_{i,1} , R_{i,1}$	0.36	0.26	0.21	0.14	0.03	0.98
		0.18	0.13	0.10	0.07	0.02	0.50
	EPE(j=2) $\rho_{i,2} , R_{i,2}$	0.52	0.32	0.22	0.1	0.02	1
		0.23	0.15	0.1	0.05	0.01	0.46
6	NPE(j=1) $\rho_{i,1} , R_{i,1}$	0.48	0.31	0.03	0.17	0.1	0.97
		0.23	0.15	0.01	0.08	0.05	0.48
	EPE(j=2) $\rho_{i,2} , R_{i,2}$	0.57	0.23	0.20	0.13	0.06	1
		0.26	0.10	0.09	0.06	0.03	0.46
Blockage No.	IN_PUA_SPP	VN (i=1)	VE (i=2)	AZ (i=3)	R (i=4)	P (i=5)	T*T (i=6)
1	NPE(j=1) $\rho_{i,1} , R_{i,1}$	0.71	0.11	0.07	0.20	0.21	0.21
		0.47	0.07	0.05	0.13	0.14	0.14
	EPE(j=2) $\rho_{i,2} , R_{i,2}$	0.64	0.32	0.17	0.06	0.08	0.04
		0.49	0.24	0.13	0.05	0.06	0.03
2	NPE(j=1) $\rho_{i,1} , R_{i,1}$	0.79	0.70	0.50	0.18	0.12	0.32
		0.30	0.27	0.19	0.07	0.05	0.12
	EPE(j=2) $\rho_{i,2} , R_{i,2}$	0.08	0.21	0.19	0.11	0.01	0.02
		0.13	0.34	0.30	0.18	0.02	0.03
5	NPE(j=1) $\rho_{i,1} , R_{i,1}$	0.43	0.02	0.26	0.10	0.03	0.08
		0.47	0.02	0.28	0.11	0.03	0.09
	EPE(j=2) $\rho_{i,2} , R_{i,2}$	0.31	0.09	0.24	0.03	0.02	0.1
		0.39	0.11	0.30	0.04	0.03	0.13
6	NPE(j=1) $\rho_{i,1} , R_{i,1}$	0.05	0.40	0.78	0.05	0.04	0.05
		0.04	0.29	0.56	0.04	0.03	0.04
	EPE(j=2) $\rho_{i,2} , R_{i,2}$	0.14	0.65	0.87	0.05	0.12	0.06
		0.07	0.35	0.46	0.03	0.06	0.03

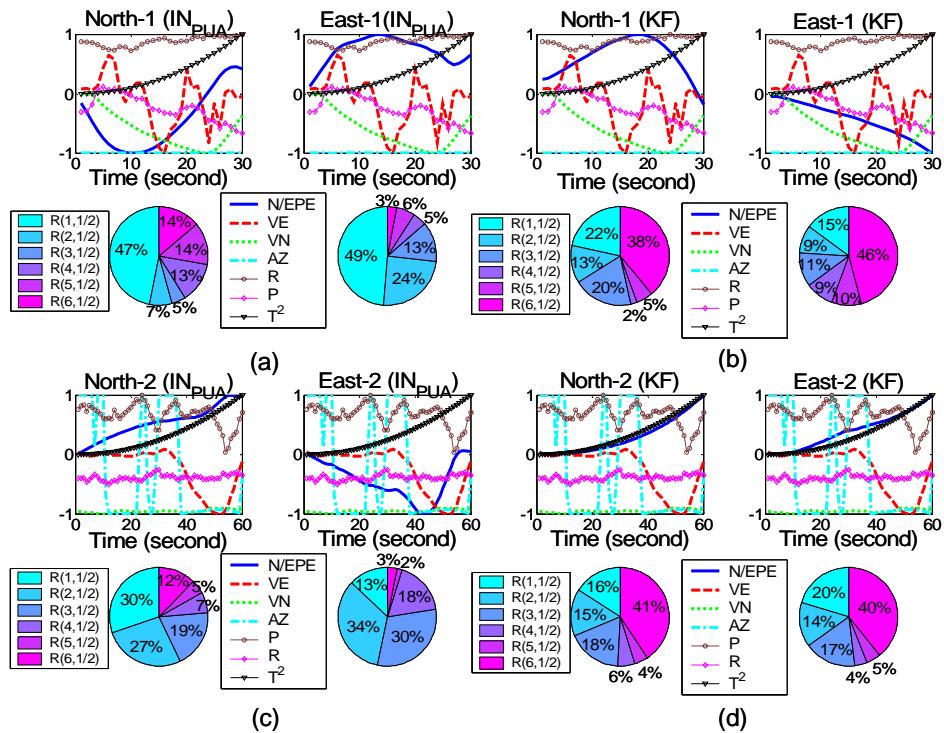


Figure 7.19: Performance analysis index (Outage No.1 and No.2)

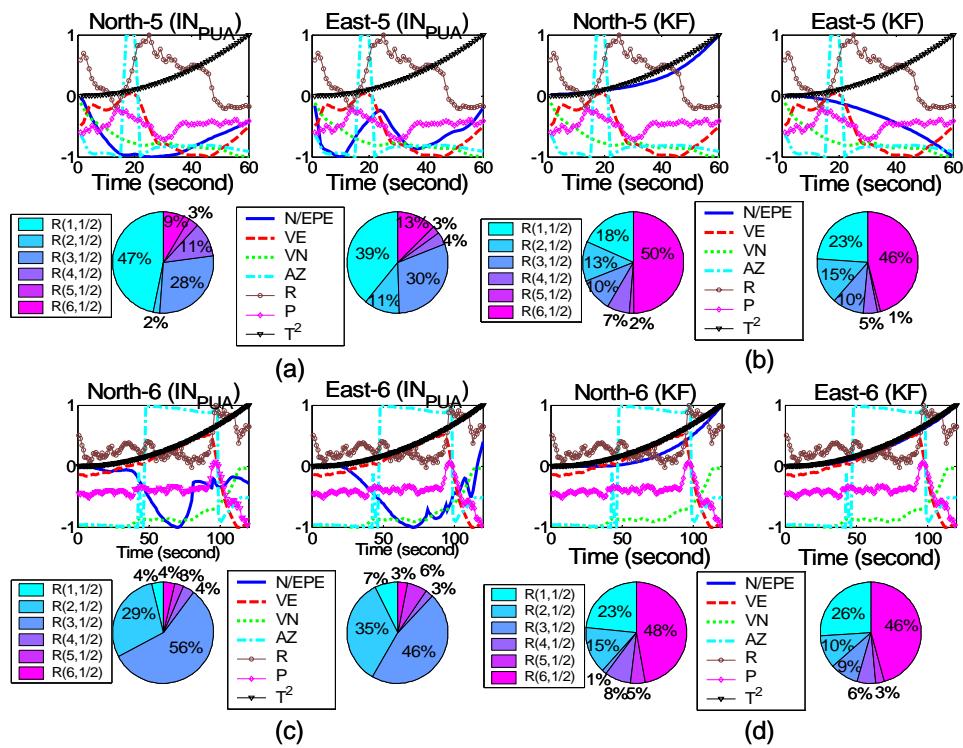


Figure 7.20: Performance analysis index (Outage No.5 and No.6)

7.5 Summary

The results presented in this chapter strongly indicate the potential of including the conceptual intelligent navigator as the core navigation algorithm for the next generation land vehicular navigation system that uses a low cost MEMS IMU integrated with a GPS receiver operating in SPP mode. The most important factor that affects the performance of the conceptual intelligent navigator is the accumulation of navigation knowledge. Theoretically, if enough navigation knowledge can be acquired in one or fewer field tests, the conceptual intelligence might be able to operate in full prediction mode for every new navigation mission. However, the knowledge accumulation should be conducted whenever new navigation knowledge is acquired as the true motion dynamics of the vehicle operating in real life is far more complicated. Thus, the challenge is to acquire large amounts of navigation knowledge and develop an efficient database management facility to accelerate the knowledge retrieval and accumulation process.

With the presence of the conceptual intelligent navigator, the traditional navigator that uses a Kalman filter should be regarded as an optimal estimator, instead of a navigator, as it doesn't have any ability to store and generalize the navigation knowledge that it has learned. In contrast, the conceptual intelligent navigator has the ability to generate, store and generalize the navigation knowledge it has learned. As indicated in previous sections, both navigators have their advantages and limitations when they are applied as the core navigation algorithm for low cost MEMS IMU/GPS integrated land vehicular navigation system. According to the results presented in this chapter, the advantages and the limitations of both navigators are concluded as follows;

- The advantages of the traditional navigator (KF)
 - It is a mature and widely applied navigation algorithm for an INS/GPS integrated system.

- It is capable of providing superior positioning accuracy during short GPS outages (< 1 minute) using a low cost MEMS IMU/GPS integrated system based on the field test data applied in this research.
 - It can estimate sensor bias explicitly. In addition, it is able to provide more states than the conceptual intelligent navigator (i.e., roll and pitch).
- The limitations of traditional navigator (KF)
- Its position error grows with time. Thus, it can not provide acceptable navigation solutions during long GPS outages (>2 minutes).
 - It requires a human expert to tune the optimal parameters of the Kalman filter (i.e., Q and R matrices). In addition, these parameters are sensor dependent.
 - The parameters are not adaptive once the Kalman filter starts navigation. In other words, unless its parameters are optimal, the Kalman filter might not be able to provide reasonable positioning accuracy during certain GPS outages. This issue might not be critical for a navigation grade IMU due to its superior quality; however, it is an important factor that affects the performance of low cost MEMS IMUs due to their poor stability.
 - A Kalman filter generates navigation solutions by receiving information from INS and GPS as it is an optimal estimator in nature. However, it can not recall the experience and the navigation knowledge which are important to a real “navigator”.
- The advantages of the conceptual intelligent navigator

- It can be adapted to the latest sensor error characteristics and vehicle motion dynamic variations by self-learning.
- It is capable of providing superior positioning accuracy during long GPS outages (>1 minute) using a low cost MEMS IMU/GPS integrated system based on the field test data applied in this research.

- It is a sensor independent INS/GPS integration architecture if the accuracy of the system applied for providing the navigation solutions has a similar accuracy level as the system applied to acquire the stored navigation knowledge.
- The importance of synaptic weights which are stored as part of the navigation knowledge to the conceptual intelligent navigator is equivalent to that of the Q and R matrices to the traditional navigator. However, the optimal synaptic weights are acquired adaptively through a continuous self-learning process and the optimal Q and R matrices are given by a human expert through extensive trial and error. In other words, both navigators require a learning process. The learning process of the conceptual intelligent navigator is done by self-learning while the learning process of the traditional navigator is done by a human expert.
- The navigation knowledge can be learned and accumulated continuously. In other words, the up-to-date navigation knowledge is always available for the conceptual intelligent navigator as long as the learning process continues.
- The stored navigation knowledge is an important factor for a conceptual intelligent navigator to provide superior positioning accuracy during long GPS outages even with a low cost IMU/GPS integrated system.

■ The limitations of the conceptual intelligent navigator

- As the training target information is provided by GPS, the navigation states that can be provided by the conceptual intelligent navigator are limited. For example, it can not estimate sensor errors explicitly and it can not provide roll and pitch information as a single GPS receiver can not provide such information for training.
- Currently, the optimal ANN architecture (i.e., number of hidden neurons) of the conceptual intelligent navigator is decided empirically. This is the fundamental problem of a MFNN. In other words, the number of hidden neurons is fixed during data processing. For application like INS/GPS

integration, it might require a more flexible approach to adjust the number of the hidden neurons as the navigation knowledge accumulates continuously.

- It requires more storage space to store the navigation knowledge. As the accumulation of the navigation knowledge makes the conceptual intelligent navigator different from the traditional navigator, it is the price to pay.

Given the fact that the incorporation of artificial intelligence to the navigation algorithm is new to the navigation community, it needs more extensive research to accelerate wider inclusion of such an idea to commercial products. In fact, using artificial intelligence for mobile robot navigation has been studied extensively in robotic engineering related research works since the field of artificial intelligence started. Therefore, developing a new artificial intelligent INS/GPS integration architecture that can overcome some of the limitations of the traditional navigator in a land vehicle environment is a huge challenge.

Unlike other applications in signal processing and control, the INS/GPS integration application requires that synaptic weights stored as part of the navigation knowledge to be adaptive, so that the conceptual intelligent navigator can be adapted to the latest sensor error characteristic and vehicle motion dynamic variations. In other words, the stored navigation knowledge is expected to be adjustable during navigation to ensure the learning process can be conducted continuously.

Although the results presented in this research work are still far from producing a commercial product, the overall achievement presented in this chapter demonstrates the advantages of implementing the conceptual intelligent navigator as the alternate navigation algorithm for the next generation land vehicular navigation system in comparison with the traditional navigator from a software aspect. It is worth mentioning that some of the conclusions draw from this chapter might not be able to be generalized yet due to the limited number of the field test data applied in this research.

CHAPTER 8

CONCLUSIONS AND RECOMMENDATIONS

The first part of this research presented a novel cascade denoising algorithm to improve the positioning accuracy using a traditional INS/GPS integration methodology, Kalman filtering, during GPS outages. The results demonstrated that the proposed algorithm was able to overcome the limitations of existing denoising algorithm in frequency domain and improve the positioning accuracy of the INS system during GPS outages.

The second part of this research presented an alternative INS/GPS integration methodology, the conceptual intelligent navigator, to overcome the limitations of traditional navigator that use a Kalman filter as INS/GPS integration architecture. The conceptual intelligent navigator was developed using artificial neural networks. The major distinction between the traditional navigator and the conceptual intelligent navigator is that the latter has the ability to mimic a human navigator as it can generate, accumulate and generalize the navigation knowledge it has learned over time (i.e. from the different data sets it process). By retrieving the navigation knowledge stored in the navigation information database, the developed navigator is capable of providing real time prediction. The overall performance presented in this research demonstrated its advantage and potential for being applied as an alternative INS/GPS integration architecture for the development of the next generation land vehicular navigation software.

The conclusions drawn from the results presented in this research work are given first to conclude the overall achievement of this thesis. It is worth mentioned that some of the conclusion might not yet to be generalized due to the limited number of field test data set. In addition, some topics that could not be investigated in this thesis are given as recommendations for future work.

8.1 Conclusions

- (1) The universal thresholding algorithm developed in this research is able to preserve the advantage of soft and hard thresholding algorithm. It is capable of providing superior denoising performance in terms of signal-to-noise-ratio and RMS error.
- (2) The second generation denoising algorithm using a translation invariant wavelet transform can overcome the pseudo Gibb's phenomenon generated by the first generation denoising algorithm to provide superior denoising performance in terms of signal-to-noise-ratio and RMS error.
- (3) The spectrum of true motion dynamics is band limited in land vehicle environment. Due to the road irregularities (i.e., bumps); the motion frequencies for X-gyro, Y-gyro and Z-accelerometer mainly appear in the 0-6 Hz band.
- (4) The frequency band of acceleration and deceleration motion in X or Y direction mainly appear in the 0-1Hz band. This indicates a much smoother translation motion along the trajectory. Similarly, the frequency band of the azimuth or heading change mainly appears in the 0-1Hz band. This indicates a much smoother rotation about the vertical body axis. The spectrum characteristics of X-accelerometer, Y-accelerometer and Z-gyro meet the nature of a land vehicle motion.
- (5) The cascade denoising algorithm developed in this research can overcome the limitations of existing denoising algorithm in frequency domain; in addition, it is capable of providing superior performance in position domain.
- (6) The benefit of denoising operation decreases with the quality of IMU. The results demonstrated that the cascade denoising algorithm provided the most significant percentage of improvement using a CIMU/GPS integrated system in DGPS and SPP mode. The percentage of improvement reached 58% and 48%, respectively. For a LN200/GPS integrated system; the percentage of improvement approached 40% in DGPS and SPP mode. For a XBOW/GPS integrated system; it provided the least improvement among these tested systems; however, the percentage of improvement reached 30% in DGPS or SPP mode.
- (7) For a navigation grade IMU (i.e., CIMU) , the long term errors of IMU (i.e., bias, drifts) are stable and well behaved relatively, thus the remaining short term errors

of IMU account for most of the residual position error during GPS outage. However, for a MEMS IMU, due to its poor stability of long term errors, the impact of short term errors is less significant than long term errors. Thus, the long term errors dominate the residual position errors for current MEMS IMUs, however, the use of cascade denoising algorithm is still able to improve the positioning accuracy during GPS outages by reducing the impact of short term errors.

