

AN INTELLIGENT NAVIGATION SYSTEM FOR
AN AUTONOMOUS UNDERWATER VEHICLE

by

Dedy Loebis

A Thesis Submitted to the Faculty of Technology
SCHOOL OF ENGINEERING
In Partial Fulfilment of the Requirements
for the Degree of
DOCTOR OF PHILOSOPHY
THE UNIVERSITY OF PLYMOUTH

In Collaboration with the J&S Marine Ltd.,
Qinetiq, Subsea 7 and South West Water PLC

2004

© *This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without the author's prior consent.*

DEDY LOEBIS

AN INTELLIGENT NAVIGATION SYSTEM FOR
AN AUTONOMOUS UNDERWATER VEHICLE

ABSTRACT

The work in this thesis concerns with the development of a novel multisensor data fusion (MSDF) technique, which combines synergistically Kalman filtering, fuzzy logic and genetic algorithm approaches, aimed to enhance the accuracy of an autonomous underwater vehicle (AUV) navigation system, formed by an integration of global positioning system and inertial navigation system (GPS/INS).

The Kalman filter has been a popular method for integrating the data produced by the GPS and INS to provide optimal estimates of AUVs position and attitude. In this thesis, a sequential use of a linear Kalman filter and extended Kalman filter is proposed. The former is used to fuse the data from a variety of INS sensors whose output is used as an input to the later where integration with GPS data takes place. The use of an adaptation scheme based on fuzzy logic approaches to cope with the divergence problem caused by the insufficiently known *a priori* filter statistics is also explored. The choice of fuzzy membership functions for the adaptation scheme is first carried out using a heuristic approach. Single objective and multiobjective genetic algorithm techniques are then used to optimize the parameters of the membership functions with respect to a certain performance criteria in order to improve the overall accuracy of the integrated navigation system. Results are presented that show that the proposed algorithms can provide a significant improvement in the overall navigation performance of an autonomous underwater vehicle navigation.

The proposed technique is known to be the first method used in relation to AUV navigation technology and is thus considered as a major contribution thereof.

ACKNOWLEDGEMENTS

I specially would like to thank my supervisor, Professor Robert Sutton and Dr. John Chudley who have carefully read this manuscript and provided me valuable comments and discussions regarding this research work. Thanks also for their supervision throughout the period of this research programme.

I am also very grateful to the other *Hammerhead* team members, Wasif Naeem, Fraser R. Dagleish and Dr. Stephen W. Tetlow for their excellent team work during the course of the project.

Thanks also to the other member of MIDAS group, Chris, Mike, Seon, Chirag and Michelle.

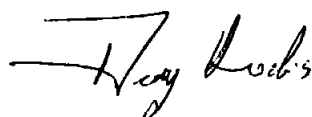
To my father, Miskuddin Loebis and my mother, Fatmawati Nasution, no words will ever say how grateful I am to have you as my parents.

Finally, I also want to express my gratitude to my loving wife, Elviera Soekadis, to my beloved daughter, Safira Alethia Loebis, who always stood by me and believed in me that every night before midnight I will be home.

AUTHOR'S DECLARATION

This is to certify that

- the thesis comprises only my original work towards the PhD,
- due acknowledgement has been made in the text to all other material used,
- during the candidature, I have not been registered for any other award at any other institution,
- this study was financed with the aid from the Engineering and Physical Sciences Research Council, UK,
- relevant scientific seminars and conferences were regularly attended at which work was often presented; external institutions were visited for consultation purposes and several papers prepared for publication,
- the thesis is 39,550 words in length, exclusive of tables, bibliographies and appendices.



04022005

Dedy Loebis

TABLE OF CONTENTS

LIST OF FIGURES xv

LIST OF TABLES xix

CHAPTER 1 INTRODUCTION 1

 1.1 BACKGROUND 1

 1.2 AIM AND OBJECTIVES OF THE THESIS 7

 1.3 AUTHOR'S CONTRIBUTION 7

 1.4 PUBLICATIONS 8

 1.5 THESIS ORGANISATION 9

CHAPTER 2 REVIEW OF MULTISENSOR DATA FUSION
TECHNIQUES AND THEIR APPLICATION TO AUTONOMOUS
UNDERWATER VEHICLE NAVIGATION 11

 2.1 INTRODUCTION 11

 2.2 AUTONOMOUS UNDERWATER VEHICLE NAVIGATION 13

 2.2.1 Dead Reckoning Navigation 13

 2.2.2 Radio Navigation 15

 2.2.3 Visual Navigation 16

 2.2.4 Acoustic Navigation 18

 2.2.5 Terrain-Relative Navigation 20

 2.3 MULTISENSOR DATA FUSION 22

 2.3.1 Benefits of Multisensor Data Fusion 23

 2.3.2 Problems and Issues 23

 2.3.3 Levels of Multisensor Data Fusion 24

 2.3.4 Multisensor Data Fusion Algorithms 25

 2.4 APPLICATIONS OF MULTISENSOR DATA FUSION FOR
AUTONOMOUS UNDERWATER VEHICLES 27

 2.4.1 GPS and Inertial-Based Systems 27

 2.4.2 Acoustic-Based Systems 29

 2.4.3 Acoustic- and Visual-Based Systems 31

 2.5 CONCLUDING REMARKS 33

TABLE OF CONTENTS — *Continued*

CHAPTER 3 THE <i>HAMMERHEAD</i> AUTONOMOUS UNDERWATER VEHICLE	34
3.1 THE EVOLUTION STRUCTURE	34
3.2 HARDWARE SETUP	37
3.3 DETAILS ON THE SENSORS	41
3.3.1 GPS	41
3.3.2 TCM2 Electronic Compass	43
3.3.3 Inertial Measurement Unit	45
3.3.4 Pressure Transducer	47
3.4 SYSTEM IDENTIFICATION FOR THE <i>HAMMERHEAD</i> AUV . .	48
3.4.1 System Identification	49
3.4.2 <i>Hammerhead</i> Trials Setup for System Identification	52
3.4.3 Identification Results	54
3.4.4 Modelling of Rudder-Yaw Channel	56
3.4.5 Model Validation	57
3.4.6 Model Analysis	59
3.5 CONCLUDING REMARKS	63
CHAPTER 4 FUZZY KALMAN FILTER MULTIOBJECTIVE GENETIC ALGORITHM: SIMULATION	64
4.1 INTRODUCTION	64
4.2 THE ADAPTIVE KALMAN FILTER ALGORITHM	65
4.3 FUZZY KALMAN FILTER	66
4.4 FUZZY LOGIC OBSERVER	69
4.5 FUZZY MEMBERSHIP FUNCTIONS OPTIMIZATION	70
4.6 FUSION OF INS SENSOR DATA	71
4.7 SIMULATION RESULTS	73
4.8 GPS/INS SURFACE (2D) NAVIGATION	78
4.9 CONCLUDING REMARKS	85
CHAPTER 5 FUZZY KALMAN FILTER MULTIOBJECTIVE GENETIC ALGORITHM: A PSEUDO REAL-TIME IMPLEMENTATION	86
5.1 INTRODUCTION	86
5.2 2D/SURFACE MISSION	87
5.2.1 Implementation Results	92

TABLE OF CONTENTS — *Continued*

5.3 3D/SURFACE-DEPTH MISSION	99
5.4 CONCLUDING REMARKS	118
CHAPTER 6 CONCLUSIONS AND FUTURE WORK	120
6.1 CONCLUSIONS	120
6.2 FUTURE WORK	122
APPENDIX A PUBLICATIONS	124
APPENDIX B THE <i>HAMMERHEAD</i> STATE SPACE MODEL PROPERTIES	198
B.1 SECOND ORDER	199
B.2 THIRD ORDER	200
B.3 FOURTH ORDER	201
B.4 FIFTH ORDER	202
B.5 SIXTH ORDER	203
B.6 SEVENTH ORDER	204
B.7 EIGHTH ORDER	205
B.8 NINTH ORDER	206
B.9 TENTH ORDER	207
APPENDIX C ANALOG TO DIGITAL CONVERTER FOR THE PRESSURE TRANSDUCER	209
APPENDIX D LINEAR AND EXTENDED KALMAN FILTER ALGORITHM	211
D.1 THE LINEAR KALMAN FILTER	211
D.2 THE EXTENDED KALMAN FILTER	212
APPENDIX E FUZZY LOGIC	214
E.1 BRIEF CHRONOLOGICAL RETROSPECTIVE	215
E.2 FUZZY SYSTEMS	215
E.3 FUZZIFICATION	218
E.4 FUZZY RULE BASE	219
E.5 INFERENCE ENGINE	221
E.6 DEFUZZIFICATION	225
E.7 FUZZY SYSTEMS TUNING AND OPTIMIZATION	225
APPENDIX F GENETIC ALGORITHM	227
F.1 SINGLE OBJECTIVE GENETIC ALGORITHMS	227

TABLE OF CONTENTS — Continued

F.2 MULTI-OBJECTIVE GENETIC ALGORITHMS	229
F.2.1 Pareto-Ranking	231
F.2.2 Trade-Off Sets Visualisation and Analysis	233
APPENDIX G CO-ORDINATE TRANSFORMATION	235
G.1 TRANSFORMATION FROM BODY TO NORTH-EAST-DOWN CO-ORDINATE FRAME	236
G.2 TRANSFORMATION FROM EARTH-CENTERED-EARTH-FIXED TO NORTH-EAST-DOWN CO-ORDINATE FRAME	240
REFERENCES	241

Nomenclature

δ	impulse function
δ_{r_k}	rudder deflection
Γ	model coefficients
\hat{y}	predicted model output
\hat{z}_k	estimated measurement
\hat{C}_{inn_k}	actual covariance value of innovation
Ω	a set of constraints on design parameters of optimization problem
A	system matrix of the state space model
B	matrix that relates the control input to the state vector
$F(.)$	matrix of nonlinear kinematic model
F_{lin_k}	discretised matrix of the linearised kinematic model
$F_{lin}(t)$	linearised matrix of the kinematic model
$H(.)$	measurement matrix of the state space model
H_{lin_k}	discretised measurement matrix of kinematic model
$H_{lin}(t)$	linearised measurement matrix of kinematic model
P_k	state error covariance
Q_k	process noise covariance
R_k	measurement error covariance
S_k	theoretical covariance value of innovation

ϕ	roll Euler angle
$\phi_{\varepsilon\varepsilon}$	autocorrelation of residuals
$\phi_{u_{in}\varepsilon}$	cross correlation of residuals and input respectively
ψ	yaw Euler angle
ψ_{im}	yaw angle from the fused heading sensor
\mathbf{delta}_k	discrepancy between actual and theoretical value of covariance \mathbf{R}_k
θ	pitch Euler angle
ε	residuals
c_k	weight of estimated sensor output
$d(t)$	plant external noise
F	vector of objective function of design parameters
f	scalar function of design parameters
i	number of fuzzy membership functions
Inn_k	innovation
J	performance index
j_0	first sample inside estimation window
J_{ze}	performance measure of fused sensor output
J_{zv}	performance measure of non-fused sensor output
k	discrete-time index
l	number of sensors
M	size of the moving estimation window
N	the number of data points in quadratic error function

n	number of samples
p	roll rate
q	pitch rate
t	time
u	surge
v	sway
$V(t)$	measurement white noise of the kinematic model
v_k	measurement white noise of the state space model
w	heave
$W(t)$	process white noise of the kinematic model
w_k	process white noise of the state space model
$X(t)$	states of the kinematic model
x_k	states of the state space model
X_{NED}	North position in the NED co-ordinate frame
y	actual output in quadratic error function
Y_{NED}	East position in the NED co-ordinate frame
z_k	measurement vector the state space model
Z_{NED}	Down position in the NED co-ordinate frame
za_k	actual value measurement
ACF	autocorrelation
ADC	analog to digital converter
AI	artificial intelligence

ARMAX	auto-regressive moving average with exogeneous input
ARX	auto-regressive with exogeneous input
AUV	autonomous underwater vehicle
CCF	crosscorrelation function
DGPS	differential global positioning system
DMT	deep mobile target
DTED	digital terrain elevation data
DVL	Doppler velocity log
DVS	Doppler velocity sonar
ECEF	Earth-centered Earth-fixed
EGNOS	Euro geostationary satellite augmentation system
EKF	extended Kalman filter
FEKF	fuzzy extended Kalman filter
FKF	fuzzy Kalman filter
FLO	fuzzy logic observer
GA	genetic algorithm
GPS	global positioning system
IAE	innovation adaptive estimation
IMU	inertial measurement unit
INS	inertial navigation system
IV	instrumental-variable
KF	Kalman filter

LS	least squares
LSI	laser stripe illumination
MLE	maximum-likelihood
MMAE	multiple model adaptive estimation
MOGA	multiobjective genetic algorithm
MSAS	multifunctional satellite augmentation system
MSDF	multisensor data fusion
NED	North-East-Down
PEM	prediction-error method
PRBS	pseudo random binary sequence
RIB	rigid inflated boat
ROV	remotely operated vehicle
SBL	short baseline
SI	system identification
SISO	single-input-single-output
SLAM	simultaneous localisation and mapping
TERCOM	terrain contour matching
UDRN	uniformly distributed random numbers
USBL	ultra short baseline
UTC	universal time coordinated
VNS	visual navigation system
VRU	vertical reference unit
WAAS	wide area augmentation system

LIST OF FIGURES

1.1	(a) ROV navigation and control system (b) AUV navigation and control system	1
1.2	AUV navigation and control system using multiple sensors	2
1.3	(a) The <i>Hammerhead</i> vehicle side view, (b) The <i>Hammerhead</i> vehicle front-top view	5
1.4	The <i>Hammerhead</i> laser viewing system experiments: (a) top view, (b) side view and (c) bottom view	6
2.1	<i>Jason</i> ROV explores ancient artifacts (courtesy of WHOI website www.whoi.edu)	12
2.2	Photomosaic acquired by <i>Jason</i> ROV(courtesy of MIT-WHOI website www.web.mit.edu/mit-who)	12
2.3	Error propagation in image chain as described by Huster <i>et al.</i> (1998) . .	17
2.4	(a) USBL system, (b) SBL system and (c) LBL system as described by Vickery (1998)	19
2.5	SLAM algorithm as described by Majumder <i>et al.</i> 2000a; Majumder <i>et al.</i> 2000b; Majumder 2001 <i>et al.</i>	21
2.6	GPS/INS navigation as described by McGhee <i>et al.</i> (1995)	28
2.7	MSDF in SLAM algorithm as described by Majumder <i>et al.</i> 2000a; Majumder <i>et al.</i> 2000b; Majumder 2001 <i>et al.</i>	32
3.1	The DMT deployed from a chute	34
3.2	(a) Rudder and rear hydroplanes, (b) front hydroplanes	35
3.3	(a) Camera port, (b) camera and laser scanning unit	36
3.4	(a) The schematic of the <i>Hammerhead</i> , (b) The <i>Hammerhead</i> on its trailer	36
3.5	(a) Rudder and rear hydroplanes motor, (b) front hydroplanes motor . .	38
3.6	(a) PCMCIA to 4×RS232/serial converter, (b) converter-laptop connection	39
3.7	(a) The <i>Hammerhead</i> hardware setup and sensor configuration, (b) full scale trial	40
3.8	(a) The wireless <i>Hammerhead</i> hardware setup and sensor configuration, (b) full scale trial	41
3.9	(a) Mounted GPS unit, (b) installed GPS antenna	41
3.10	(a) TCM2 circuitry, (b) mounted TCM2	43
3.11	(a) Watson Industries Ltd. inertial measurement unit, (b) mounted inertial measurement unit	45
3.12	(a) Pressure transducer, (b) ADC card	47
3.13	The overall system identification procedure.	51
3.14	(a) UDRN input, (b) 32 length PRBS input, (c) multistep input and (d) doublet input for system identification	53
3.15	(a) Original data set and (b) interpolated data set	54
3.16	(a) PRBS input and (b) PRBS heading	55
3.17	(a) UDRN input and (b) UDRN heading	55
3.18	(a) multistep input, (b) multistep heading	56
3.19	(a) Autocorrelation of residuals (b) cross correlation of residuals and the input	58