- (8) The results demonstrated that the INS/GPS integration architecture using an Elman network demanded more computation resources, more training time in comparison with the INS/GPS integration architecture using a MFNN. In addition, the former did not demonstrate any improvement in term of the positioning accuracy during GPS outages, thus, the MFNNs were applied to develop the INS/GPS integration architecture required by the conceptual intelligent navigator.
- (9) The azimuth constrained algorithm developed in this research provides a “pseudo” azimuth sensor that can overcome the numerical instability of raw DGPS velocity or position derived azimuth. In addition, such measurements allow the implementation of PVAUA. Based on the field test data applied in this research ,the results demonstrated that the constrained DGPS velocity or position derived azimuth was able to provide more accurate azimuth measurements in comparison with the azimuth measurements generated by a tactical grade IMU and MEMS IMU operating in stand-alone mode for 30 minutes.
- (10) The length of GPS outage is the most dominated factor to the position errors generated by the traditional navigator. In contrast, the motion dynamic variation is the most dominated factor to the position errors generated by the conceptual intelligent navigator. However, the impact of motion dynamic variation can be reduced incorporating with more sophisticated INS/GPS integration architectures, such as a PVUA or a PVAUA.
- (11) The optimal parameters of the traditional navigator are provided by a human expert through an extensive tuning process before the actual use of the system. In contrast, the parameters of the conceptual intelligent navigator are included in the

- navigation knowledge. Thus, they can be updated without a human expert during navigation whenever newly updated navigation knowledge is acquired.
- (12) The most important factor that affects the performance of the conceptual intelligent navigator is the accumulation of navigation knowledge. Given the fact that the true motion dynamic of vehicle operating in real life is far more complicated thus the knowledge accumulation should be conducted whenever new navigation knowledge is acquired.
 - (13) Based on the field test data applied in this research, the PVUA architecture is recommended as the INS/DGPS integration architecture for systems using a navigation grade IMU. It can achieve 5 metres level positioning accuracy requirement for real time prediction that lasts more than 15 minutes without GPS. In contrast, the positioning accuracy provided by the conceptual intelligent navigator using a PUA as INS/SPP integration architecture can reach 30 metres during the same period.
 - (14) Based on the field test data applied in this research, the PVAUA architecture is recommended as the INS/DGPS integration architecture for systems using a tactical grade IMU. It can achieve 15 metres level positioning accuracy requirement for real time prediction that lasts more than 15 minutes without GPS. In contrast, the positioning accuracy provided by the conceptual intelligent navigator using a PUA as INS/SPP integration architecture can reach 90 metres during the same period.
 - (15) Based on the field test data applied in this research , the PVAUA architecture is recommended as the INS/DGPS integration architecture for systems using a low cost MEMS IMU, It can achieve 200~300 metres level positioning accuracy requirement using for real time prediction that lasts more than 15 minutes without GPS. In contrast, the positioning accuracy provided by the conceptual intelligent navigator using a PUA as INS/SPP integration architecture can reach 1.5 kilometres during the same period.
 - (16) The continuous learning process can be regarded as the evolution process of conceptual intelligent navigator. It ensures the conceptual intelligent navigator can be adapted to the latest sensor error characteristic and vehicle's motion

- dynamic variation. Therefore, the latest updated navigation knowledge can be applied to provide real time prediction during GPS outages.
- (17) The performance analysis index that makes use of the correlation coefficients between the position errors generated by the navigator and motion dynamics information provided by the reference system can be applied to interpret the quantized impact of the vehicle's motion dynamic variation on the position errors for post-mission analysis.
 - (18) For developing the next generation land vehicular navigation system, a low cost IMU and a GPS receiver that operates in SPP mode are preferred due to the cost and the simplicity of system integration in hardware and software aspect. The results presented in this thesis strongly indicate the potential of including the conceptual intelligent navigator as the core navigation algorithm for the next generation land vehicular navigation system.

8.2 Recommendations

Based on the work done in this research, the following recommendations for future research are proposed:

- (1) The research works associated with the development of cascade denoising algorithm can be extended to incorporate either unsupervised neural networks (i.e., Self-organizing map, SOM) or supervised neural networks (i.e., static neural networks or dynamic neural networks). For example, the denoising unit using wavelet decomposition can be integrated into the neurons in a MFNN. The cost function used in neural network learning is also applied as the denoising criterion and hence denoising itself is treated as a part of the integrated model.
- (2) The correlation between motion dynamic variation and the degradation of the positioning accuracy during GPS outage after applying cascade denoising algorithm should be further analyzed, thus, the cause of such inconsistent performance can be investigated.

- (3) The alignment procedure applied in this research relies on Kalman filter , therefore, to develop a true ANN based INS/GPS scheme without using the Kalman filter, an ANNs based alignment procedure should be investigated.
- (4) Unlike the traditional navigator, the conceptual intelligent navigator developed in this research can not estimate INS sensor biases or drifts explicitly. This is mainly due to the limitations of desired information provided by the GPS navigation solutions. In other words, it doesn't have the ability to feedback estimated biases and drifts to compensate the raw IMU measurements as those parameters are estimated implicitly within the parameters of conceptual intelligent navigator during navigation. This is the major limitation of supervised neural networks. Therefore, an unsupervised neural network (i.e., SOM) can be implemented as part of the intelligent navigator to estimate INS sensor biases or drifts explicitly and continuously during navigation then feedback to compensate the raw IMU measurements since SOM doesn't require desired information during learning or training procedure.
- (5) The critical parameters of MFNN can only be decided empirically, such as the number of hidden neurons. Thus a modified version of MFNN, cascade correlation networks, is recommended. It begins with a minimal network, then automatically trains and adds new hidden units one by one creating a multi-layer feed-forward structure. In other words, the number of hidden neurons can be decided based on the complexity of the problem automatically during the accumulation of the knowledge.
- (6) The conceptual intelligent navigator developed in this research uses artificial neural networks. In fact, the incorporation of other artificial intelligent algorithms with navigation applications should be investigated with more efforts. Such as Fuzzy logic, expert system and genetic algorithms.
- (7) Like the traditional navigator, the conceptual intelligent navigator developed in this research relays on the traditional INS mechanization as the source of the INS navigation solution. Theoretically, the conceptual intelligent navigator can be modified to mimic the INS mechanization. In other words, it can receive the raw

- measurements provided by an IMU and generate navigation solutions to bridge the gap between GPS outages.
- (8) Most of the GPS outages applied in this research were simulated in an ideal way by simply removing GPS solutions completely; however, the GPS outages happened in urban area usually degrade gradually. Although the conceptual intelligent navigator developed in this research has the ability to detect the degraded performance of GPS, it should be evaluated using more realistic GPS outages as well.
 - (9) As the ANN model is implemented based on empirical and adaptive model, it is not possible to provide proper stochastic model for further analysis, on the contrary, the traditional navigator using a Kalman filter is an optimal estimator in nature. On the other hand, the conceptual intelligent navigator using artificial intelligent algorithms has the ability to accumulate and generalize navigation knowledge as a real “navigator”. Therefore, a hybrid intelligent navigator incorporating a Kalman filter with the conceptual intelligent navigator that has a proper stochastic model can be expected to enhance the advantage in terms of estimation optimality; in addition, it can accumulate navigation knowledge continuously.

REFERENCES

1. Abhijit, S.P. and Robert, B.M. (1996): Pattern Recognition with Neural networks in C++, IEEE PRESS.
2. Aboufadel, E. and Schlicker S.(1999): Discovering Wavelets, Wiley-Interscience.
3. Aguilar, M., Barniv, Y., and Garrett, A. (2003): Prediction of pitch and yaw head movements via recurrent neural networks. Proceedings of IEEE International Joint Conference on Neural Networks 2003.
4. Alban, S., Akos, D., Rock, S., and Gebre-Egziabher, D. (2003): Performance Analysis and Architectures for INS-Aided GPS Tracking Loops Proceeding of the Institute of Navigation's National Technical Meeting 2003, January Anaheim, CA.
5. Anderson, J.A., Silverstein, J.W., Ritz, S.A., and Jones, R.S. (1977): Distinctive features, Categorical Perception, and Probability Learning: Some Applications of a Neural Model. *Psychological Review*, 84, pp. 413–451.
6. Anderson, D. and McNeill, R.(1992): Artificial Neural Network Technology, Technical Reports, ITT Industries.
7. Anthony, M.M. and Bartlett, P.L. (1999): *Neural Network Learning: Theoretical Foundations*, Cambridge University Press.
8. Arbib, M.A. (1987): *Brains ,Machines and Mathematics*, Second Edition, Springer-Verlag, New York.
9. Ashkenazi, A., Moore, T., Dumville, M., Lowe, D., and Tsakiri, M. (1995): An Artificial Intelligent Highway system., In: Proc. of the 8th International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GPS-95, Palm Springs, California USA.(CD)
10. Bar-Shalom, Y., Li, X., and Kirubarajan, T. (2001): *Estimation with Applications to Tracking and Navigation*, John Wiley & Sons, Inc.
11. Barto, A.G., Bradtk, S.J., and Anderson, C.W. (1983): Neuronlike adaptive elements that can solve difficult learning control problem, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol.SMC-13,pp. 834-846.

12. Battiti, R. and Masulli, F. (1990): BFGS Optimization for Faster and Automated Supervised Learning, Proceeding of INCC90 Paris, Int. Neural Networks Conference., pp. 757-760.
13. Battiti, R. (1992): First and Second Order Methods for Learning: Between Steepest Descent and Newton's Method, Neural Comput., Vol.5, pp. 141-166.
14. Bender, E.A.(1996): Mathematical Methods in Artificial Intelligence, IEEE Computer Society Press.
15. Bishop, C.M. (1995): Neural Networks for Pattern Recognition, Oxford University Press.
16. Black, H.D. and Eisner, A. (1984): Correcting satellite Doppler data for Tropospheric range effects, Journal of Geophysical Research,89(D2), pp. 2616-2626.
17. Britting, K. R. (1971): Inertial Navigation Systems Analysis. John Wiley & Sons, Inc.
18. Brown, R. G. and Hwang, P. Y. C. (1992): Introduction to random signals and applied Kalman filtering. John Wiley & Sons, Inc. New York.
19. Bruton, A.M., Skaloud, J., and Schwarz, K.P, (1999): The Use of Wavelets for the Analysis and De-noising of Kinematic Geodetic Measurements. Geodesy Beyond 2000- IAG General Assembly, Birmingham, UK, July 19-24, International Association of Geodesy Symposia, Vol. 121, 2000, Springer, pp. 227-232.
20. Callatay, A.M.D. (1992): Natural and Artificial Intelligence: Misconceptions about Brains and Neural Networks, Elsevier Science & Technology.
21. Cawsey, A. (1998): The Essence of Artificial Intelligence, Prentice Hall PTR.
22. Cerepakhin, Y. Z. and Greenwood, D. (1994): Neural Network Loran-C Calibration Using GPS. Proceeding of the 7th International Technical Meeting of the Satellite Division of the Institute of Navigation, ION94, Salt Lake City, Utah, U.S.A., September 20-23.
23. Chansarkar, M. (1999): GPS Navigation using Neural Networks, Proceeding of ION GPS-99, Nashville, TN.
24. Chatfield, A. B. (1997): Fundamentals of High Accuracy Inertial Navigation. American Institute of Aeronautics and Astronautics, Inc.

25. Chiang, K.W. and El-Sheimy, N. (2002): INS/GPS Integration using Neural Networks for Land Vehicle Navigation Applications, Proceedings of the US Institute of Navigation (ION) GPS'2002 meeting, September 24-27, 2002 - Oregon Convention Center, Portland, Oregon, USA (CD).
26. Chiang, K.W., Noureldin, A., and El-Sheimy, N.(2003): Multi-sensors Integration using Neuron Computing for Land Vehicle Navigation, GPS Solutions., Vol. 6, No. 3, pp. 209-218.
27. Chiang, K.W.(2003): The Utilization of Single Point Positioning and a Multi-Layers Feed-forward Network for INS/GPS Integration, Institute of Navigation (ION) GPS'2003 meeting, September 9-12, 2003 - Oregon Convention Center, Portland, Oregon, USA (**Best Student Paper Award**)(CD).
28. Chiang, K.W.(2004): Development of an Optimal GPS/MEMS Integration Architecture for Land Vehicle Navigation Utilizing Neural Network. Journal of Global Position System and CPGPS student paper competition (**Best Student Paper Award**).
29. Chiang, K.W. and El-Sheimy, N. (2004a): The Performance Analysis of Neural Network Based INS/GPS Integration Method for Land Vehicle Navigation. The 4th International Symposium on Mobile Mapping Technology MMT'2004), March 29-31, 2004 – Kunming, China (CD).
30. Chiang, K.W. and El-Sheimy, N. (2004b): Improving Positioning Accuracy during GPS Outages Utilizing a Neural Networks based GPS/MEMS Integration Architecture for Land Vehicle Navigation, Geoid Annual Scientific Conference, May 30-June 1,Gatineau, Quebec, Canada (Poster).
31. Chiang, K.W. and El-Sheimy, N. (2004c): Artificial Neural Networks in Direct Georeferencing, Photogrammetric Engineering and Remote Sensing, July, 2004 (Invited column paper).
32. Chiang, K.W., El-Sheimy, N., and Noureldin, A. (2004a): A New Weights Updating Method for Neural Networks based INS/GPS Integration Architectures, Measurement Science and Technology, Vol. 15, No.10, pp. 2053-2061.
33. Chiang, K.W., Hu, H., Niu, X., and El-Sheimy, N. (2004b): Improving Positioning Accuracy of a GPS/MEMS Integrated Vehicular Navigation System

- During DGPS Outages Periods Utilizing Cascade Denoising Algorithm, GNSS 2004, ION, September 21-24, 2004 - Long Beach, California, USA.
34. Christodoulou, C. and Georgopoulos, M. (2000): Applications of Neural Networks in Electromagnetics, Artech House Publishers.
 35. Cody, M.A. (1992): The Fast Wavelet Transform, Dr. Dobb's Journal, April 1992.
 36. Coifman, R.R. and Donoho,D.L. (1995): Translation-Invariant De-Noising, Lecture Notes in Statistics, 103,pp.125-150,Yale University and Yale University
 37. Czerniak, R. J. and Reilly J. P. (1998): Application of GPS for Surveying and Other Positioning Needs in Departments of Transportation. Synthesis of Highway Practice, 258, Washington, DC: National Academy Press, 1998.
 38. Czompo, T. (1990): Use of Spectral Methods in Strapdown ISS Data processing, Proceedings of the international Symposium on Kinematic System for Geodesy, Navigation and remote sensing, Banff, September 10-14, Springer-Verlag, New York.
 39. Daubechies, I. (1992): Ten Lectures on Wavelets. Philadelphia:SIAM.
 40. Daubechies, I. (1996): Where do wavelets come from? (A personal point of view), Proceedings of the IEEE, Special Issue on Wavelets, 84 (no. 4), pp. 510--513, April 1996.
 41. Donoho, D.L. (1992): De-Noising via Soft Thresholding. IEEE Trans. Info. Theory, Vol.41,No.3, May 1995, pp. 613-627.
 42. Donoho, D.L. and Johnstone,I.M..(1995): Adapting to Unknown Smoothness via Wavelet Shrinkage, Journal of the American Statistical Association, Vol. 90, No. 432, December, pp. 1200-1224.
 43. Douglas, S.C., and Meng, T.H.Y. (1991): Linearized Least Squares Training of Multilayer Feedforward Neural Networks, Proceeding of IEEE IJCNN, Vol.I, pp. 307-312, Seattle, WA.
 44. Dumville, M. and Tsakiri, M. (1994): An Adaptive Filter for Land Navigation using Neural Computing., In: Proc. of the 7th International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GPS-94, Salt Lake City, USA.

45. Drane, C. and Rizos C. (1998): Positioning Systems in Intelligent Transportation System.Artech House Publishers.
46. Elman, J.L. (1999): Finding Structure in Time, Connection Science,Vol.14, pp. 179-211.
47. El-Sheimy, N. (2000a): An Expert Knowledge GPS/INS System for Mobile Mapping and GIS Applications, Proceeding of the 2000 National Technical Meeting of the Satellite Division of the Institute of Navigation - January 26-28, 2000, California, USA.
48. El-Sheimy, N. (2000b): Integrated Systems and their Impact on the Future of Positioning, Navigation, and Mapping Applications, The FIG Working Week, Prague, May 29-June 1, 2000.
49. El-Sheimy, N., Chiang, K.W., and Noureldin, A. (2003): The Utilization of Artificial Neural Networks for Multi-Sensor System Integration in Navigation and Positioning Instruments, IEEE Transactions on Instrumentation and Measurement. (submitted on April ,2003,revised on October,2004).
50. El-Sheimy, N. (2004a): Inertial Techniques and INS/DGPS Integration, ENGO 623 lecture notes, Department of Geomatics Engineering, The University of Calgary.
51. El-Sheimy, N. (2004b) : The MEMS Sensor Triad – Next generation IMU for Land Vehicle Applications, The GEOIDE Network's sixth Annual Scientific Conference, Gatineau, Quebec, May 30-June 1st, 2004.
52. El-Sheimy, N.,Chiang, K.W., Niu, X., and Noureldin, A.(2004): Performance Analysis of GPS/MEMS based IMU Data Fusion Utilizing Artificial Neural Networks, European Journal of Navigation, Vol.2,No.4,November,2004.
53. Enge, P., Swanson, E., Mullin, R., Ganther, K., Bommerito, A., and Kelly, R. (1995): Terrestrial Radionavigation Technologies, Journal of the Institute of Navigation, No.1,Vol.42, pp.61-108.
54. Farrell, J. A. and Barth, M. (1998): The Global Positioning System & Inertial Navigation. McGraw–Hill.
55. Fine, T.L. (1999): Feedforward Neural Network Methodology. Springer, New York.

56. Forrest, M., Spracklen, T., and Ryan, N. (2000): An inertial Navigation data fusion system employing an artificial neural network as the data integrator. In: Proc. of the ION NTM 2000 Anaheim, CA. January 26-28, 2000 (CD).
57. Frykman, P. (2003): Applied Particle Filters in Integrated Aircraft Navigation System, (M.sc, Thesis), Department of Electrical Engineering, The university of LinkÄoping, Sweden.
58. Fredrik, G., Niclas B., Urban, F., Jonas, J., Rickard, K., and Per-Johan, N. (2002): Particle filters for positioning, navigation and tracking. IEEE Transactions on Signal Processing, Vol. 50, No. 2, 2002.
59. French, R.L. (1986): Historical Overview of Automobile Navigation technology, proceeding of 36th IEEE Vehicular Technology Conference, Dallas, TX, May 20-22,1986, pp. 350-358.
60. French, R.L. (1987): Automobile Navigation: Where is it going? IEEE AES Magazine, May, pp. 6-12.
61. French, R.L. (1990): Land Vehicle Navigation-A world wide perspective. Proceeding of the Land Navigation and Information Systems, 19-20 September,1990, Warwick,U.K., paper ,Vol.2, No. 2.
62. French, R.L. (1995): From Chinese Chariots to Smart Cars:2000 Years of Vehicular Navigation, Journal of the Institute of Navigation, Spring, Vol.1 No.1, pp. 235-258.
63. Gad-El-Hok, M. (2001): The MEMS Handbook, CRC Press.
64. Gelb, A. (1974): Applied Optimal Estimation. The MIT Press.
65. Golden, R.M. (1996): Mathematical Methods for Neural Network Analysis and Design.
66. Grewal, M. S., Weill, L. R., and Andrews, A. P. (2001): Global Positioning Systems, Inertial Navigation, and Integration. John Wiley & Sons, Inc.
67. Greenspan, R.L. (1995): Inertial Navigation Technology from 1970-1995, Journal of the Institute of Navigation, No.1 , Vol.42, pp. 165-183.
68. Guo, H. (1995): Theory and Applications of the Shift-Invariant, Time-Varying and Undecimated Wavelet Transforms, M.S. thesis. ECE Dept. Rice University.

69. Guo, H., Lang, M., Odegard, J. E., and Burrus, C. S. (1995): Nonlinear Shrinkage of Undecimated DWT for Noise Reduction and Data Compression. Proceedings of the International Conference on Digital Signal Processing. Limassol, Cyprus. June 26-28, 1995.
70. Ham, F.M. and Kostanic, I. (2001): Principles of Neurocomputing for Science and Engineering . McGraw-Hill.
71. Haykin, S. (1999): Neural Networks: A Comprehensive Foundation, 2nd Edition, Prentice-Hall, New Jersey.
72. Haykin, S. (2001): Kalman Filtering and Neural Networks, John Wiley & Sons, Inc.
73. Hein, G.W., Godet, J. I., Jean-Luc M., Jean-Christophe E., Philippe L., Rafael, and Pratt,T. (2002): Status of Galileo frequency & signal design, Proceedings of the US Institute of Navigation (ION) GPS'2002 meeting, September 24-27, 2002 - Oregon Convention Center, Portland, Oregon, USA (CD).
74. Hofmann-Wellenhof,, B., Lichtenegger, H., and Collins J.(2001): GPS Theory and Practice, Springer New York.
75. Hofmann-Wellenhof, B., Legat, K., and Wieser, M.M. (2003): Navigation - principles of positioning and guidance. Springer, Wien New York.
76. Honavar, V. and Uhr L. (1994): Artificial Intelligence and Neural Networks : Steps Toward Principled Integration, Boston: Academic Press.
77. Hopfield, J.J. (1982): Neural Networks and Physical System Systems With Emergent Collective Computational Abilities, Proceeding of the National Academy of Sciences,USA, Vol.79,pp. 8429-8433.
78. How, J. and Deyst, J. (2004): Advanced Estimation for GPS and Inertial Navigation, lecture notes (Spring 2004), MIT.
79. Jansen, M. (2001): Noise reduction by wavelet thresholding, New York Springer,
80. Jekeli, C. (2001): Inertial Navigation Systems with Geodetic Applications, Walter De Gruyter, Berlin, New York.
81. Johansson, E.M., Dowla, F.U., and Goodman, D.M. (1990): Backpropagation Learning for Multi-Layer Feed-Forward Neural Networks Using the Conjugate

- Gradient Method, Lawrence Livermore National Laboratory, Preprint UCRL-JC-104850.
82. Jordan, M.I. (1986): Attractor Dynamics and Parallelism in a Connectionist Sequential Machine, The Eighth Annual Conference of the Cognitive Science Society,pp. 531-546, Amherst,MA.
 83. Kalman, R.E. (1960): A New Approach to Linear Filtering and Prediction Problem,ASME Journal of Basic Engineering,Vol.82,pp. 34-45.
 84. Kayton, M. and Fried, W.R. (1996): Avionic Navigation Systems, Second Edition, John Wiley & Sons, Inc.
 85. Kim, C.N. and Mohan, T.(1998): A neuro-fuzzy controller for mobile robot navigation and multirobot convoying, IEEE Transactions on Systems, Man, and Cybernetics--Part B: Cybernetics v28, n6 (Dec, 1998).
 86. Kohonen, T. (1977): Associative memory: A system theoretical approach. New York: Springer.
 87. Kohonen, T. (2001): Self-Organizing Maps Third Edition, Springer, New York.
 88. Krakiwsky, E.J. (1993): Tracking the Worldwide Development of IVHS Navigation Systems. GPS World, Vol.5, No.2, pp. 40-47.
 89. Krakiwsky, E.J. (1994): Innovations in Vehicle Tracking and Navigation. GPS World, Vol.5, No.5 pp. 42-46.
 90. Kreye, C., Eissfeller, B., Sanroma, D.,and Lück, T. (2002): Performance Analysis and Development of Tightly Coupled GNSS/INS System,; NATO Sensor & Electronics Technology Panel Symposium on "Emerging Military Capabilities Enabled by Advances in Navigation Sensors", 14-16 October 2002, Istanbul, Türkei.
 91. Lachapelle, G. (2001): NAVSTAR GPS: Theory and Applications , ENGO625 lecture notes, Department of Geomatics Engineering, The University of Calgary
 92. Leyden, M., Toal, D., and Flanagan, C.(1999): A Fuzzy Logic Based Navigation System for a mobile Robot, Automatisierungs symposium Wismar.
 93. Lin, C.F.(1993): Advanced Control System Design, Prentice Hall.

94. MacDonald, K.D. (2001): The Modernization of GPS: Plans, New Capabilities and the Future Relationship to Galileo, Journal of Global Position System, No.1,Vol.1, pp. 1-17.
95. Mallat, S.(1989a): A Theory of Multiresolution Signal Decomposition: The Wavelet Representation, IEEE Trans.PAMI,11,pp. 674-693.
96. Mallat, S.(1989b): Multiresolution approximation and wavelet orthonormal bases of L₂, Transaction of the American Mathematical Society, vol. 315, pp. 69-87, September.
97. Mallat, S. (2001): A Wavelet Tour of Signal Processing, Second Edition, Academic Press.
98. Mandic, D.P. and Chambers, J.A. (2001): Recurrent Neural Networks for Prediction: Architectures, Learning Algorithms and Stability. John Wiley & Sons
99. Masters, T.(1995): Neural, Novel & Hybrid Algorithms for Time Series Prediction , John Wiley & Sons, Toronto.
100. Meng, M. and Kak, A.C.(1993): Mobile robot navigation using neural networks and nonmetrical environment models, IEEE Control Systems, pp. 30-39, October 1993.
101. Misra, P. and Enge, P. (2001): Global Positioning System : Signals, Measurements, and performance, Ganga-Jamuna Press,Massachusetts,2001.
102. Misiti, M., Misiti, Y., Oppenheim, G., and Poggi, J. (2000): Wavelet Toolbox Manual, The Mathworks.
103. Mohamed, A. H. (1999): Optimizing the Estimation Procedure in INS/GPS Integration for Kinematic Application. UCGE Reports Number 20127, Department of Geomatics Engineering, The University of Calgary.
104. Moller, M. (1993): A Scaled Conjugate Gradient Algorithm for Faster Supervised Learning, Neural Networks,Vol.6, No.4, pp. 525-534.
105. Nasser, S. (2004): Improving the Inertial Navigation System (INS) Error Model for INS and INS/DGPS Applications. PhD thesis UCGE Reports 20183, Department of Geomatics Engineering, The University of Calgary.

106. Nason, G.and Silverman, B. (1995): The stationary wavelet transform and some statistical applications, in Wavelets and Statistics (Antoniadis and Oppenheim eds.), pp. 281-299.
107. Norgarrd, M.,Rave, O.,Poulsen, N.K., and Hansen, L.K.(2001): Neural Networks for Modelling and Control of Dynamic System, Springer.
108. Noureldin, A.(2002): New Measurement- While- Drilling Surveying Technique Utilizing Sets of Fiber-Optic Rotation Sensors, Department of Electrical Engineering and Computer Science, (Ph.D. thesis), 2002, The university of Calgary, March 2002.
109. NSTB/WAAS, (2003): Global Positioning System (GPS) Standard Positioning Service (SPS) Performance Analysis Report, William J. Hughes Technical Center, NSTB/WAAS T&E Team.
110. Ojeda, L. and Borenstein, J. (2002): FLEXnav: Fuzzy Logic Expert Rule-based Position Estimation for Mobile Robots on Rugged Terrain., Proceedings of the 2002 IEEE International Conference on Robotics and Automation, Washington DC, USA, 11 - 15 May 2002, pp. 317-322.
111. Ojeda, L. Reina, G., and Borenstein, J. (2004): Experimental Results from FLEXnav: An Expert Rule-based Dead-reckoning System for Mars Rovers., IEEE Aerospace Conference, Big Sky, MT, March 5-14, 2004
112. O'Keefe, K. (2001): Availability and Reliability Advantages of GPS/Galileo Integration. Proceedings of the US Institute of Navigation (ION) GPS'2001 meeting, Salt Lake City, September 11-14, 2002 -, USA (CD).
113. Olivier, R. and Vetterli, M. (1991): Wavelets and signal processing, IEEE Signal Processing Magazine, October 1991, pp. 14-38.
114. Parkinson, B.W., Spilker, J.J., Axelrad, P., and Enge, P. (1996): GPS: Theory and Application Vol I, II, American Institute of Aeronautics and Astronautics.
115. Petovello, M.G. (2003): Real-time Integration of a Tactical-Grade IMU and GPS for High-Accuracy Positioning and Navigation, PhD thesis, April 2003, UCGE Report No. 20173.

116. Percival, B.D., Andrew, T., Walden, R.G., Ripley, B. D., Ross, S., Stein, M., and Williams, D. (2000): Wavelet Methods for Time Series Analysis (Cambridge Series in Statistical and Probabilistic Mathematics).
117. Principe, J.C., Euliano N.R., and Lefebvre, W.C. (2000): Neural and Adaptive Systems, John Wiley & Sons, Inc.
118. Raquet, J.F. (1998): Development of a Method for Kinematic GPS Carrier-Phase Ambiguity Resolution Using Multiple Reference Receiver, (Ph.D. thesis), May 1998, UCGE Report No. 20116. Department of Geomatics Engineering, The University of Calgary.
119. Rivals, I. and Personnaz, L. (1998): A recursive algorithm based on the extended Kalman filter for the training of feedforward neural models, Neurocomputing 20(1-3): pp. 279-294.
120. Rogers, R.M. (2003): Applied Mathematics in Integrated Navigation Systems, Second Edition. American Institute of Aeronautics and Astronautics, Inc.
121. Rumelhart, D.E. and McClelland, J.L. (1986): Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations, Vol.1, MIT Press, Cambridge, MA.
122. Saad, D. (1998): On Line Learning in Neural Networks, Cambridge University Press.
123. Salychev, O. (1998): Inertial Systems in Navigation and Geophysics, Bauman MSTU Press, Moscow.
124. Salychev, O.S., Voronov, V.V., Cannon, M.E., Nayak, N., and Lachapelle, G. (2000): Low Cost INS/GPS Integration: Concepts and testing. In Proceedings of the ION National Technical Meeting, pages 98–105, Anaheim, CA
125. Savage, P.G. (2000): Strapdown Analytics Part I ,II, Strapdown associate
126. Savant, Jr, C.J., Howard, R.C., Solloway, C.B., and Savant, C.A. (1961): Principles of Inertial Navigation, McGraw–Hill.
127. Scherzinger, B. (2004): Estimation with Application to Navigation, ENGO 699.11 lecture notes, Department of Geomatics Engineering, The University of Calgary.