LIST OF FIGURES — Continued

3.20	Cross validation test for yaw-rudder channel using different sets of data whose fit between the simulated and measured outputs are respectively (a) 83.188%, (b) 70.887%, (c) 70.196% and (d) 51.915%	58
3.21	(a) Pole zero plot for the rudder-yaw channel model, (b) step response of the rudder-yaw channel model and (c) impulse response of the rudder-yaw channel model	60
3.22	m -circular motion of the vehicle, with $m=1,2,\dots$, as a response to a step input	61
3.23	Motion of the vehicle as a response to an impulse input	62
4.1	Membership function of (a) δ_k and (b) $\Delta \mathbf{R}_k$	68
4.2	Membership function of (a) $ \delta_k $ and (b) \mathbf{R}_k	69
4.3	Membership function and boundaries of \mathbf{R}_k	70
4.4	(a) Measured and estimated yaw output and error of sensor-1, (b) measured and recovered yaw output and error of sensor-1	74
4.5	Measured and estimated yaw output and error of sensor-2	75
4.6	Measured and estimated yaw output and error of sensor-3	75
4.7	(a) Measured and estimated yaw output and error of sensor 4, (b) measured and recovered yaw output and error of sensor-4	76
4.8	\mathbf{R} time series for sensor (a) 1, (b) 2, (c) 3 and (d) 4	77
4.9	Block diagram of GPS/INS using FKF and FEKF	79
4.10	(a) Longitude error, (b) latitude error	82
4.11	(a) Initial AUV trajectory using (a) standard EKF (b) FEKF	83
4.12	Final AUV trajectory using (a) standard EKF (b) FEKF	84
5.1	(a) Rudder input, (b) TCM2 electronic compass and IMU output	89
5.2	(a) Sensor-3 noise, (b) sensor-4 noise	90
5.3	Membership function of (a) δ_k and (b) $\Delta \mathbf{R}_k$ before optimization, and (c) $\Delta \mathbf{R}_k$ after optimization	91
5.4	Trade-off graphs for the FKF search	92
5.5	Measured and estimated yaw output and error of sensor-1	92
5.6	Measured and estimated yaw output and error of sensor-2	93
5.7	Measured and estimated yaw output and error of sensor-3	93
5.8	Measured and estimated yaw output and error of sensor-4	94
5.9	Measured and estimated yaw output and error of fused sensor	94
5.10	(a) AUV trajectory obtained using GPS, INS sensors (dead reckoning method) and GPS/INS using EKF without adaptation, (b) AUV trajectory obtained using GPS, INS sensors (dead reckoning method) and GPS/INS using EKF with adaptation	98
5.11	Surge	101
5.12	Sway	101
5.13	Heave	102
5.14	Pitch	103
5.15	Pitch rate	103
5.16	(a) Yaw sensor-1, (b) yaw rate sensor-1	108
5.17	True trajectory, GPS fixes and GPS/INS using yaw produced by sensor-1 only	108
5.18	(a) Initial true trajectory, GPS fixes and GPS/INS estimated trajectory using yaw produced by sensor-1 only, (b) final true trajectory, GPS fixes and GPS/INS estimated trajectory using yaw produced by sensor-1 only	109

LIST OF FIGURES — Continued

5.19	(a) Yaw sensor-2, (b) yaw rate sensor-2	110
5.20	True trajectory, GPS fixes and GPS/INS using yaw produced by sensor-2 only	110
5.21	(a) Initial true trajectory, GPS fixes and GPS/INS estimated trajectory using yaw produced by sensor-2 only, (b) final true trajectory, GPS fixes and GPS/INS estimated trajectory using yaw produced by sensor-2 only	111
5.22	(a) Yaw sensor-3, (b) yaw rate sensor-3	112
5.23	True trajectory, GPS fixes and GPS/INS using yaw produced by sensor-3 only	112
5.24	(a) Initial true trajectory, GPS fixes and GPS/INS estimated trajectory using yaw produced by sensor-3 only, (b) final true trajectory, GPS fixes and GPS/INS estimated trajectory using yaw produced by sensor-3 only	113
5.25	(a) Yaw sensor-4, (b) yaw rate sensor-4	114
5.26	True trajectory, GPS fixes and GPS/INS using yaw produced by sensor-4 only	114
5.27	(a) Initial true trajectory, GPS fixes and GPS/INS estimated trajectory using yaw produced by sensor-4 only, (b) final true trajectory, GPS fixes and GPS/INS estimated trajectory using yaw produced by sensor-4 only	115
5.28	(a) Yaw sensor fused, (b) yaw rate sensor fused	116
5.29	True trajectory, GPS fixes and GPS/INS using yaw produced by fused sensor	116
5.30	(a) Initial true trajectory, GPS fixes and GPS/INS estimated trajectory using yaw produced by fused sensor, (b) final true trajectory, GPS fixes and GPS/INS estimated trajectory using yaw produced by fused sensor	117
B.1	(a) 83.1888% fit between simulated and measured output, (b) autocorrelation residuals and cross correlation of residuals and the input, (c) step response, (d) impulse response and (e) pole zero plot	199
B.2	(a) 81.9476% fit between simulated and measured output, (b) autocorrelation residuals and cross correlation of residuals and the input, (c) step response, (d) impulse response and (e) pole zero plot	200
B.3	(a) 81.5526% fit between simulated and measured output, (b) autocorrelation residuals and cross correlation of residuals and the input, (c) step response, (d) impulse response and (e) pole zero plot	201
B.4	(a) 82.3896% fit between simulated and measured output, (b) autocorrelation residuals and cross correlation of residuals and the input, (c) step response, (d) impulse response and (e) pole zero plot	202
B.5	(a) 83.5926% fit between simulated and measured output, (b) autocorrelation residuals and cross correlation of residuals and the input, (c) step response, (d) impulse response and (e) pole zero plot	203
B.6	(a) 82.6636% fit between simulated and measured output, (b) autocorrelation residuals and cross correlation of residuals and the input, (c) step response, (d) impulse response and (e) pole zero plot	204
B.7	(a) 88.2953% fit between simulated and measured output, (b) autocorrelation residuals and cross correlation of residuals and the input, (c) step response and (d) pole zero plot	205

LIST OF FIGURES — Continued

B.8	(a) 83.2608% fit between simulated and measured output, (b) autocorrelation residuals and cross correlation of residuals and the input, (c) step response and (d) pole zero plot	206
B.9	(a) 83.0287% fit between simulated and measured output, (b) autocorrelation residuals and cross correlation of residuals and the input, (c) step response, (d) impulse response and (e) pole zero plot	207
C.1	ADC Diagram	209
C.2	PCB track layout	210
E.1	The fuzzy set ' <i>HIGH</i> '	218
E.2	The partition of the fuzzy variable <i>Velocity</i>	220
F.1	A general structure of a GA	228
F.2	A general structure of a MOGA	231
F.3	Pareto ranking with goal values	232
F.4	The parallel co-ordinate visualisation plot	233
G.1	AUVs navigation co-ordinate frames	235
G.2	(a) Angular velocities in body co-ordinate frame, (b) linear velocities in body co-ordinate frame , (c) angular rotation in NED co-ordinate frame and (d) linear translation in NED co-ordinate frame	237

LIST OF TABLES

3.1	The physical structure of the <i>Hammerhead</i> AUV	37
3.2	<i>GPGGA</i> sentence	42
3.3	TCM2-20 technical specification (courtesy of Precision Navigation website www.precisionnav.com)	43
3.4	TCM2 string	44
3.5	IMU technical specification	46
3.6	The degree of fit between simulated output and measured response of the identified <i>Hammerhead</i> AUV models	59
4.1	Fuzzy rule based FLO	69
4.2	FKF boundaries	71
4.3	MOGA parameters	72
4.4	Comparison of performance	78
5.1	MOGA parameters	87
5.2	MOGA parameters	91
5.3	Comparison of performance	95
D.1	Linear discrete Kalman filter equations	211
D.2	Discrete extended Kalman filter equations	213

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

The most basic function of an underwater vehicle navigation system is to estimate accurately its position and orientation. In many existing underwater vehicles, this is typically achieved by a remote or on-board computer that continuously collects data from sensors that are mounted inside the vehicle and processes them to render results according to a certain navigation algorithm. These results are subsequently fed to a human operator or to an automatic control system, which in turn produce an appropriate control action or control signal required to drive the vehicle in accordance with a predetermined mission scenario. A block diagram to represent this type of system is shown in Figure 1.1.

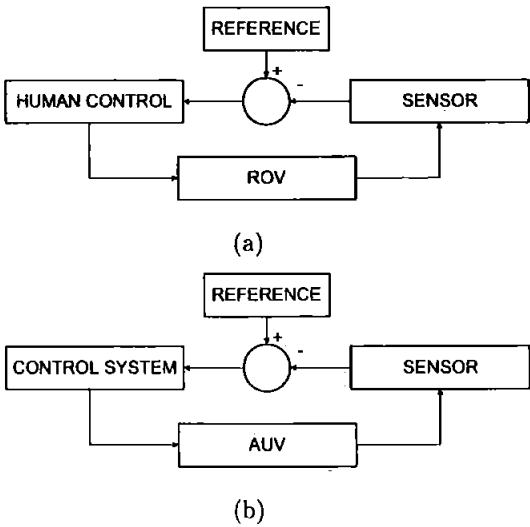


Figure 1.1: (a) ROV navigation and control system (b) AUV navigation and control system

The class of underwater vehicles controlled using the configuration represented in Figure 1.1(a) are known as remotely operated vehicles (ROVs) , while the ones represented in Figure 1.1(b) are known as autonomous underwater vehicles (AUVs). In both configurations, when an error occurs on the sensor, the input to the controller will also contain error and this subsequently results in an incorrect control action/signal. This problem raises the idea of using multiple sensors in the system (see Figure 1.2, for example). The implementation of multiple-sensor algorithms to provide an enhanced accuracy to an AUV (known as the *Hammerhead*) navigation system is the aim of this thesis, and consequently discussion henceforth is directed towards this topic.

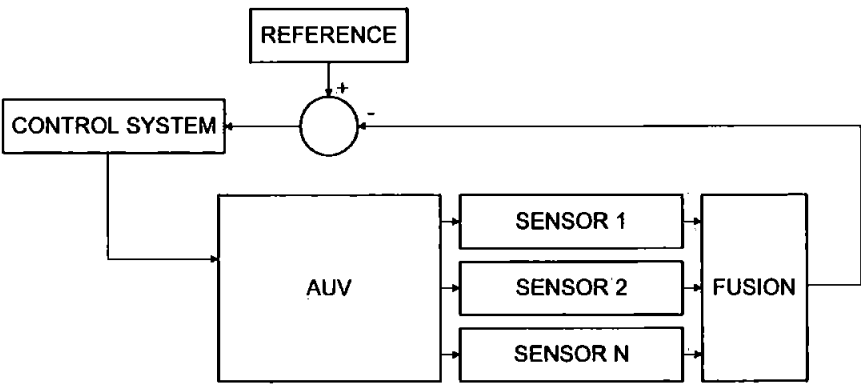


Figure 1.2: AUV navigation and control system using multiple sensors

Examples of navigation sensors in the *Hammerhead* AUV navigation system include a global positioning system (GPS) receiver, an electronic compass, a gyroscope and an accelerometer commonly assembled in an inertial measurement unit (IMU). Despite the fact that the purpose of the GPS is to provide the ability to compute location in 3D space, an AUV navigation system cannot, in general, continuously determine the vehicle's position using a GPS receiver alone. The main reason for this is that at times, the GPS position fixes are inaccurate, and for most underwater missions these are unavailable as the signals have only a limited water penetrating capability. The inaccuracies of a GPS receiver are caused by several factors. One of the most defining one is the requirement for the receiver to be able to lock onto signals from at least 4 different satellites for a period of time that is long enough to receive the information

encoded in the transmission.

Based on these facts, many AUV navigation systems utilise other navigation aids in conjunction with GPS fixes to enhance overall system performance. These aids usually include some combination or fusion of multiple sensors. The technique is known as multisensor data fusion (MSDF). Any sensors other than GPS that are used to position the vehicle are collectively referred to as a dead reckoning unit. The key component in this unit is an IMU and the navigation technique utilises this is known as an inertial navigation system (INS). An INS is a self-contained system that continuously measures AUV acceleration and angular rates, from which its velocity and position vectors are computed. However, an INS cannot be used alone to provide an accurate AUV navigation solution for indefinitely long periods of time because the error in a position estimate computed thereby can grow without bound and occasional measurements of absolute position with bounded errors are necessary. The errors that appear in a GPS and an INS are therefore complementary in nature. The INS smoothes out the short-term GPS errors, and GPS fixes calibrate or reset the INS drift over long time periods. Proper fusion of the GPS position fixes with the INS solution can take advantage of these complementary errors, producing a positioning performance that is better than could be obtained with either type of data alone.

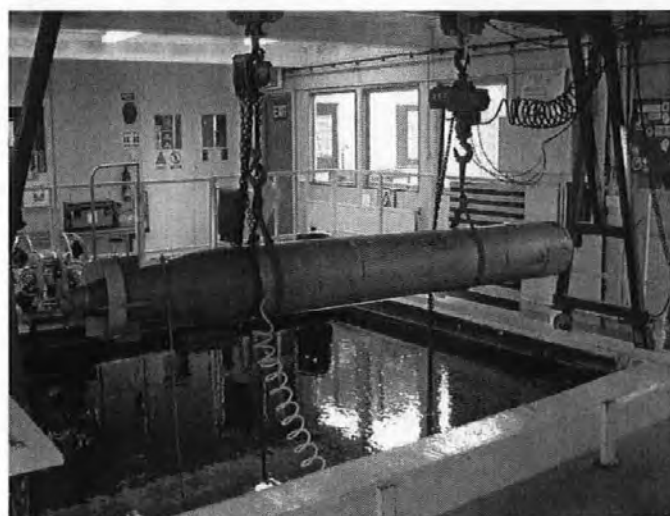
One of the most popular algorithms for combining data with complementary characteristics is the Kalman filter (Brown and Hwang, 1997; Grewal and Andrews, 2001; Grewal *et al.*, 2001). Introduced in 1960, Kalman filtering is a statistical technique developed to arrive at an estimate of the state of a system by combining a knowledge of system dynamics, represented as a state space model, with the statistical characteristics of system errors. The state estimate utilises a weighting function, called the Kalman gain, which is optimized to produce a minimum error variance. For this reason, the Kalman filter is called an optimal filter. In order for a Kalman filter to produce a statistically optimal estimate of its state, the filter's model equations, measurement equations and covariance matrices must exactly describe the actual dynamical and statistical properties of the AUV system of interest. In other words, the time-history of the system's state must be described *precisely* by known linear stochastic equations driven by white Gaussian noise with known statistical properties.

However, it is frequently the case that these are not available. It has been shown that the absence of this information can reduce the precision of the estimated filter states (Mehra, 1970; Mehra, 1971) or introduce biases to their estimates (Sangsuk-lam and Bullock, 1990). In addition, incorrect *a priori* information can lead to practical divergence of the filter (Chaer *et al.*, 1998).

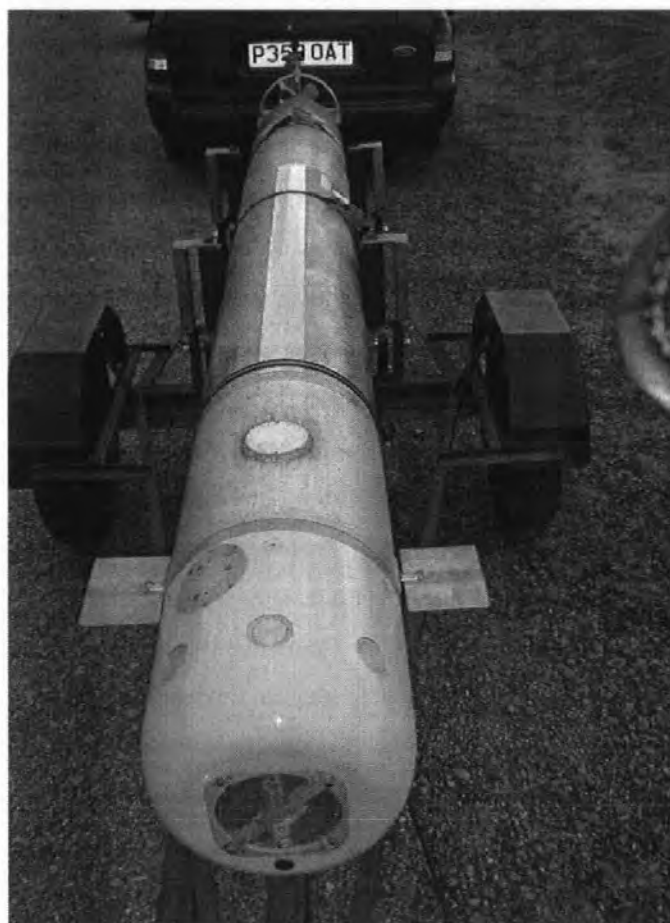
In GPS/INS applications, the estimation environment of the integrated system is non-stationary. In such an environment, imperfect *a priori* information will lead to the aforementioned problems. This implies that there is a major drawback in using a fixed Kalman filter designed by conventional methods. An adaptive filtering formulation, therefore tackles the problem of imperfect *a priori* information and may provide a significant improvement in performance over the fixed filter through a filter learning process.