128. Schwarz, K.P. and El-Sheimy, N. (1999): Future Positioning and Navigation Technologies. Department of Geomatics Engineering, The University of Calgary, Calgary, Canada.
129. Schwarz, K.P. and Wei, M. (2000): INS/GPS Integration for Geodetic Applications: Lecture Notes ENGO 623. Department of Geomatics Engineering., The University of Calgary, Calgary, Canada.
130. Shi, X.H., Liang, Y.C, and Xu, X. (2003): An improved Elman model and recurrent back-propagation control neural networks. *Journal of Software*, 2003, 14(6): pp. 1110~1119.
131. Shin, E.H. (2001): Accuracy Improvement of Low Cost INS/GPS for Land Application, (M.Sc. thesis), Dec. 2001, UCGE Report No. 20156. Department of Geomatics Engineering, The University of Calgary.
132. Shin, E. H. and El-Sheimy, N. (2004): An Unscented Kalman Filter for In-Motion Alignment of Low Cost IMUs. Proceeding of IEEE PLANS 2004, Monterey, California, USA.
133. Singhal, S. and Wu, L.(1989): Training Multilayer Perceptrons with the Extended Kalman Algorithm, *Advances in Neural Information Processing System I*, Morgan Kaufmann, San Mateo, CA, pp. 133-140.
134. Skaloud, J. (1999): Optimizing Georeferencing of Airborne Survey Systems by INS/DGPS, Department of Geomatics Engineering, University of Calgary, Calgary, Canada, UCGE Report 20126.
135. Smagt, P.(1994): Minimization Methods for Training Feedforward Neural Networks, *Neural Networks*, Vol.7, No.1,pp. 1-11.
136. Sparto, P. J., Parnianpour, M., Barria, E.A. and Jagadeesh, J.M. (1999): Wavelet Analysis of Electromyography for Back Muscle Fatigue Detection During Isokinetic Constant - Torque Exertions", *SPNE*, Vol. 24, No.17, pp. 1791-1798.
137. Strang, G.(1993): Wavelet transforms versus Fourier transforms, *Bull. (New Series) Amer. Math. Soc.*, Vol. 28, No. 2, 1993, pp. 288-305.
138. Strang, G. and Nguyen, T. (1997): *Wavelets and Filter Banks*.Wellesley-Cambridge Press.

139. Sukkarieh, S. (2000): Low Cost, High Integrity, Aided Inertial Navigation Systems for Autonomous Land Vehicles. Ph.D. Thesis, Australian Centre for Field Robotics, Dept. of Mechanical and Mechatronic Engineering, The University of Sydney, Sydney, Australia.
140. Sunkel, H. (2002): Advanced Mathematical Methods and Application in Geoscience; ENGO 699.72 lecture notes, Department of Geomatics engineering ,The University of Calgary, May 2002.
141. Titterton, D. H. and Weston, J. L. (1997): Strapdown Inertial Navigation Technology, Peter Peregrinus Ltd.
142. Townsend, N.W., Brownlow, M., and Tarassenko L.(1994): Radial Basis Function Networks for Mobile Robot Localisation, Proceedings of the 1994 INNS World Conference On Neural Networks, Vol.2, pp. 9-14.
143. Tsakiri, M. (1996): Integrated GPS and DR for Land Vehicle Navigation, The Proceeding of WALIS forum.96 April 23-24, 1996.
144. Vanícek, P. and Omerbasic M. (1999): Does a navigational algorithm have to use Kalman filter? Canadian Aeronautical and Space Institute Journal, Vol. 45, No. 3, pp. 292-296.
145. Vetterli, M. and Herley, C. (1992): Wavelets and Filter Banks: Theory and Design, IEEE Transactions on Signal Processing, Vol. 40, pp. 2207-2232.
146. Vidyasagar, M. (2002): Learning and Generalization, Springer-Verlag New York
147. Wu, S.I. (1995): Mirroring Thought processes, IEEE Potentials 14, pp. 36-41.
148. Xu, B. (1996): A New Navigation Filter. PhD Thesis, Technical Report No. 182, The Department of Geodesy and Geomatics Engineering, University of New Brunswick, Fredericton, NB Canada.
149. Zhao, Y. (1997): Vehicle Location and Navigation Systems, Artech House Publishers.

APPENDIX A

A.1 Examples of Associated Frequency

Table.A.1: Digested associated frequency of DB3 ($\omega=0.8$)

scale	Fs=200 Hz		Fs=100 Hz		Fs=50 Hz	
	F_p (Hz)	F_A (Hz)	F_p (Hz)	F_A (Hz)	F_p (Hz)	F_A (Hz)
1	160	128	80	64	40	32
2	80.0000	64.0000	40.0000	32.0000	20.0000	16.0000
3	53.3333	42.6667	26.6667	21.3333	13.3333	10.6667
4	40.0000	32.0000	20.0000	16.0000	10.0000	8.0000
5	32.0000	25.6000	16.0000	12.8000	8.0000	6.4000
6	26.6667	21.3333	13.3333	10.6667	6.6667	5.3333
7	22.8571	18.2857	11.4286	9.1429	5.7143	4.5714
8	20.0000	16.0000	10.0000	8.0000	5.0000	4.0000
9	17.7778	14.2222	8.8889	7.1111	4.4444	3.5556
...
15	10.6667	8.5333	5.3333	4.2667	2.6667	2.1333
16	10.0000	8.0000	5.0000	4.0000	2.5000	2.0000
...
31	5.1613	4.1290	2.5806	2.0645	1.2903	1.0323
32	5.0000	4.0000	2.5000	2.0000	1.2500	1.0000
...
64	2.5	2.00	1.25	1.00	0.625	0.5
...
128	1.25	1.00	0.625	0.5	0.3125	0.25

Table A.2: Digested associated frequency of DB5 ($\omega=0.96$)

scale	Fs=200 Hz		Fs=100 Hz		Fs=50 Hz	
	F_p (Hz)	F_A (Hz)	F_p (Hz)	F_A (Hz)	F_p (Hz)	F_A (Hz)
1	133.3333	128.000	66.6667	64.0000	33.333	32.0000
2	66.6667	64.0000	33.3333	32.0000	16.6667	16.0000
3	44.4444	42.6667	22.2222	21.3333	11.1111	10.6667
4	33.3333	32.0000	16.6667	16.0000	8.3333	8.0000
...
7	19.0476	18.2857	9.5238	9.1429	4.7619	4.5714
8	16.6667	16.0000	8.3333	8.0000	4.1667	4.0000
9	14.8148	14.2222	7.4074	7.1111	3.7037	3.5556
...
15	8.8889	8.5333	4.4444	4.2667	2.2222	2.1333
16	8.3333	8.0000	4.1667	4.0000	2.0833	2.0000
32	4.1667	4.0000	2.0833	2.0000	1.0417	1.0000
..
64	2.08	2.00	1.04	1.00	0.52	0.5
...
128	1.04	1.00	0.52	0.50	0.26	0.25

A.2 Multiresolution Analysis

A MRA of $L^2(R)$ is defined as a sequence of closed space V_j of $L^2(R)$ satisfying the following properties [Mallat, 1989a and 1989b]:

- $V_{j+1} \subset V_j$,
- $v(t) \in V_j \Leftrightarrow v(2t) = V_{j+1}$,
- $v(t) \in V_0 \Leftrightarrow v(t+1) = V_0$,
- $\bigcup_{j=-\infty}^{\infty} V_j$ is dense in $L^2(R)$ and $\bigcap_{j=-\infty}^{\infty} V_j = \{0\}$,
- A scaling function $\varphi \in V_0$ with non-vanishing integral exists such that $\{\varphi(t-l) | l \in \mathbb{Z}\}$ is a Riesz basis of V_0 .

Where $L^2(R)$ is Hilbert space and Z is integer. The following terminology will be used:

a level of a MRA is one of the V_j subspaces and one level is *coarser* (respectively *finer*) with respect to another whenever the index of the corresponding space is *bigger* (respectively *smaller*). The last property states that a basis must exist follows for the space V_0 . It follows from the definition of the MRA that there exists a sequence (h_k) such that the scaling function $\varphi(t)$ satisfies [Mallat, 1989a and 1989b].

$$\varphi(t) = 2 \sum_k h_k \varphi(2t - k) \quad (\text{A.1})$$

This is known as the refinement equation, dilation equation, or scaling function. The wavelet or detail spaces W_j are spaces complementing V_j in V_{j-1} i.e. a space that satisfies

$$V_{j-1} = V_j \oplus W_j \quad (\text{A.2})$$

This says that every element of V_{j-1} can be written as a sum of an element in V_j and W_j in a unique way. The detail spaces W_j contains the detail information needed to go from an approximation at resolution $j-1$ to approximation at resolution j , consequently,

$$\oplus W_j = L^2(R) \quad (\text{A.3})$$

The wavelet function ψ exists if the collection of functions $\{\psi(t-l) | l \in \mathbb{Z}\}$ is a Riesz basis of W_0 . The collection of wavelet functions $\{\psi_{j,l} | l, j \in \mathbb{Z}\}$ is then a Riesz basis of $L^2(R)$. Since the wavelet ψ is an element of V_1 , a sequence (g_k) such that the wavelet function $\psi(t)$ satisfies [Mallat, 1989a and 1989b]

$$\psi(t) = 2 \sum_k g_k \varphi(2t - k) \quad (\text{A.4})$$

Consequently,

$$v_j(t) = \sum_k \lambda_{j,k} \varphi_{j,k} = v_{j+1}(t) + w_{j+1}(t) = \sum_l \lambda_{j+1,l} \varphi_{j+1,l}(t) + \sum_l \gamma_{j+1,l} \psi_{j+1,l}(t) \quad (\text{A.5})$$

Figure (A.1) also illustrates the examples of scaling and wavelet function.

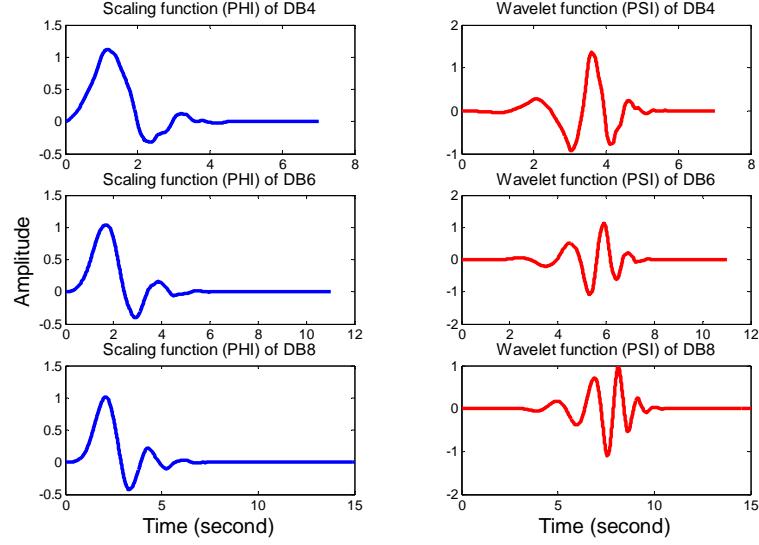


Figure A.1: Examples of wavelet and scaling functions

In other words, there are two representations of the function v_j , one as an element in V_j and associated with the sequence $\{\lambda_{j,k}\}$, and another as a sum of elements in V_{j+1} and W_{j+1} and associated with the sequences $\{\lambda_{j+1,k}\}$ (approximation sequence) and $\{\gamma_{j+1,k}\}$ (detail sequence) that are given as follow:

$$\lambda_{j+1,l} = \langle v_j, \tilde{\varphi}_{j+1,l} \rangle = \sqrt{2} \left\langle v_j, \sum_k h_{k-2l} \tilde{\varphi}_{j,k} \right\rangle = \sqrt{2} \sum_k h_{k-2l} \lambda_{j,k} \quad (\text{A.6})$$

$$\gamma_{j+1,l} = \sqrt{2} \sum_k g_{k-2l} \lambda_{j,k} \quad (\text{A.7})$$

The opposite direction from, $\{\lambda_{j+1,l}\}$ and $\{\gamma_{j+1,l}\}$ to $\{\lambda_{j,k}\}$, is given as follow:

$$\lambda_{j,k} = \sqrt{2} \sum_l \tilde{h}_{k-2l} \lambda_{j+1,l} + \sqrt{2} \tilde{g}_{k-2l} \gamma_{j+1,l} \quad (\text{A.8})$$

When applied recursively, these formulate define the Fast Wavelet Transform (FWT); the equation A.6 and A.7 define the forward transform, while Equation A.8 defines the inverse transform [Cody, 1992], as shown in the Figure (A.2).

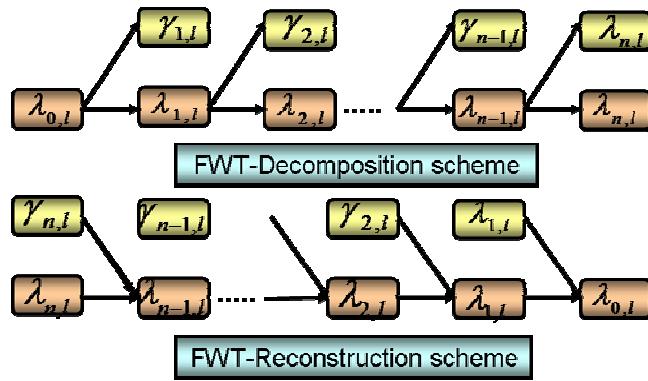


Figure A.2: Schemes of FWT

Prior to analyzing the spectrum of DWT, the distinction between approximation/detail signal (A/D) and approximation /detail coefficients (cA/cD) is given. For many signals, the low frequency content is the most important part. It is what gives the signal its identity. The high-frequency content, on the other hand, imparts flavor or nuance. For example, removing the high-frequency components of human voice, the voice sounds different, but it is possible for a person to tell what's being said. However, removing enough of the low-frequency components, only gibberish can be heard. In MRA, approximations and details terms are used. The approximations are the high-scale, low-frequency components of the signal. The details are the low-scale, high-frequency components [Vetterli and Herley, 1992].

Figure (A.3) illustrates the distinction between approximation/detail signal (A/D) and approximation /detail coefficients (cA/cD). The original signal, \mathbf{S} , passes through two

complementary filters and emerges as two signals. Unfortunately, if this operation on a real digital signal is performed, it results in with twice as much data as the beginning. Suppose, for instance, that the original signal S consists of 1000 samples of data. Then the resulting signals will each have 1000 samples, for a total of 2000. These signals A and D are interesting, but it gets 2000 values instead of the 1000. As mentioned previously, there exists a more subtle way to perform the decomposition using wavelets. By looking carefully at the computation, we may keep only one point out of two in each of the two 2000-length samples to get the complete information. This is the notion of downsampling which produces two sequences called cA and cD .

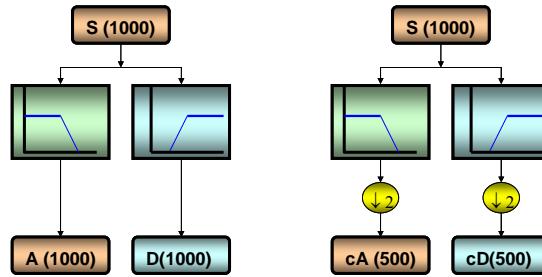


Figure A.3: Distinctions between (A/D) and (cA/cD)

To illustrate this process, Figure (A.4) depicts the distinctions between approximate/detail signal and DWT coefficients. The decomposition is three and the result shows an important fact that the length (sampling rate) of approximation (A) and detail signals is identical to that of original signal. On the contrary, the length (sampling rate) of DWT coefficients at each decomposition level decreases at rate of 2^k , k is the decomposition level. It is a well known fact that the bandwidth, Nyquist frequency, F_n , (maximum detectable frequency) of Fourier analysis is mainly related to the sampling rate F_s . In fact, the relationship between F_n and F_s can be described as followed.

$$F_n = \frac{F_s}{2} \quad (\text{A.9})$$

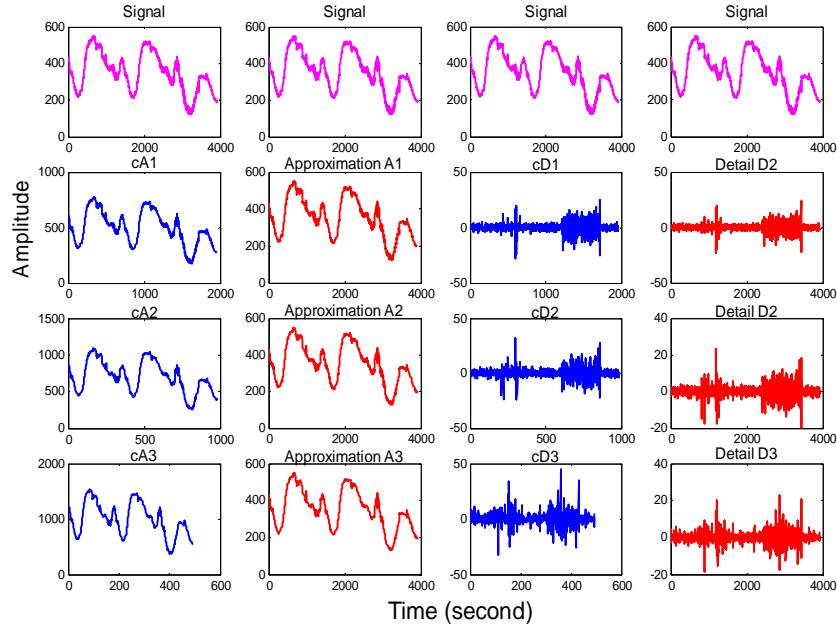


Figure A.4: Example of the distinction between (A/D) and a (cA/cD)

Consequently, the relationship between decomposition level, maximum detectable frequency F_n and sampling rate F_s (i.e., 200Hz, 100Hz and 50 Hz) is given in Table A.3.

Table A.3: Frequencies and decomposition level

	A/D		cA/cD	
	F_n (Hz)	F_s (Hz)	F_n (Hz)	F_s (Hz)
DL=0	100	200	100	200
DL=1	100	200	50	100
DL=2	100	200	25	50
DL=3	100	200	12.5	25
DL=4	100	200	6.25	12.5
	F_n (Hz)	F_s (Hz)	F_n (Hz)	F_s (Hz)
DL=0	50	100	50	100
DL=1	50	100	25	50
DL=2	50	100	12.5	25
DL=3	50	100	6.25	12.5
DL=4	50	100	3.125	6.25
	F_n (Hz)	F_s (Hz)	F_n (Hz)	F_s (Hz)
DL=0	25	50	25	50
DL=1	25	50	12.5	25
DL=2	25	50	6.25	12.5
DL=3	25	50	3.125	6.25
DL=4	25	50	1.5625	3.125

A.3 Performance Analysis of Different Thresholding Algorithm

Figure (A.5), and Table A.4 illustrates the performance using different thresholding algorithm and the 1st generation denoising algorithm. A clean signal $X(n)$ was generated the noise with certain SNR was added. In addition, the denoised signal is denoted as $\hat{X}(n)$. The evaluations of denoising performance can be carried out according to [Dohono, 1992]:

- Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} [\hat{X}(n) - X(n)]^2} \quad (\text{A.10})$$

- Signal to noise ratio (SNR) in decibels.

$$SNR(X, \hat{X}) = 10 \log_{10} \frac{\sum_{n=0}^{N-1} |X(n)|^2}{\sum_{n=0}^{N-1} |\hat{X}(n) - X(n)|^2} \quad (\text{A.11})$$

Table A.4: Performance Comparison (DWT)

	SNR (Original =7)	RMS
Soft	30.13	0.23
Hard	30.28	0.22
Universal	31.02	0.18
No	28.44	0.28

Be aware that the no thresholding operation was conducted by extracting the approximation signal (A) at designated decomposition level. Figure (A.6), and Table A.5 illustrates the performance using different thresholding algorithm and the 1st generation denoising algorithm.

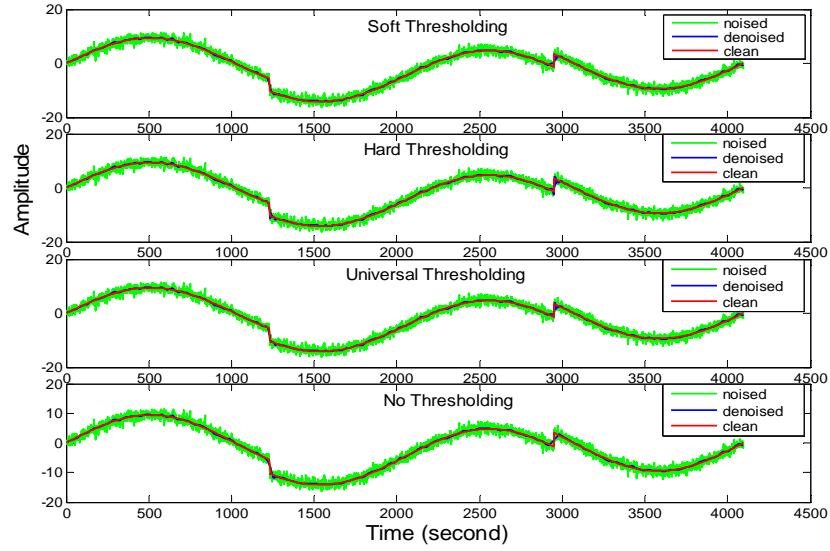


Figure A.5: DWT Denoising Examples

Table A.5: Performance Comparison (TIW)

	SNR (Original =7)	RMS
Soft	30.25	0.22
Hard	31.07	0.17
Universal	31.53	0.15

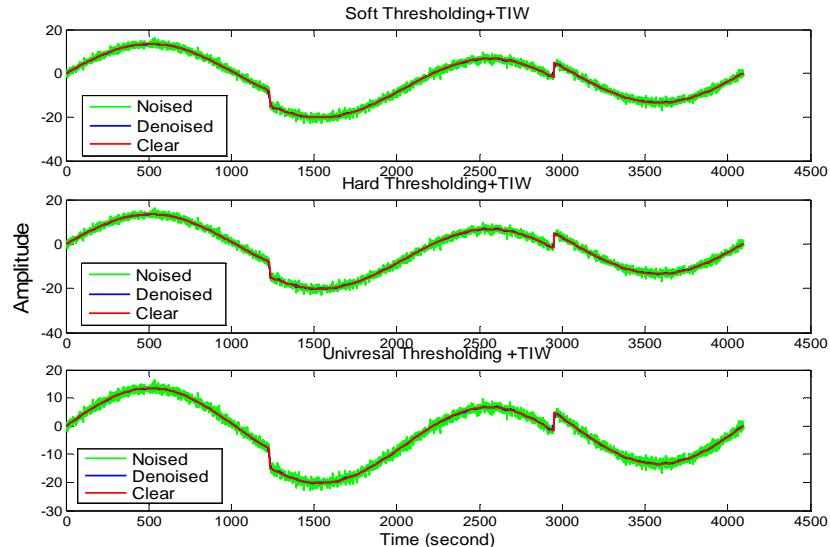


Figure A.6: TIW Denoising Examples

A.4 Performance Analysis of TIW Lowpass Filter Denoising

For comparison, Figure (A.7) and Table A.6 illustrate the performance of the wavelet based low pass filters using TIW. The stop bands of those low pass filters were decided using Table 4.5.

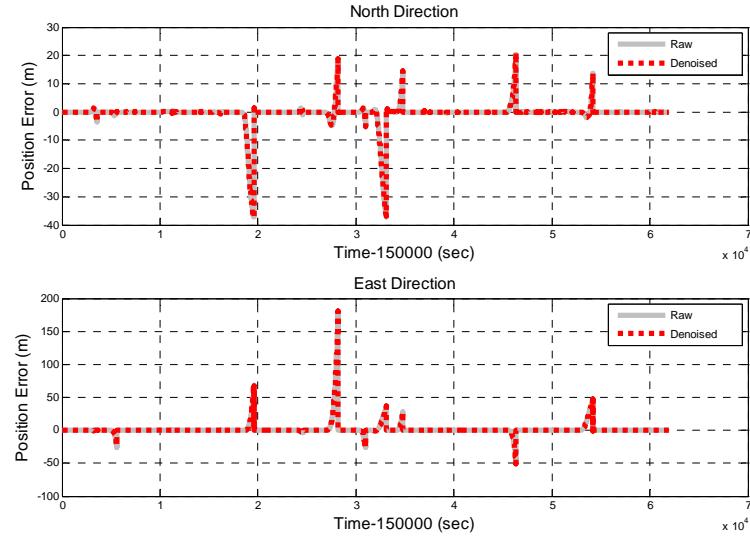


Figure A.7: Position errors during GPS blockages

Table A.6: Performance summary of wavelet low pass filter (TIW)

Blockage no.	RAW				Low Pass Filter (TIW)		
	Blockage Length(S)	MAX_N (m)	MAX_E (m)	RMS Total	MAX_N (m)	MAX_E (m)	RMS Total
1	30	3.43	1.12	1.18	3.28	1.25	2.09
2	30	1.23	25.82	9.89	2.95	23.54	9.08
3	60	37.07	68.27	33.86	37.08	69.16	34.02
4	30	1.45	3.50	1.84	2.03	2.25	1.64
5	60	18.90	179.30	70.58	19.90	158.72	63.36
6	30	5.16	25.53	10.92	5.54	24.70	10.68
7	70	37.04	36.97	25.23	36.41	31.09	22.80
8	30	14.39	27.27	15.26	14.45	28.91	15.62
10	44	20.39	50.52	18.43	20.93	43.50	15.76
11	60	13.50	47.48	23.69	16.83	44.01	22.75

A.5 Performance Analysis of Cascade Denoising Algorithm (LN200)

Figure (A.8) and Table A.7 indicate the performance summary of the positional errors generated by comparing the raw and denoised INS (LN200)/DGPS integrated navigation solutions with the reference trajectory during each GPS outage, respectively.

Table A.7: Performance summary of positional errors (LN200/DGPS)

Blockage no.	RAW				Cascade Denoising _DGPS		
	Blockage Length(s)	MAX_N (m)	MAX_E (m)	RMSE Total(m)	MAX_N (m)	MAX_E (m)	RMSE Total(m)
1	30	4.16	19.56	6.12	4.79	21.45	6.85
2	30	9.48	8.11	4.83	10.46	9.28	4.90
3*	60	5.03	1.66	2.84	2.16	2.99	1.92
4	60	25.50	11.04	12.95	26.40	12.67	13.56
5*	60	2.86	7.70	2.99	2.51	7.38	2.77
6*	30	3.59	3.06	2.20	3.37	0.20	1.68
7*	60	28.56	9.15	14.28	7.08	3.03	3.92
8*	120	42.77	19.67	18.28	22.32	12.48	8.91

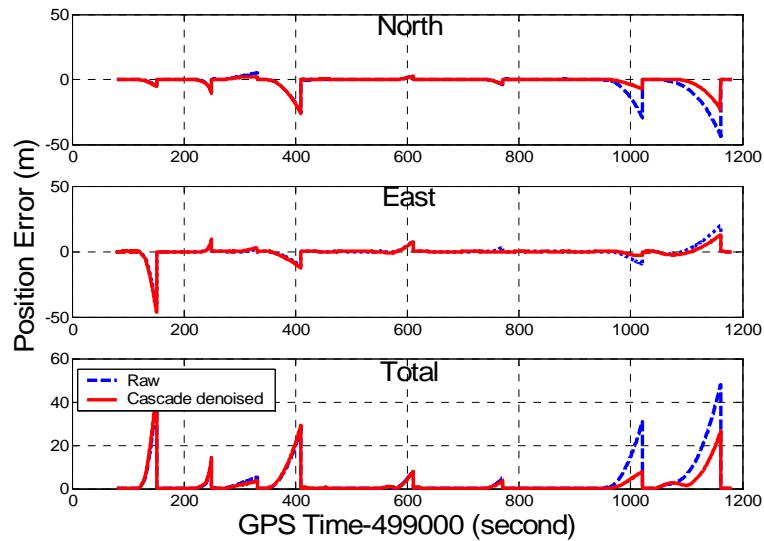


Figure A.8: Position errors (LN200/DGPS)

Figure (A.9) and Table A.8 illustrate the performance summary associated the position errors generated by comparing the denoised INS/SPP integrated navigation solutions, the raw INS/SPP integrated navigation solutions with the reference trajectory during each GPS outage respectively.

Figure (A.10) compares the impact of proposed cascade denoising algorithm on the INS/DGPS and INS/SPP integrated systems in terms of the percentage of improvement or degradation during each GPS outage period, respectively.

Table A.8: Performance summary of positional errors (LN200/SPP)

Blockage no.	RAW				Cascade Denoising_SPP		
	Blockage Length(s)	MAX_N (m)	MAX_E (m)	RMSE Total(m)	MAX_N (m)	MAX_E (m)	RMSE Total(m)
1	30	5.02	31.18	15.96	5.31	39.70	15.08
2	30	5.85	3.78	5.51	6.38	7.45	7.32
3*	60	17.01	44.84	27.01	24.21	27.66	18.10
4*	60	29.46	42.64	24.23	29.52	9.49	20.97
5*	60	53.61	9.35	22.78	10.87	4.65	7.10
6	30	2.35	10.16	5.58	12.24	6.29	7.04
7*	60	23.37	21.19	13.42	4.39	17.31	10.52
8*	120	105.56	66.55	45.39	65.77	21.76	31.64

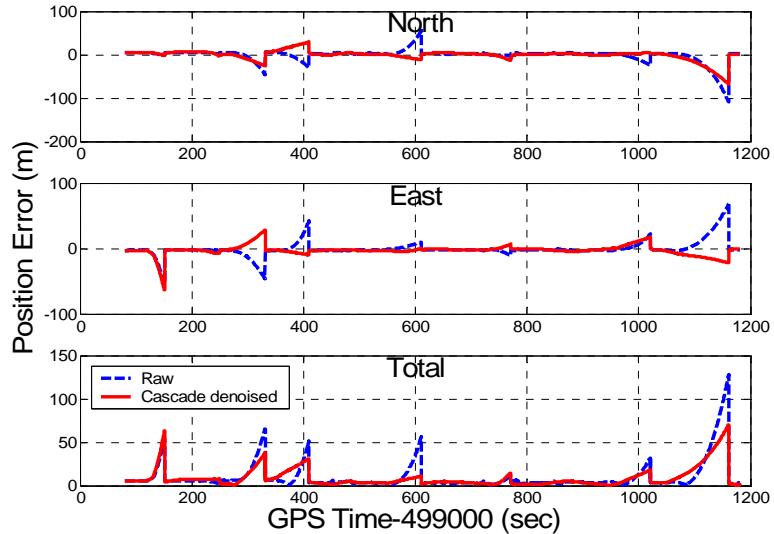


Figure A.9: Position errors (LN200/SPP)

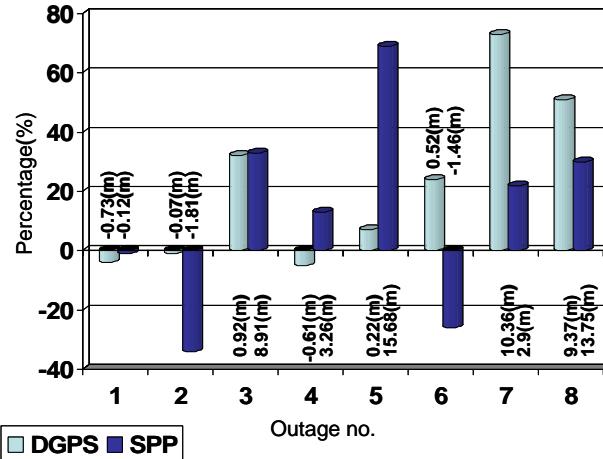


Figure A.10: The impact of denoising on the integrated systems (LN200/GPS)

A.6 Performance Analysis of Cascade Denoising Algorithm (XBOW)

Figure (A.11) and Table A.9 indicate the performance summary associated the position errors generated by comparing the denoised INS/DGPS integrated navigation solutions, the raw INS/DGPS integrated navigation solutions using a MEMS IMU, Crossbow DMU AHRS-400CC (XBOW), with the reference trajectory during each GPS outage respectively.