In this thesis, the adaptation process is based on the detection of the dynamics of innovation sequences proposed by Mehra (1970) and Mehra (1971), coupled with fuzzy logic techniques. The fuzzy logic membership functions for the adaptation mechanisms are initially established by a combination of knowledge, experience and observation and therefore may not be optimal. Additionally, fine-tuning of its performance is still a matter of trial and error. Single objective and multiobjective genetic algorithm (MOGA) techniques are therefore used to optimize the parameters of the membership functions with respect to a certain performance criteria in order to improve the overall accuracy of the integrated navigation system.

In particular, the work proposed in this thesis is designed for use on pure simulated data and on navigational data gathered by the *Hammerhead* AUV developed and operated by the University of Plymouth and Cranfield University. *Hammerhead*, shown in Figure 1.3, was developed from a deep mobile target (DMT) torpedo of 3.5(m) length and 35(cm) diameter that was purchased by Cranfield University. Initial modifications were made to transform the torpedo into a PC controlled AUV (Naylies, 2000). Details of subsequent modifications on the vehicle are given in Chapter 3.



(a)



(b)

Figure 1.3: (a) The *Hammerhead* vehicle side view, (b) The *Hammerhead* vehicle front-top view

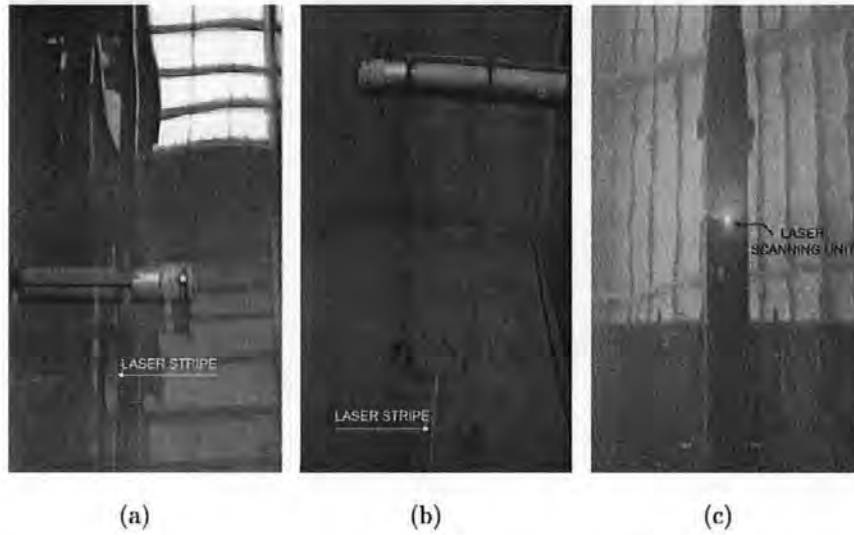


Figure 1.4: The *Hammerhead* laser viewing system experiments: (a) top view, (b) side view and (c) bottom view

As part of the collaboration, the University of Plymouth research team was also responsible for designing an automatic control system for the vehicle. Details of this work can be found in Naeem (2004). The work undertaken by Cranfield University research team involved developing a visual navigation system (VNS) based on a laser stripe illumination methodology developed previously there (Tetlow and Alwood, 1995), utilised to enhance the viewing below the vehicle as it gathers navigational data and underwater images for surveying purposes. Examples of the *Hammerhead* during an experiment to test the laser system are shown in Figure 1.4. Details of this work can be found in Dalglish (2004).

Individual work from each research team is put together to achieve the overall objectives of the project, *i.e.*, to design and develop an interactive navigation system consisting of the GPS/INS and VNS to interact with an appropriate control system. Interaction between the GPS/INS and VNS is driven by the requirement to have an underwater absolute positioning system to minimise the drift inherent in the INS solution. This allows the vehicle to operate underwater without the need for frequent excursions to the surface to obtain GPS fixes. The navigation solution from the GPS/INS and VNS integrated system is constantly passed to the control system

which in turn drives the vehicle reaching its predetermined set points more accurately than it does using a single sensor alone.

1.2 AIM AND OBJECTIVES OF THE THESIS

The main aim of this research is to enhance the performance of an AUV GPS/INS integrated system by a combination of Kalman filtering and artificial intelligence techniques.

To achieve the above aim, the following objectives were defined:

1. To investigate AUV navigation systems, in general, and GPS/INS integrated systems, in particular, and to critically review the associated MSDF techniques used therein, with a particular attention given to Kalman filtering.
2. To acquire a dynamic model of the *Hammerhead* AUV using a set of relatively low cost sensors prior to developing a Kalman filtering algorithm thereof. This was accomplished through implementing system identification techniques to data sets obtained from a series of full scale trials.
3. To develop a novel method employing a synergistic use of soft computing techniques in overcoming the drawbacks of utilising fixed Kalman filtering.
4. To analyse the performance of the developed adaptive Kalman filtering against fixed Kalman filter in both simulation and pseudo real-time environment.

1.3 AUTHOR'S CONTRIBUTION

This thesis introduces a novel alternative to the widely used fixed Kalman filter for application in an AUV GPS/INS integrated system. The proposed method is built upon a synergistic combination between soft computing and Kalman filtering techniques. The novelty factor originates from the use of MOGA approaches to optimize the membership functions of fuzzy inference systems which are used to adjust the values of *a priori* statistical information of the filter to cope with the changes in

the estimation environment. This is known to be the first method used and is thus considered as a major contribution in relation to AUV technology.

The analysis of this method is carried out thoroughly and implemented to different AUV mission scenarios to show the effectiveness and suitability of the adaptive techniques. It is shown that the proposed techniques are not only able to relax the requirements to have a good *a priori* statistical information, but also able to outperform the fixed filter in both simulation and pseudo real-time environment, and consequently has the potential in enhancing the performance of an AUV navigation system. It is important to note that although the analysis techniques in this thesis are developed for AUVs, they can effortlessly be applied to other autonomous vehicles, which are employed in the aerospace, underground and land environments. Thus the adaptive Kalman filtering algorithms proposed herein will be valuable as a generic method for all types of navigation system designs.

1.4 PUBLICATIONS

To date the following papers have been published or accepted as a direct result of this research programme and can be found in Appendix A:

1. Loebis, D., Naeem, W., Sutton, R. and Chudley, J. (2004). The Navigation, Guidance and Control of the *Hammerhead* Autonomous Underwater Vehicle, (To appear in: *Advances in Unmanned Marine Vehicle* (Roberts, G. N. and Sutton, R. (Ed)). Peter Peregrinus Ltd., Herts.)
2. Loebis, D., Sutton, R., Chudley, J. and Naeem, W. (2004). Adaptive Tuning of a Kalman Filter via Fuzzy Logic for an Intelligent AUV Navigation System. *Control Engineering Practice*, **12** (12), pp. 1531-1539.
3. Loebis, D., Sutton, R. and Chudley, J. (2004). A Fuzzy Kalman Filter Optimized Using a Multiobjective Algorithm for Enhanced Autonomous Underwater Vehicle Navigation. *Proceedings of the Institution of Mechanical Engineers Part M*, **218** (M1), pp. 53-69.