Table A.9: Performance summary positional errors (XBOW/DGPS)

Blockage no.	Blockage Length(s)	RAW			Cascade Denoising _DGPS		
		MAX_N (m)	MAX_E (m)	RMSE Total(m)	MAX_N (m)	MAX_E (m)	RMSE Total(m)
1	30	5.09	25.51	12.44	5.09	28.36	15.12
2	30	3.35	42.79	19.06	3.41	18.18	8.31
3	60	57.30	196.42	86.90	27.24	29.01	15.50
4	60	7.84	22.51	15.36	13.42	33.48	19.24
5	60	74.86	74.97	45.64	91.59	43.19	44.33
6	30	15.34	25.28	11.62	17.12	11.47	8.13
7	60	83.50	53.05	42.59	53.01	12.89	21.49
8	120	355.01	163.66	172.47	269.10	63.47	137.28

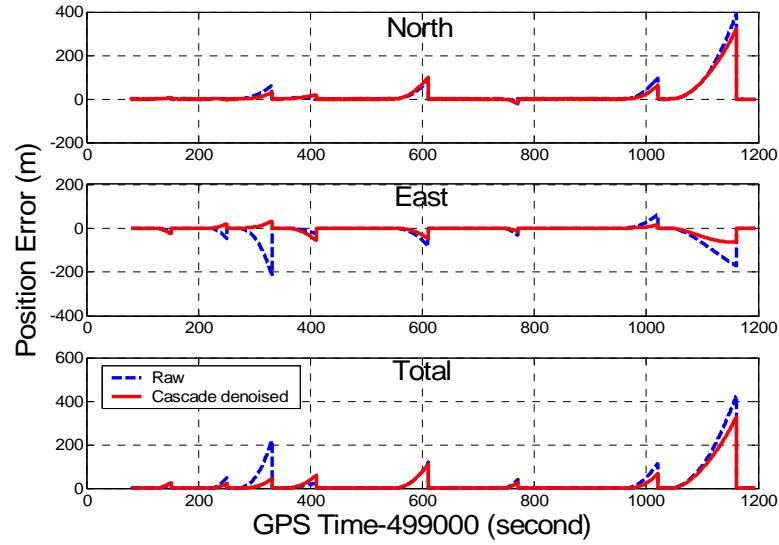


Figure A.11: Position errors (XBOW/DGPS)

Figure (A.12) and Table 7.9 depict the performance summary associated the position errors generated by comparing the denoised INS/SPP integrated navigation solutions, the raw INS/SPP integrated navigation solutions with the reference trajectory during each GPS outage period respectively. Figure (A.13) compares the impact of proposed cascade denoising algorithm on the INS/DGPS and INS/SPP integrated systems in terms of the percentage of improvement or degradation during each GPS outage period.

Table A.10: Performance summary of positional errors (XBOW/SPP)

Blockage no.	RAW				Cascade Denoising_SPP			
	Blockage Length(s)	MAX_N (m)	MAX_E (m)	RMSE Total(m)	MAX_N (m)	MAX_E (m)	RMSE Total(m)	
1	30	13.02	28.56	22.78	4.75	44.78	23.35	
2	30	6.80	48.64	18.53	7.56	4.15	6.81	
3	60	36.62	236.26	103.70	39.89	54.36	28.21	
4	60	23.97	130.85	61.63	132.21	3.61	62.84	
5	60	271.55	118.25	128.87	116.14	93.01	65.93	
6	30	61.97	150.98	64.03	20.46	2.52	8.36	
7	60	29.33	45.49	27.11	11.52	58.57	27.13	
8	120	486.07	63.90	220.08	364.72	26.69	163.18	

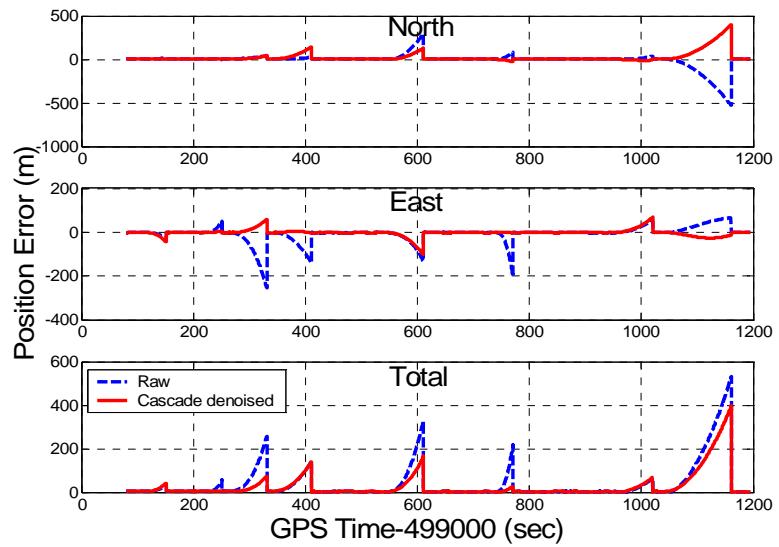


Figure A.12: Position errors (XBOW/SPP)

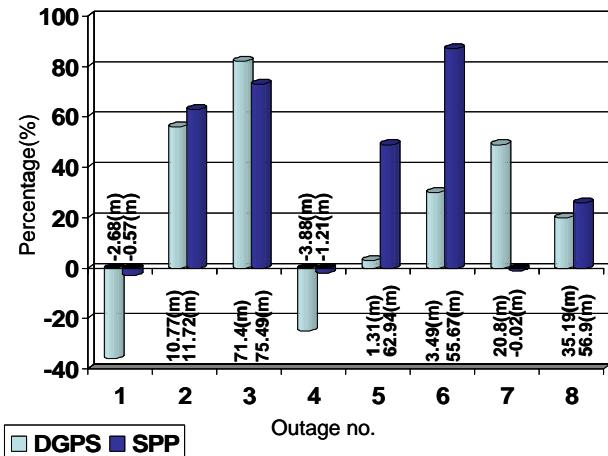


Figure A.13: The impact of denoising on the integrated systems (XBOW/GPS)

APPENDIX B

B.1 Biological Neurons and Neural Networks

The neural system of the human body consists of three components: receptors, a neural network, and effectors. The receptors receive the stimuli either internally or from the external world, and then transmit the information into the neurons via the form of electrical impulses. The neural network then processes the inputs then makes proper decision of outputs. Finally, the effectors translate electrical impulses from the neural network into responses to the outside environment. Figure (B.1) shows the bidirectional communication between components for feedback [Arbib, 1987].

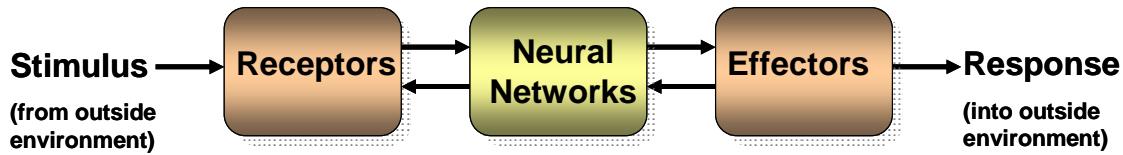


Figure B.1: Three stages of biological neural networks

The fundamental processing element of the neural network is called a neuron. As shown in figure (B.2). This biological cell receives and processes information and communicates with various part of the human body. The nerve cell body is called the soma and is surrounded by a thin plasma membrane filled with cytoplasm. The soma is approximately 30 μm in diameter. Within the soma is a cell nucleus. Every nerve cell receives many inputs through the dendrites, and after some processing generates a single output along its axon. The junction point of an axon with a dendrite is called the synapse. The information generated by a neuron is transmitted along its axon.

In fact, the human brain has more than 10 billion neurons, which have complicated interconnections, and these neuron architectures provide a large-scale signal processing and memory neural networks. Therefore, the mathematical study of a single neuron model and its various extensions is the first step in the design of a complex neural

network for application such as neural signal processing, pattern recognition, control of complex system and other sophisticated decision-making processes.

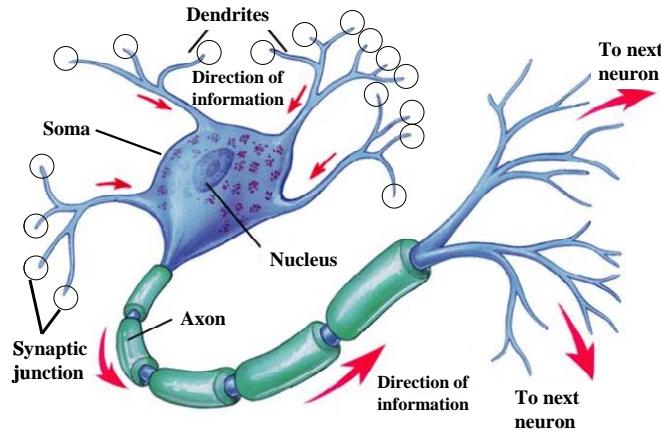


Figure B.2: A biological neuron

B.2 Second Order Optimization Learning Algorithms

As mentioned previously, the standard backpropagation algorithm (gradient descent algorithm) and the modified standard backpropagation algorithm (gradient descent algorithm plus momentum algorithm) employ only the first order partial derivatives of the error function (gradient). Such first order learning algorithms have weakness in dealing with a large number of classification and function approximation problems [Master, 1995]. They usually required large number of iterations to optimally tune the synaptic weights thus it is not possible to adapt them for on-line application, such as adaptive control. As a result, recent development of backpropagation algorithm utilizes the high order optimization algorithms to speed up the learning speed [Battiti, 1992].

In terms of numerical analysis, not only the local gradient but also the second derivatives are usually applied to solve nonlinear problems. The first order optimization algorithm uses the linear terms that include the first order derivatives in the Taylor series expansion to approximate a function. On the contrary, the second order optimization algorithm applies up to the second order derivatives in the Taylor series thus a more precise function approximation can be obtained [Anderson and McNeill, 1992].

Given a vector w_0 in the weight space, a second order Taylor series approximation of the error function around this vector is expressed as [Fine, 1999];

$$E(w) = E(w_0) + g\Delta w + \frac{1}{2}(\Delta w)^T H \Delta w \quad (\text{B.1})$$

$$g = \left. \frac{\partial E}{\partial w} \right|_{w=w_0} \quad \text{and} \quad H = \left. \frac{\partial E^2}{\partial w^2} \right|_{w=w_0} \quad (\text{B.2})$$

Applying equation B.1 yields

$$g = \frac{\partial E(w)}{\partial w} = \frac{1}{2} \begin{bmatrix} \frac{\partial \sum_{p=1}^m e_p^2}{\partial w_1} \\ \vdots \\ \frac{\partial \sum_{p=1}^m e_p^2}{\partial w_N} \end{bmatrix} = \begin{bmatrix} \sum_{p=1}^m e_p \frac{\partial e}{\partial w_1} \\ \vdots \\ \sum_{p=1}^m e_p \frac{\partial e}{\partial w_N} \end{bmatrix} = J^T e \quad (\text{B.3})$$

$$H = \begin{bmatrix} \frac{\partial^2 E(w)}{\partial w_1^2} & \frac{\partial^2 E(w)}{\partial w_1 \partial w_2} & \dots & \frac{\partial^2 E(w)}{\partial w_1 \partial w_N} \\ \frac{\partial^2 E(w)}{\partial w_2 \partial w_1} & \frac{\partial^2 E(w)}{\partial w_2^2} & \dots & \frac{\partial^2 E(w)}{\partial w_2 \partial w_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 E(w)}{\partial w_N \partial w_1} & \frac{\partial^2 E(w)}{\partial w_N \partial w_2} & \dots & \frac{\partial^2 E(w)}{\partial w_N^2} \end{bmatrix} \quad (\text{B.4})$$

Where g is the gradient and H is the Hessian matrix. Thus the minimum of the function E are located where the gradient of E is Zero

$$\frac{\partial E}{\partial w} = g + H \Delta w = 0 \quad (\text{B.5})$$

Thus the optimal value of w is given by

$$w = w_0 + \Delta w = w_0 - H^{-1}g \quad (\text{B.6})$$

Equation B.6 is known as the Newton's method. As indicated in the above equation, the computation of the inversion of the Hessian matrix is the key issue for the second order optimization. However, the inversion of the H might not exist as H might not be positive definitive in some cases. Furthermore, the direct evaluation of the true H and its inversion is computationally too expensive if H is positive definitive. In general, second order optimization learning algorithm fall into two categories:

- **Direct second order learning algorithm**

Such algorithm involves the computation of the second order derivatives (Hessian H) or their alternatives directly; two of the most popular direct second order learning algorithms are given as follow.

- **Quasi-Newton algorithm**

Alternative algorithm, known as quasi-Newton methods is based on equation (B.6). But instead of calculating H directly, and then calculating its inversion they build up an approximation to the inversion of H over a number of steps. It first generates the iterative algorithm to the form

$$w(k+1) = w(k) - M(k)g(k) \quad (\text{B.7})$$

Thus the objective of the quasi-Newton methods is to iteratively compute matrices $M(k)$ such that

$$\lim_{k \rightarrow \infty} M(k) = H^{-1} \quad (\text{B.8})$$

Two most commonly used update algorithm are the Davidson-Fletcher-Powell (DFP) and Broyden-Fletcher-Goldfarb-Shanno (BFGS) procedures. Only the BFGS is given here since it is generally regarded as being superior [Fine, 1995]

$$M(k+1) = M(k) + \frac{pp^T}{P^T v} - \frac{(M(k)v)v^T M(k)}{v^T M(k)v} + (v^T M(k)v)uu^T \quad (\text{B.9})$$

Where

$$p = w(k+1) - w(k),$$

$$v = g(k+1) - g(k),$$

$$u = \frac{p}{p^T v} - \frac{M(k)v}{v^T M(k)v}$$

A significant advantage of quasi-Newton methods is to accelerate the learning speed of MFNN about one order magnitude compared to the backpropagation algorithm [Battiti and Masulli, 1990]. However, its potential disadvantage is the storage requirement of M is quadratic in the number of weights of the network. For small networks this is not critical, however, for the networks with more than a few thousands weights it could lead to large memory requirements.

■ Levenberg-Marquardt Algorithm

The Levenberg-Marquardt algorithm exploits the fact the error is a sum of squares, as indicated in Equation (5.25). Applying Equation B.3, The Jacobian matrices can be obtained

$$J = \begin{bmatrix} \frac{\partial e_1}{\partial w_1} & \dots & \frac{\partial e_1}{\partial w_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial e_m}{\partial w_1} & \dots & \frac{\partial e_m}{\partial w_N} \end{bmatrix} \quad (\text{B.10})$$

An expression for H is given as follow

$$[\nabla^2 E(w)]_{k,j} = \frac{\partial^2 E(w)}{\partial w_k \partial w_j} = \sum_{p=1}^m \left(\frac{\partial e_p}{\partial w_k} \frac{\partial e_p}{\partial w_j} + e_p \frac{\partial^2 e_p}{\partial w_k \partial w_j} \right) \quad (\text{B.11})$$

Applying equation B.10

$$\nabla^2 E(w) = J^T J + S, S = \sum_{p=1}^M e_p \nabla^2 e_p \quad (\text{B.12})$$

Approaching the minimum of E, the elements of matrix S become small thus the H can be closely approximated by

$$H \approx J^T J \quad (\text{B.13})$$

Applying equation B.3 and B.6 yields

$$w(k+1) = w(k) - [J_k^T J_k]^{-1} J_k^T e_k \quad (\text{B.14})$$

The problem associated with iterative update using the above equation is the requirement of the inversion of matrix $H = J^T J$ which might be ill conditioned or singular. Thus a modification of equation B.13 is given

$$H \approx J^T J + \mu I \quad (\text{B.15})$$

Where μ is a small number and I is the identity matrix. As a result, the Levenberg-Marquardt algorithm for synaptic weights updating is given

$$w(k+1) = w(k) - [J_k^T J_k + \mu_k I]^{-1} J_k^T e_k \quad (\text{B.16})$$

Similar to quasi-Newton algorithm, the storage requirement is the main challenge of Levenberg-Marquardt algorithm [Ham and Kaotansic, 2001].

- **Indirect second order learning algorithm**

Such algorithm doesn't require the direct calculation of the second order derivatives as it works in an iterative manner. The conjugate gradient algorithm is generally considered the most powerful all purpose minimization algorithms [Ham and Kaotansic,

2001]. The synaptic weights updating algorithm that is similar to gradient descent algorithm is given as

$$w(k+1) = w(k) + \eta(k)\Delta w(k) \quad (\text{B.17})$$

As indicated in the above equation, the conjugate gradient learning algorithm includes two parameters, the time varying learning rate parameter $\eta(k)$ and the increment weight vector $\Delta w(k)$.

The time varying learning rate parameter $\eta(k)$ is updated through the following line search method

$$\eta(k) = \min_{\eta} \{E(w(k) + \eta\Delta w(k)) : \eta > 0\} \quad (\text{B.18})$$

Instead of keeping the learning rate parameter η fixed as gradient descent algorithm, conjugate gradient algorithm can seek the learning rate parameter $\eta(k)$ automatically at each step.

According to Smagt [1994], the conjugate condition for $\Delta w(k)$ is given

$$(\Delta w(k))^T H(k) \Delta w(k+1) = 0 \quad (\text{B.19})$$

The Hessian $H(k)$ is calculated at the point $w(k)$. Thus the $\Delta w(k)$ is updated as [Smagt, 1994]

$$\Delta w(k) = -g(k) + \alpha(k-1)\Delta w(k-1) \quad (\text{B.20})$$

The expression for the time varying updating parameter $\alpha(k-1)$ that satisfies the conjugate condition between $\Delta w(k)$ and $\Delta w(k+1)$ is given as [Fine, 1995];

I. The Fletcher-Reeves formulation:

$$\alpha(k) = \frac{(g(k+1))^T g(k+1)}{(g(k))^T g(k)} \quad (\text{B.21})$$

II. The Hestenes-Stiefel formulation:

$$\alpha(k) = \frac{[g(k) - g(k-1)]^T g(k)}{(\Delta w(k-1))^T [g(k) - g(k-1)]} \quad (\text{B.22})$$

III. The Polak-Ribiere formulation

$$\alpha(k) = \frac{(g(k+1))^T [g(k+1) - g(k)]}{(g(k))^T g(k)} \quad (\text{B.23})$$

The best choice of $\alpha(k)$ formulation is highly application dependent. Some investigations indicate that the Hestenes-Stiefel formulation and the Polak-Ribiere formulation can provide better performance in many cases [Bishop, 1995]. Comparing equation 5.55 and B.20, it is clear that increment weight vector $\Delta w(k)$ updating equation introduce the momentum with a time varying updating parameter $\alpha(k)$ that meets the conjugate condition.

Equations 5.55, B.17, B.18 and B.20 indicate that the conjugate gradient learning algorithm is a special type of backpropagation algorithm. The second derivatives information is applied to update the time varying learning rate parameter $\eta(k)$ and time varying momentum updating parameter $\alpha(k)$. In fact, instead of using negative gradient direction information for updating $\Delta w(k)$, the conjugate gradient learning algorithm utilizes the information about the direction search about $\Delta w(k)$ from the previous iteration to accelerate the convergence. Each direction is conjugate if the objective function is quadratic [Bishop, 1995].

The problem of gradient descent algorithm plus momentum is how to determine the learning rate parameter η and momentum constant α since the optimal value of those parameters varies from one step to another. The conjugate gradient learning algorithm

provides an effective solution to this problem as the effective learning parameter $\eta(k)$ is determined automatically by line minimization and the momentum is determined by time varying momentum updating parameter $\alpha(k)$ at each step [Moller, 1993].

Comparing to direct second order learning algorithms, the conjugate gradient algorithm requires relatively accurate line searches, while the quasi-Newton algorithm remains robust even if the line searches are performed to relatively low accuracy [Johansson et al., 1990]. Therefore, significant computational effort needs to be expended on each line search for conjugate gradient algorithm[Moller, 1993].

However, the conjugate gradient algorithm requires less storage space than either quasi-Newton or Levenberg-Marquardt Algorithm. In fact, the conjugate gradient algorithm been widely applied and has been demonstrated as a powerful industrial tool [Sunkel, 2002]. To gain more appreciation about the performance of first and second order learning algorithm in terms of the minimization, the Rosenbrock's function is applied

$$y = 100 * (x_2 - x_1^2)^2 + (1 - x_1)^2 \quad (\text{B.24})$$

The Rosenbrock's function has long been applied as a standard problem for evaluating nonlinear optimization algorithm, as shown in the Figure (B.3). The gradient descent algorithm, quasi-Newton algorithm, Levenberg-Marquardt algorithm and conjugate gradient algorithm are evaluated in term of their efficacy. The initial point of the approximation is set at (-1, 1) and the objective is to track the iteration step and computation time required for each algorithm until the global minimum point (1, 1) or the maximum iteration step is reached. The search path for each algorithm is shown in Figure (B.4) and the performance parameter is listed in Table B.1.

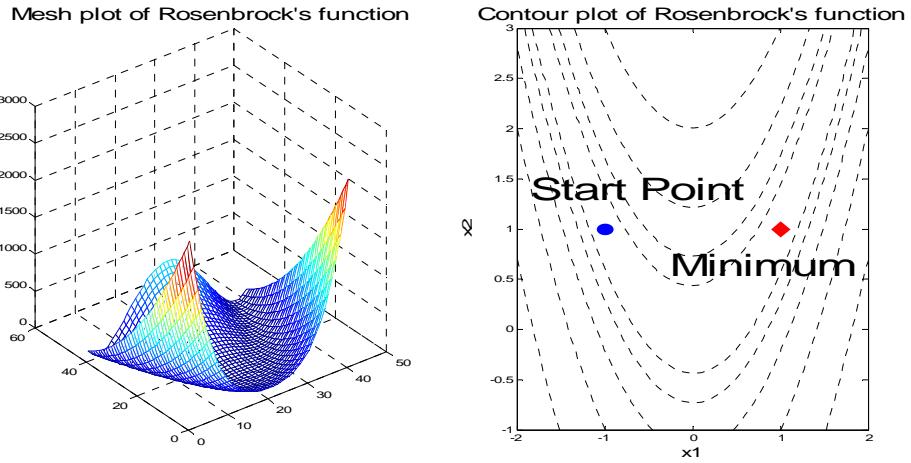


Figure B.3: Shape and contour plot of Rosenbrock's function

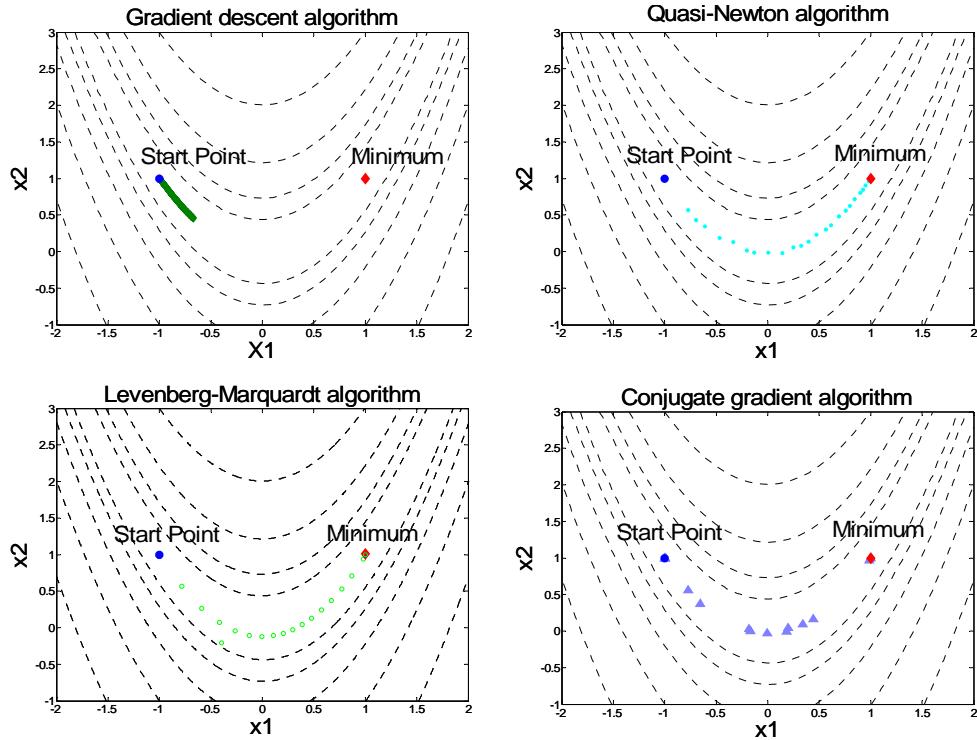


Figure B.4: Search path of the minimum

As shown in the Figure (B.4) and Table B.1, the gradient descent algorithm descended along the path very slowly and had not moved a quarter distances within the full 100 iteration steps. All other algorithms followed the curved path to the minimum reasonable

well. Conjugate gradient algorithm required the smallest amount of iteration steps to reach the global minimum; however, it spent more time than quasi-Newton and Levenberg-Marquardt since it required accurate line search procedure.

Table B.1: Performance summary

	GD	QN	LM	CG
Iteration step	>100	30	23	16
Computer time (second)	1.828	0.188	0.113	0.781

B.3 Linearized Recursive Estimation Learning Algorithms

As mentioned previously, the structure of MFNNs can be considered as a nonlinear input-output mapping, which is the heart of many problems in system identification problems, pattern recognition and nonlinear dynamic control. In fact, the weight learning problem can be regarded as the parameter identification problem of a nonlinear system with a known structure. For example, Douglas and Meng [1991] suggested a least square (LS) learning algorithm and Singhal and Wu [1989] proposed an extended Kalman filter (EKF) based learning algorithm.

According to Douglas and Meng [1991], Linearized recursive least square learning (LRLSL) can simply represent an inexplicit input and output function relationship of a MFNN by following equation

$$y(k) = f(w, x(k)), x = R^l, y = R^m \text{ and } w = R^n \quad (\text{B.25})$$

Where x and y are the input and the output of the MFNN, respectively and w is the synaptic weights. The purpose of the weight learning for a MFNN is to estimate the optimal synaptic weights thus the error between actual response y (k) and the desired response d (k) can be minimized to zero as k approaches infinity. Therefore, the synaptic weight learning process can be regarded as a parameter identification problem as the weights of a MFNN can be viewed as the parameters of a nonlinear input-output nonlinear problem.

The desired response can be modeled as

$$d(k) = f(w, x(k)) + v(k) \quad (\text{B.26})$$

Where $v(k)$ is a zero mean stationary white noise disturbance. For a nonlinear mapping function $f(w, x)$, the traditional recursive estimation techniques for linear system can not be applied to the problem. The idea is to linearize the function about the current estimate of the synaptic weight $w(k)$ to obtain a first order approximated version of the f .

$$\begin{aligned} f(w, x(k)) &\approx f(w, x(k)) + (G(k))^T (w - w(k)) \\ G(k) &= \left. \frac{\partial f(w, x(k))}{\partial w} \right|_{w=w(k)} \end{aligned} \quad (\text{B.27})$$

Where G is the Jacobian matrix dependent on the current estimate of the synaptic weight $w(k)$. Define

$$\hat{d}(k) = d(k) - f(w(k), x(k)) + (G(k))^T w(k) = G(k)^T w(k) + v(k) \quad (\text{B.28})$$

Then, $\hat{d}(k)$, which is linear in the unknown parameter w vector and can be calculated at time k using the given weight vector estimate $w(k)$. Consequently, a least square minimization to w using the new “observed” $\hat{d}(k)$. The cost function is given as

$$E(w(k)) = \frac{1}{k} \sum_{l=1}^k \alpha_l [\hat{d}(l) - (G(l))^T w]^2 \quad (\text{B.29})$$

Where α_l are appropriate weighting factors. Minimizing the above equation with respect to w yields

$$w(k+1) = \left[\sum_{l=1}^k \alpha_l G(l)(G(l))^T \right]^{-1} \sum_{l=1}^k \alpha_l G(l) \hat{d}(l) \quad (\text{B.30})$$

Defines a conditional covariance matrix as

$$R(k) = \sum_{l=1}^k \alpha_l G(l) G(l)^T = R(k-1) + \alpha_k G(k)(G(k))^T \quad (\text{B.31})$$

Applying Equations B.28, B.30 and B.31 yields

$$\begin{aligned} w(k+1) &= R^{-1}(k)[R(k-1)w(k) + \alpha_k G(k)(G(k))^T w(k) + \alpha_k G(k)v(k)] \\ &= R^{-1}(k)[R(k-1)w(k) + \alpha_k G(k)\hat{d}(k)] \end{aligned} \quad (\text{B.32})$$

Applying Equations B.31 and B.32 yields

$$\begin{aligned} w(k+1) &= R^{-1}(k)[R(k)w(k) - \alpha_k G(k)G(k)^T w(k) \\ &\quad + \alpha_k G(k)(d(k) - f(w(k), x(k)) + G(k)^T w(k))] \end{aligned} \quad (\text{B.33})$$

Cancelling terms yields

$$w(k+1) = w(k) + \alpha_k R^{-1}(k)G(k)(d(k) - f(w(k), x(k))) \quad (\text{B.34})$$

Define

$$e(k) = d(k) - f(w(k), x(k)) \quad (\text{B.35})$$

Thus, the linearized recursive least square learning algorithm is given [Douglas and Meng, 1991];

$$w(k+1) = w(k) + \alpha_k R^{-1}(k)G(k)e(k) \quad (\text{B.36})$$

$$R(k) = R(k-1) + \alpha_k G(k)(G(k))^T \quad (\text{B.37})$$

Indeed, the Equation B.36 has a similar form to the weights updating mentioned above. The elements of the product of $G(k)$ and $e(k)$ do in fact correspond to the update term of Equation 5.39 and 5.52. However, the difficulty is the computation of $R(k)^{-1}$. Since the number of weight is usually very large; this calculation introduces a significant amount of computation burden.