4. Loebis, D., Sutton, R. and Chudley, J. (2004). A Soft Computing Method for an AUV Navigation System with Pseudo-Real-Time Applicability. *Proc. 2004 IFAC Conference on Control Applications in Marine Systems*, Ancona, Italy, pp. 421-426.
5. Loebis, D., Sutton, R., Chudley, J., Dalglish, F. R. and Tetlow, S. (2004). The Application of Soft Computing Techniques to an Integrated Navigation System of an AUV. *Proc. 5th IFAC Symposium on Intelligent Autonomous Vehicles*, Lisbon, Portugal, MA-3-2 (CD-ROM Preprints)
6. Loebis, D., Chudley, J. and Sutton, R. (2003). A Fuzzy Kalman Filter Optimized Using a Genetic Algorithm for Accurate Navigation of an Autonomous Underwater Vehicle. *Proc. 6th IFAC Conference on Manoeuvring and Control of Marine Craft*, Girona, Spain, pp. 19-24.
7. Loebis, D., Dalglish, F. R., Sutton, R., Tetlow, S., Chudley, J., and Alwood, R. L. (2003). An Integrated Approach in the Design of a Navigation System for an AUV. *Proc. 6th IFAC Conference on Manoeuvring and Control of Marine Craft*, Girona, Spain, pp. 329-334.
8. Loebis, D., Chudley, J. and Sutton, R. (2003). A Fuzzy Kalman Filter for Accurate Navigation of an Autonomous Underwater Vehicle. *A Proceedings Volume from the IFAC Workshop on Guidance and Control of Underwater Vehicles (ISBN: 0080442021)*, Newport, South Wales, UK, pp. 157-162.
9. Loebis, D., Sutton, R. and Chudley, J. (2002). Review of Multisensor Data Fusion Techniques and Their Application to Autonomous Underwater Vehicle Navigation. *Journal of Marine Engineering and Technology*, A1, pp. 3-14. (This was given the Stanley Gray Award for the most worthy Offshore Technology paper in the journal during 2001/2002).

1.5 THESIS ORGANISATION

Chapter 2 discusses previous work and recent developments in AUV navigation and introduces MSDF techniques as a means of improving AUV navigation capability.

Problems and issues of the techniques are briefly discussed. The levels of representation in fusion processes are identified herein. The methods of MSDF are presented, followed by examples of their application in AUV navigation systems with special attention being given to Kalman filtering techniques.

Chapter 3 introduces the *Hammerhead* vehicle in more detail. The evolution of the vehicle's hardware from its early stage as a DMT to a fully autonomous vehicle is presented. The sensors and their technical specifications, including the necessary electronics work undertaken to interface the sensors with a CPU are discussed. These are then associated with the system identification full scale trials discussed thereafter. Results and analysis of the identification are also presented.

Chapter 4 supplies the theoretical background of the adaptive Kalman filtering techniques followed by their applications to a set of simulated 2D/surface GPS/INS data. The proposed method is first applied to fuse data coming from different INS measurements. The results are subsequently used synergistically with other measurements to obtain a GPS/INS integrated navigation solution. A performance comparison between the fixed and the proposed Kalman filters are made. It will clearly be seen that the proposed method can significantly improve the performance of the *Hammerhead* navigation system.

Chapter 5 extends the work in Chapter 4 by implementing the adaptation mechanisms to a set of GPS/INS real data obtained from *Hammerhead* full scale trials. Both GPS/INS applications for 2D/surface and 3D/surface-depth mission scenarios are considered.

Chapter 6 provides conclusions and recommendations for future work.

Additionally, support of some of the above chapters are appendices.

CHAPTER 2

REVIEW OF MULTISENSOR DATA FUSION TECHNIQUES AND THEIR APPLICATION TO AUTONOMOUS UNDERWATER VEHICLE NAVIGATION

2.1 INTRODUCTION

The oceans cover 70 per cent of the Earth's surface and contain an abundance of living and non-living resources that remain largely untapped and waiting to be discovered. However, a number of complex issues, mainly caused by the nature of underwater environments, make exploration and protection of these resources difficult to perform. In the past few decades, various worldwide research and development activities in underwater robotic systems have increased in order to meet this challenge. One class of these systems is tethered and remotely operated and referred to as ROVs. Figure 2.1 shows an example of ROVs with a photomosaic obtained by the vehicle from a particular mission shown in Figure 2.2. The ROVs serve a range of military, scientific and commercial needs. The tether is used to send power and control signals and to receive data from the on-board sensors. However, as depth or speed increases, the drag of the tether becomes more significant and more effort is required from the operator to control the vehicle. This, if must be done for a long period of time, may degrade the ability of the operator to control the vehicle accurately. The demand for a more sophisticated underwater robotic technology that eliminates the need for human operator and therefore capable of operating autonomously becomes apparent. These requirements lead to the development of AUVs.

To achieve truly autonomous behaviour, an AUV must be able to navigate accurately within an area of operation. In order to achieve this, an AUV needs to employ a navigation sensor with a high level of accuracy and reliability. However, in practice,



Figure 2.1: *Jason* ROV explores ancient artifacts (courtesy of WHOI website www.whoi.edu)

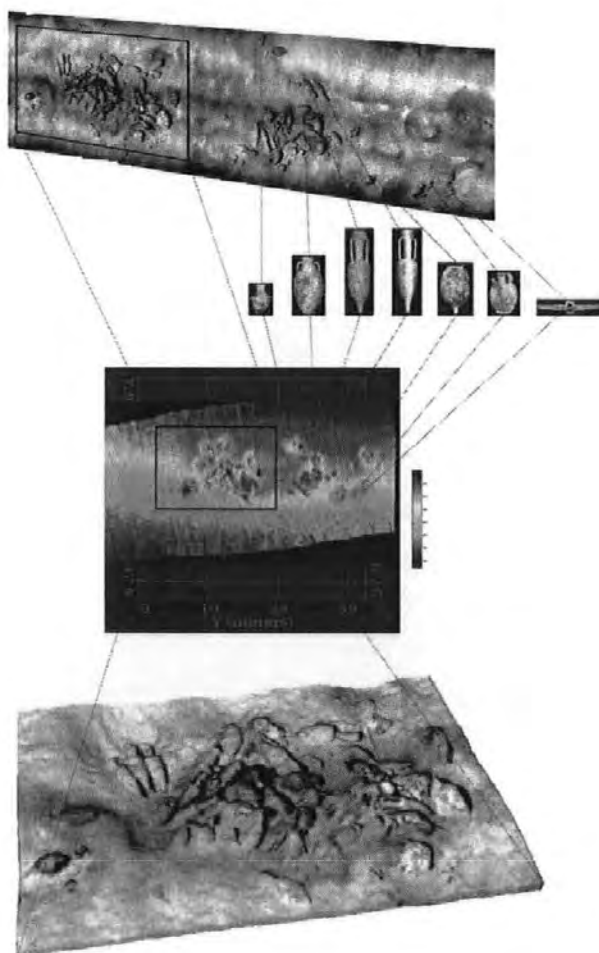


Figure 2.2: Photomosaic acquired by *Jason* ROV (courtesy of MIT-WHOI website www.web.mit.edu/mit-who)

as will be discussed in the next section, a single sensor alone may not be sufficient to provide an accurate and reliable navigation system, as it can only operate efficiently under certain conditions or it has inherent limitations when operating in underwater environments. It is therefore necessary to use a number of sensors and combine their information to provide the necessary navigation capability. To achieve this, a MSDF approach, which combines data from multiple sensors and related information from associated databases, can be used.

The aim of this chapter is to survey previous work and recent development in AUV navigation and to introduce MSDF techniques as a means of improving AUVs navigation capability. The majority of the material presented in this chapter being based up on Loebis *et al.* (2002), which was given the Stanley Gray Award by The Institute of Marine Engineering, Science and Technology for the most worthy Offshore Technology paper in the Journal of Marine Engineering and Technology during 2001/2002. The structure of this chapter is as follows: Section 2.2 describes the navigation systems that are currently being used in AUVs. MSDF is discussed in Section 2.3, whilst MSDF using specific sensor combinations applied to the navigation of AUVs are given in Section 2.4. Finally, concluding remarks are made in Section 2.5.

2.2 AUTONOMOUS UNDERWATER VEHICLE NAVIGATION

Navigation systems used by AUVs that are discussed here include dead reckoning, radio, optical, acoustic and terrain-relative navigation.

2.2.1 Dead Reckoning Navigation

Dead reckoning is a mathematical means to determine position estimates when the vehicle starts from a known point and moves at known velocities. The present position is equal to the time integral of the velocity. Measurement of the vector velocity components of the vehicle is usually accomplished with a compass (to obtain direction) and a water speed sensor (to obtain magnitude). The principal problem is that the presence of an ocean current can add a velocity component to the vehicle, which

is not detected by the speed sensor.

An INS is a dead reckoning technique that obtains position estimates by integrating the signal from an accelerometer, which measures the vehicle's acceleration. The vehicle position is obtained in principle by double integration of the acceleration. The orientation of the accelerometer is governed by means of a gyroscope, which maintains either a fixed or turning position as prescribed by some steering function. The orientation may also in principle be determined by integration of the angular rates of the gyroscope. Both the accelerometer and the gyroscope depend on inertia for their operation

A dead reckoning navigation system is attractive mainly because it uses sensors that are self-contained and able to provide fast dynamic measurements. Unfortunately in practice, this integration leads to unbounded growth in position error with time due to the noise associated with the measurement and the nonlinearity (which takes form in bias and drift as the result of temperature change or external vibration (Titterton, 1997)) of the sensors, and there is no built-in method for reducing this error. Depending on the sensors used and the specific vehicle mission, the navigational error can grow rapidly to the point where either the mission will not produce useful data or it will not be achievable at all.