However, the matrix inversion lemma can be applied to simplify the computation described above by reducing the order of the matrix involved in the inverse computation. An example of the matrix inversion lemma is given [Singhal and Wu, 1989];

If A satisfies

$$A = B^{-1} + C^{-1}DC^T \quad (\text{B.38})$$

$$A^{-1} = B - BC(D + C^TBC)^{-1}C^TB \quad (\text{B.39})$$

Defining the matrix $P(k) = R^{-1}(k)$ and the gain sequence $\alpha_l = \lambda^{k-1}, 0 < \lambda \leq 1$, the following weights updating algorithm can be obtained [Singhal and Wu, 1989]

$$w(k+1) = w(k) + K(k)e(k) \quad (\text{B.40})$$

$$K(k) = P(k-1)G(k)A(k) \quad (\text{B.41})$$

$$A(k) = [\lambda I + (G(k))^T P(k-1)G(k)]^{-1} \quad (\text{B.42})$$

$$P(k) = \frac{1}{\lambda} [P(k-1) - K(k)(G(k))^T P(k-1)] \quad (\text{B.43})$$

Where A (k) is a (m*m) matrix and K (k) is an (n*m) matrix of the filter gain. Thus, the Equations B.40 to B.43 are called extended Kalman filter (EKF) based learning algorithm. Again, Equation B.40 is similar to the weight updating Equations mentioned with the error terms e (k) measured at the output layer of the network. The error is propagated to the weight through the filter gain K (k), which updates each weight through the gradient matrix G (k) and the conditional covariance matrix P (k).

B.4 Performance Analysis of Conceptual Intelligent Navigator (LN200)

Table B.2 illustrates the accuracy of different azimuth measurements obtained during the first and second field test. Figure (B.5a) and Figure (B.5b) depicts different azimuth measurements obtained during second part of the 1st and the 2nd field test, respectively. Table B.3, Figure (B.5c) and Figure (B.5d) illustrates the performance summary of VUA and AUA outputs.

Table B.2: RMSE of different azimuth measurements (LN200/DGPS)

	V (deg)	P (deg)	INS (deg)	CV(deg)	CP(deg)
1 st	69.20	68.34	5.82	5.71	5.36
2 nd	46.66	41.96	7.56	6.60	6.40

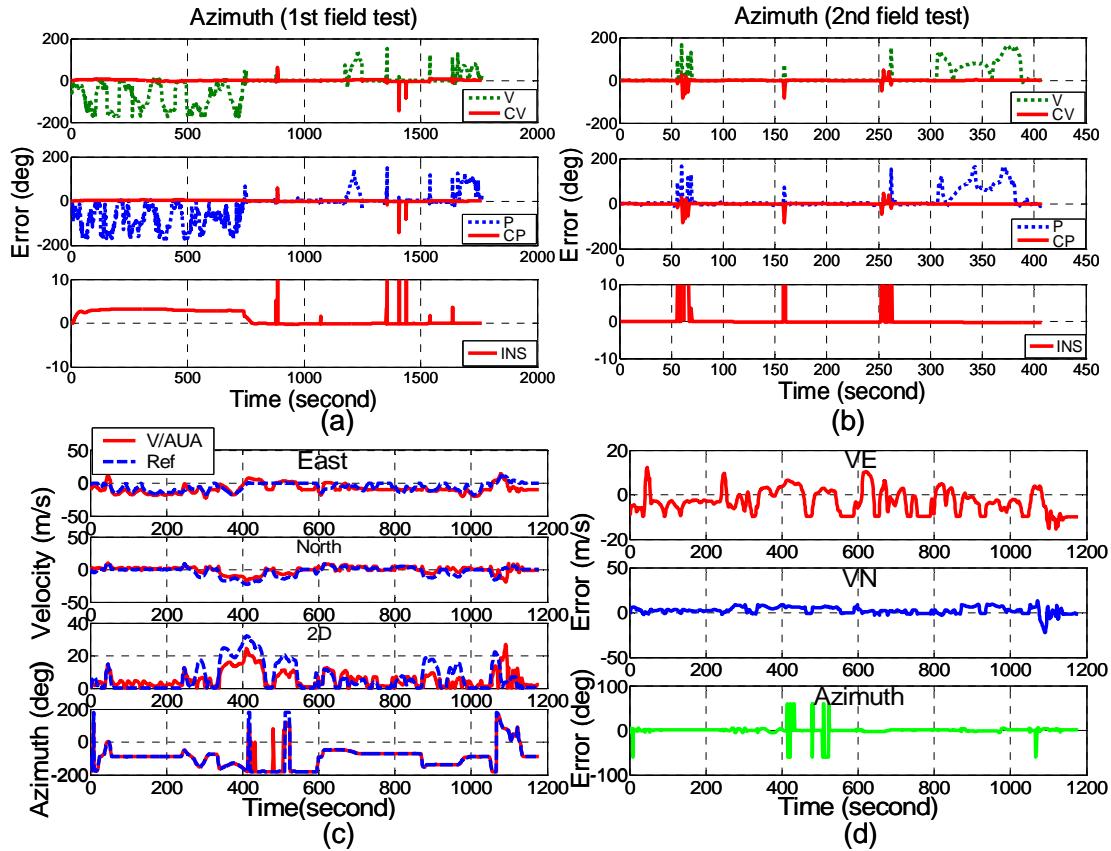


Figure B.5: Performance of CP, VUA and AUA (LN200/DGPS)

Table B.3: Performance summary of VUA and AUA (test trajectory, LN200/DGPS)

	ER_VE_VUA	ER_VN_VUA	ER_Azimuth_AUA
RMS	6.11 m/s	4.70 m/s	10.41 deg
Max	15.45 m/s	22.37 m/s	61.23 deg

B.5 Performance Analysis of Conceptual Intelligent Navigator (Xbow)

Table B.4 illustrates the accuracy of different azimuth measurements obtained during the first and second field test. Figure (B.6a) and Figure (B.6b) illustrates different azimuth measurements obtained during the first part of the 1st and the second part of the 2nd field test, respectively. Table B.5, Figure (B.6c) and Figure (B.6d) illustrates the performance summary of VUA and AUA outputs.

Table B.4: RMSE of different azimuth measurements (XBOW/DGPS)

	V (deg)	P (deg)	INS (deg)	CV(deg)	CP(deg)
1 st	45.26	47.46	27.11	16.84	14.83
2 nd	51.22	44.21	13.64	10.65	10.22

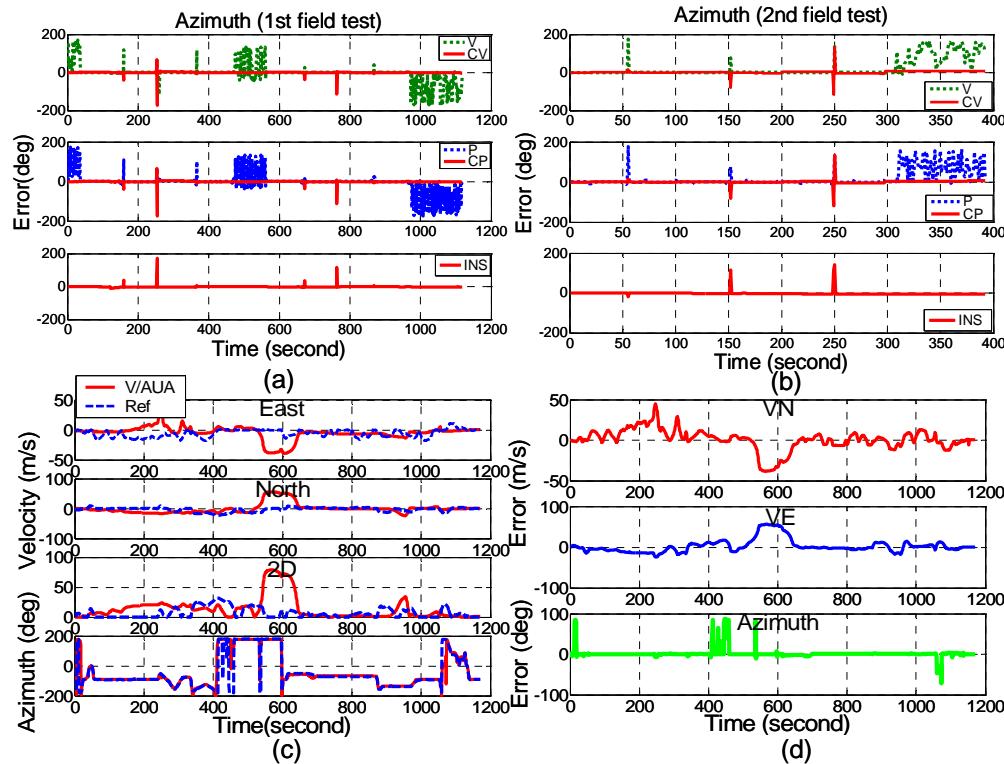


Figure B.6: CP, VUA and AUA (XBOW/DGPS)

Table B.5: Performance summary of VUA and AUA (test trajectory, XBOW/DGPS)

	ER_VE_VUA	ER_VN_VUA	ER_Azimuth_AUA
RMS	12.99 m/s	16.81 m/s	13.54 deg
Max	44.90m/s	55.95 m/s	87.86 deg

APPENDIX C

The accuracy level of an IMU can be categorized as high (strategic grade), medium (navigation grade) and low (tactical grade) in terms of the positional error, as indicated in Table C.1.

Table C.1: The category of IMUs (After El-Sheimy, 2004)

	Strategic grade	Navigation grade	Tactical grade
Positional error	<30m/hr	1-4 km/hr	20-40 km/hr
Accelerometer bias (μg)	1	50-100	100-1000
Gyro drift rate (deg/hr)	0.0001	0.015	1-10
Price	<i>Very expensive</i>	>50k USD	10k~30k USD
Application	<i>Military platform; Submarine, Intercontinental Ballistic missiles (ICBM)</i>	<i>General navigation Application and High accuracy Mobile mapping system</i>	<i>Short time Application,</i>

C.1 Specifications of Navigation Grade IMUs

The specifications of those commercial navigation grade IMUs being applied in this research are illustrated in Figure (C.1) and Figure (C.2) respectively.

CIMU

- Honeywell International Inc. (www.honeywell.com)
- Size : [193x 169 x 134 mm]
- Weight (nominal weight is less than 8 kg)
- Cost : \$ 90,000 USD~\$ 100,000 USD
- Sampling rate: 200 Hz



Technical Specification

Parameter	Accelerometer	Gyro (RLG)
Range	N/A	N/A
Bias	+/- 25(μg)	0.0022 (deg/h)
Non-linearity	N/A	N/A
Random Walk	N/A	0.0022 (deg/ h ^{1/2})
Noise	0.0025FPS / h ^{1/2}	N/A

Figure C.1: The specifications of CIMU

iNAV-RQH

- **iMAR gmbH.** (www.imar-navigation.de)
- **Size :** [350x 213 x 180 mm]
- **Weight (nominal weight is less than 8.9 kg)**
- **Cost :** \$ 150,000 USD~\$ 200,000 USD
- **Sampling rate:** 200 Hz



Technical Specification

Parameter	Accelerometer	Gyro (RLG)
Range	+/- 10 (g)	+/- 500(deg/s)
Bias	+/- 15~100(µg)	0.003~0.05 (deg/h)
Non-linearity	<1%	<0.1%
Random Walk	N/A	0.0018~0.05 (deg/ $h^{1/2}$)
Noise	8 µg / \sqrt{Hz}	N/A

Figure C.2: The specifications of iNAV-RQH

LN 90-100

- **Northup Grumman Inc.** (www.nsd.es.northropgrumman.com)
- **Sampling rate:** 64 Hz



Technical Specification

Parameter	Accelerometer	Gyro (RLG)
Range	N/A	N/A
Bias	+/- 50(µg)	0.01 (deg/h)
Misalignment	5 arcsec	1 arcsec
Random Walk	5 (µg)	0.002 (deg/ $h^{1/2}$)

Figure C.3: The specifications of LN 90-100

C.2 Specifications of Tactical Grade IMUs

The specifications of those commercial tactical grade IMUs being applied in this research are illustrated in Figure (C.4) and Figure (C.5) respectively.

HG-1700

- **Honeywell International Inc.** (www.honeywell.com)
- **Size :**[94 (dia) * 74 (ht) mm]
- **Weight (nominal weight is less than 0.9kg)**
- **Cost : \$ 20,000 USD~\$ 40,000 USD**
- **Sampling rate:** 100 Hz



Technical Specification

Parameter	Accelerometer	Gyro (RLG)
Range	+/- 50 (g)	+/- 1000(deg/s)
Bias	+/- 1(mg)	2 (deg/h)
Scale factor accuracy	0.5%	0.15%
Non-linearity	0.5% full scale	N/A
Random Walk	N/A	0.15~0.3(deg/ $h^{1/2}$)

Figure C.4: The specifications of HG-1700

LN-200

- **Northup Grumman Inc.** (www.nsd.es.northropgrumman.com)
- **Size :[89 (dia) * 85 (ht) mm]**
- **Weight (nominal weight is less than 0.7 kg)**
- **Cost : \$ 20,000 USD~\$ 40,000 USD**
- **Sampling rate: 200 Hz**



Technical Specification

Parameter	Accelerometer	Gyro (FOGs)
Range	+/- 40 (g)	+/- 1000(deg/s)
Bias	+/- 1(mg)	1 ~10 (deg/h)
Scale factor accuracy	0.3%	0.1%
Non-linearity	25 arcsec	25 arcsec
Random Walk	N/A	0.04~0.1(deg/ $h^{1/2}$)
Noise	50 μ g / \sqrt{Hz}	N/A

Figure C.5: The specifications of LN-200

C.3 Specifications of MEMS IMUs

The specifications of those three low cost IMUs being applied in this research are illustrated in Figure (C.6), Figure (C.7), and Figure (C.8), respectively.

MEMS Sensor Triad (MST)

- **MMSS Research Group**
- **Size (100x100x50 mm)**
- **Weight (nominal weight is less than 0.5kg)**
- **Cost : \$ 300 USD**
- **Sampling rate: 100 Hz**



Technical Specification

Parameter	Accelerometer	Gyro
Range	+/- 5 (g)	+/- 150(deg/s)
Bias	$+2.5V \pm 0.625V$	$+2.5V \pm 0.3V$
Scale factor	250mV/g	12.5mV/(deg/s)
Noise	$0.225mg / \sqrt{Hz}$	$0.05 \text{ deg/s} / \sqrt{Hz}$
Non-linearity	0.2% full scale	0.1% full scale
Bandwidth	32 Hz	40 Hz

Figure C.6: The specifications of MST

IMU400CC-100

- **Crossbow technology Inc.(www.xbow.com)**
- **Size (76.2x 95.3 x 81.3 mm),**
- **Weight (nominal weight is less than 0.6 kg)**
- **Cost : \$ 3,000 USD~\$ 5,000 USD**
- **Sampling rate: >100 Hz**



Technical Specification

Parameter	Accelerometer	Gyro
Range	+/- 2 (g)	+/- 100(deg/s)
Bias	<+/- 8.5 (mg)	<+/- 1.0 (deg/s)
Scale factor accuracy	<1%	<1%
Non-linearity	1% full scale	0.3% full scale
Random Walk	$<0.1 (m/s/h^{1/2})$	$<0.1 (deg/h^{1/2})$
Bandwidth	>75 Hz	>25 Hz
Data resolution	0.25 (mg)	2(arcsec/sec)

Figure C.7: The specifications of IMU400CC-100

ISI IMU

- **Inertial Science Inc.** (www.inertialscience.com)
- **Size** (72x76x58 mm)
- **Weight (nominal weight is less than 0.36kg)**
- **Cost : \$ 3,000 USD~\$ 5,000 USD**
- **Sampling rate: 200 Hz**



Technical Specification

Parameter	Accelerometer	Gyro
Range	+/- 20 (g)	+/- 90(deg/s)
Short-term bias stability	+/- 2 (mg)	<0.01 (deg/s)
Scale factor accuracy	0.2%	0.2%
Long-Term bias stability	N/A	< 1 (deg/s)
Non-linearity	0.5% full scale	0.2% full scale
Random Walk	<0.1 (m / s / h ^{1/2})	<0.5 (deg / h ^{1/2})
Bandwidth	>= 30 Hz	>=30 Hz
Data resolution	1 (mg)	2(arcsec/sec)

Figure C.8: The specifications of ISI IMU