Two types of dead reckoning sensors have been widely employed in AUVs: IMUs and Doppler velocity sonar (DVS). Many very accurate IMUs have been developed for submarines. However, these are typically very expensive devices and are used only in naval vehicles. Lower cost IMUs have been used in AUVs (Cox and Wei, 1995). However, due to the low acceleration encountered in autonomous underwater vehicles, these units are not normally of sufficient accuracy to provide stand-alone navigation.

DVS sensors provide measurement of a velocity vector with respect to the sea floor. These sensors normally comprise of three or more separate sound beams allowing construction of a full three-dimensional velocity vector. Typically, these instruments have specifications of about one per cent of the distance travelled (Bellingham, 1992). However, these results can only be achieved when the speed of sound in the AUV's

area of operation does not vary significantly as a result of changes in the salinity, temperature and density of the water. Therefore, as in the IMU case, these units are not normally used to provide stand-alone navigation.

2.2.2 Radio Navigation

Radio navigation systems mainly use the GPS (Ellowitz, 1992). The GPS is a satellite-based navigational system that provides the most accurate open ocean navigation available. GPS consists of a constellation of 24 satellites that orbit the Earth in 12 hours. There are six orbital planes (with nominally four satellites in each) equally spaced (60 degrees apart) and inclined at about 55 degrees with respect to the equatorial plane (Ellowitz, 1992). This constellation provides the user with between five and eight satellites visible from any point on the Earth. Improvement on the accuracy of ordinary GPS can be achieved using differential GPS (DGPS) techniques. The idea behind all differential positioning is to correct bias errors at one location with measured bias errors at a known position. A reference receiver, or base station, computes corrections for each satellite signal (Dana, 2000). There are several kinds of DGPS available; DGPS mode using a beacon receiver and DGPS mode using geostationary satellites. Wide area augmentation system (WAAS) in North America and its counterpart, Euro geostationary satellite augmentation system (EGNOS) in Europe and multi-functional satellite augmentation system (MSAS) in Asia fall into the second category of DGPS. It is worth noting that under ideal conditions, the accuracy of an ordinary GPS is typically $15(m)$, while the accuracy of DGPS using a beacon receiver and geostationary satellites are $3 - 5(m)$ and less than $3(m)$ respectively. A detailed discussion on the GPS is out of the scope of this thesis and interested readers can refer to Yeazel (2003).

The GPS-based navigation system is used extensively in surface vessels as these vehicles can receive signals directly radiated by the GPS. Unfortunately, these signals have a limited water penetrating capability. Therefore to receive the signals, an antenna associated with an AUV employing a GPS system must be clear and free of water. There are three possible antenna configurations to meet this requirement.

These are fixed, retractable, or expendable antennas (Kwak *et al.*, 1992). A fixed antenna is a non-moving antenna placed on the outside of the AUV. The AUV has to surface to expose this antenna and stay surfaced until the required information has been received and processed adequately. A retractable antenna is one that the AUV would deploy while still submerged. When the required information is received, the antenna is retracted back to the AUV. The expendable antenna works along the same principle as the retractable antenna except that it is used once and discarded. When required, another antenna would be deployed.

These antenna configurations require the AUV either to surface or to rise to a shallow depth, but there are several disadvantages (Kwak *et al.*, 1993). For an AUV to receive radio signals, it must interrupt its mission, expend time and energy climbing and/or surfacing, risk its safety for up to a minute on the surface or in a shallow depth of water getting the fix, which is especially dangerous in a hostile environment, then expend additional time and energy submerging to resume the mission. Even if an extremely accurate fix is obtained, the vehicle location uncertainty can grow significantly during descent before mission is ever resumed. Therefore there is a need to combine information obtained by a GPS navigation system with other underwater navigation sensors when the AUV operates underwater to maintain good navigation capability.

2.2.3 Visual Navigation

In the context of visual imaging for navigation, the underwater environment is a very special place. The reason for this is that in addition to visual-sensing issues that must be addressed in land and space-based vehicles, there are also issues specific to underwater imaging. These issues include limited range of visibility, brightness and contrast variation, and nonuniform illumination (Marks *et al.*, 1994). Limited range of visibility is caused by the attenuation of light in water by absorption and scattering by suspended matter. Light absorption and scattering cause the amount of reflected light to decay exponentially as a function of distance to scene surfaces. The absorption and scattering of light also affect image brightness and contrast. Objects

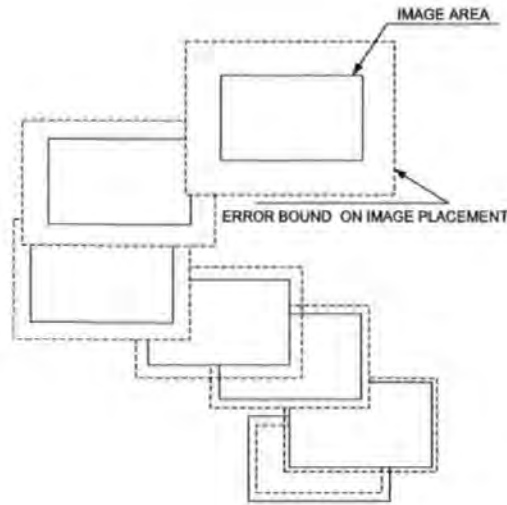


Figure 2.3: Error propagation in image chain as described by Huster *et al.*(1998)

far away appear dark; as they move nearer, their brightness and contrast increase. Changes in image intensity brightness and contrast can cause many image processing techniques to fail. If some type of intensity normalisation is not performed, brightness and contrast differences between images make it difficult to realise that the same scenery or object is being viewed (Marks *et al.*, 1994). Nonuniform illumination refers to the limitation of artificial light sources to provide uniform illumination of the entire scene under observation. A classic example that demonstrates the difficulties nonuniform lighting can cause is the imaging of a planar, perpendicular surface using a collocated camera/light source. In this case, the image centre will appear brighter than the image border. If the camera and light source are moved relative to the scene, both the absolute and relative brightness of each pixel in the image will change. Simple effects such as these can degrade correspondence (image matching) performance; more complicated effects such as shadowing can cause significant difficulties for most image correspondence techniques (Marks *et al.*, 1994).

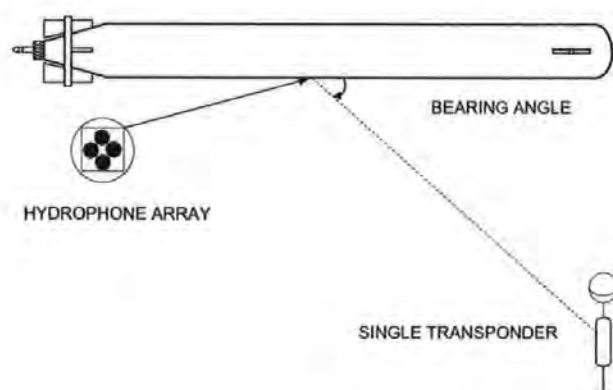
VNS involves the estimation of 3D motion from time varying imagery (Victor and Sentieiro, 1994; Hallset, 1992). Most techniques for this purpose require knowledge of relevant 2D geometric information in an image sequence. The current state-of-the-art in VNS is essentially a form of dead reckoning. This method works by creating a

mosaic where a series of images are taken from a video-stream and aligned with each other to form a chain of images along the vehicle path. When a new image is about to be added to a mosaic, it must be properly aligned with the last image in the chain of images comprising the mosaic. To accomplish this, the two images are compared, and the displacement vector between the two image centres is calculated. Therefore, to determine the current vehicle position, it would be possible to compute the total distance travelled by summing the image displacement measurements along the image chain (Huster *et al.*, 1998). As with the INS discussed in Section 2.2.1, this method has a fundamental problem: the unbounded propagation of errors on vehicle position over time. This random walk-effect is due to the accumulation of image alignment errors as the length of mosaic increases (Figure 2.3). Therefore, as in the INS case, this navigation method is not normally used to provide stand-alone navigation.

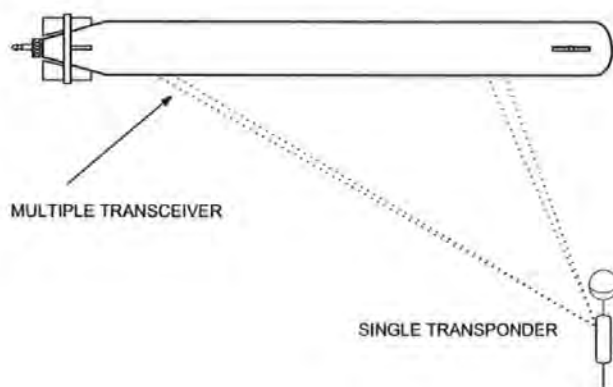
2.2.4 Acoustic Navigation

Acoustic navigation is the most widely accepted form of AUV navigation, and a variety of systems have been both researched and tested. Most require an engineered environment, meaning that something has been added to the environment to aid navigation. The distance between acoustic baselines is generally used to define an acoustic positioning system, that is the distance between the active sensing elements. Three types of system have been primarily employed; ultra short baseline (USBL), short baseline (SBL) and long baseline (LBL) with distance between acoustic baselines less than 10 cm, between 20 to 50 metres and between 100 to 6,000 metres respectively (Vickery, 1998).

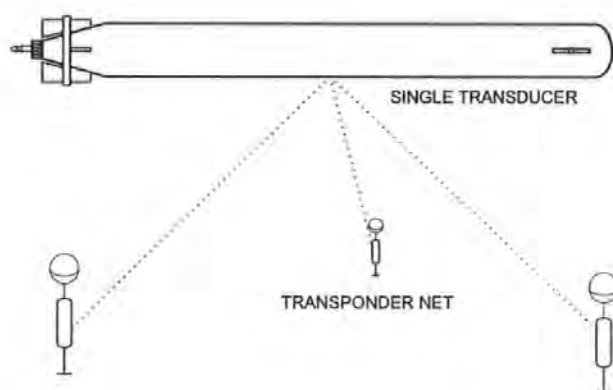
USBL systems (Figure 2.4(a)) employ a single beacon on the bottom of the seafloor which emits acoustic pulses without being interrogated from an AUV. The on-board AUV equipment consists of a two-dimensional hydrophone array mounted on the bottom of the AUV. USBL systems measure the time or phase difference of the arrival of an acoustic pulse between individual elements of the hydrophones. This time or phase difference is used to determine the bearing from the USBL transceiver to the beacon. If a time-of-flight interrogation technique is used, a range to that beacon



(a)



(b)



(c)

Figure 2.4: (a) USBL system, (b) SBL system and (c) LBL system as described by Vickery (1998)

will also be available from the USBL system. In SBL (Figure 2.4(c)) three or more transceivers are rigidly mounted on the hull of the AUV, making either an equilateral or a right-angled triangle. The distance between each transceiver is precisely known. A bearing to the transponder is derived from the detection of the relative time-of-arrival as an acoustic pulse passes each of the transceivers. If the time-of-flight interrogation technique is used, a range to that beacon will also be available from the SBL system. Any range and bearing position derived from USBL and SBL systems are with respect to the transceivers mounted on the AUV and as such the systems need a vertical reference unit (VRU), a gyroscope, and possibly a surface navigation system to provide a position that is seafloor (Earth) referenced (Vickery, 1998).

In LBL navigation systems (Figure 2.4(c)), an array of acoustic beacons separated by a range of 100 metres to a few kilometres is deployed on the seabed (Vickery, 1998; Geyer *et al.*, 1987). The vehicle determines its position by listening to the pulses emitted from the beacons and recording the arrival times. The location of these beacons must be provided, and the vehicle must be able to detect and distinguish between their signals. The two major types of LBL navigation are described as spherical and hyperbolic. In spherical navigation, the vehicle interrogates the array by emitting its own pulse and then listens for the responses from the beacons. In hyperbolic LBL navigation, the vehicle does not interrogate the array, but instead listens passively to the synchronised pulses emitted by the beacons (Bellingham *et al.*, 1992). Any range/range position derived from a LBL system is with respect to relative or absolute seafloor co-ordinates. As such a LBL system does not require a VRU or gyroscope (Vickery, 1998).

2.2.5 Terrain-Relative Navigation

For some applications of AUVs, the use of acoustic beacons is undesirable or impractical. In particular, the acoustic beacons must be pre-deployed for every mission and the vehicles can operate only over relatively short ranges, and they are far too expensive to be practical in low cost civilian AUV work. Also the accuracy of the acoustic signals tend to degrade due to noise and reverberation problem. This then motivates

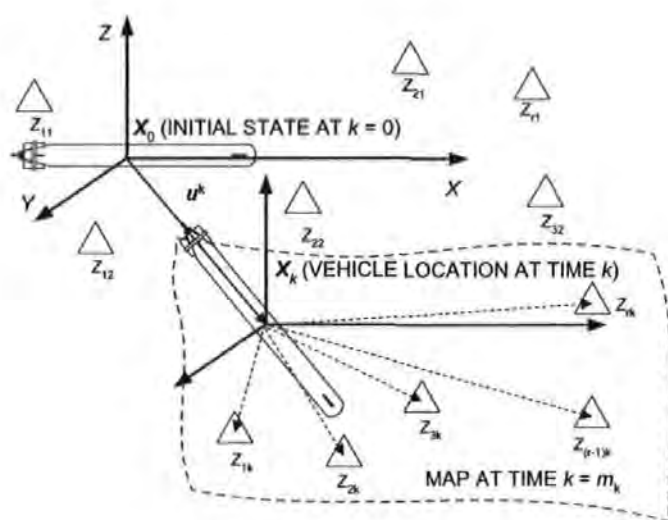


Figure 2.5: SLAM algorithm as described by Majumder *et al.* 2000a; Majumder *et al.* 2000b; Majumder 2001 *et al.*

the use of on-board terrain sensors for the purpose of navigation of an AUV. An on-board sensor is used to obtain information on the terrain surrounding the vehicle in the form of features or landmarks. The vehicle maintains a map of these landmarks which may or may not have been provided *a priori*.

As the vehicle moves through the environment the landmark observations obtained from the terrain sensor are matched to the landmarks maintained in the map and used, in much the same way as beacon observations, to correct and update the estimated location of the vehicle. In underwater environments it is very rare that an *a priori* terrain map will exist. Unlike surface applications, satellite or aircraft imagery cannot be used to build an underwater terrain map. This then precludes the common use of digital terrain elevation data (DTED) as employed by systems such as terrain contour matching (TERCOM) used for cruise missiles (Golden, 1980). This limitation then motivates the development of simultaneous localisation and mapping (SLAM) for AUV navigation (see Figure 2.5), which is the process of concurrently building a feature-based map of the environment and using this map to obtain estimates of the location of the vehicle. In essence, the vehicle relies heavily on its ability to extract

useful navigation information from the data returned by its sensors. The vehicle typically starts at an unknown location with no *a priori* knowledge of landmark locations. From relative observations of landmarks, it simultaneously computes an estimate of vehicle location and an estimate of landmark locations. While continuing in motion, the vehicle builds a complete map of the landmarks and uses these to provide continuous estimates of the vehicle location. By tracking the relative position between the vehicle and identifiable features in the environment, both the position of the vehicle and the position of the features can be estimated simultaneously. In Figure 2.5, the relationship between the vehicle, features and map at any time k is shown above. A Cartesian axes system is used to describe the vehicle location at any time k denoted by x_k . The vehicle states change as a result of the applied control input u_k . The map at any time k is defined as set of landmarks or features detected from the sensor observation z_k relative to the vehicle location. The SLAM algorithm has recently seen a considerable amount of interest from the AUV community as a tool to enable fully autonomous navigation (Majumder *et al.*, 2000a; Majumder *et al.*, 2000b; Majumder *et al.*, 2001).

2.3 MULTISENSOR DATA FUSION

It is clear from the previous discussion that information from sensors used in one navigation system need to be combined or fused with information from sensors of other navigation systems to improve the overall accuracy of the system. To achieve this, MSDF techniques, which combine data from multiple sensors and related information from associated databases can be used (Llinas and Waltz, 1990; Hall, 1992). Varshney (1997) describes MSDF as the acquisition, processing and synergistic combination of information gathered by various knowledge sources and sensors to provide a better understanding of a phenomena. In this section, a general introduction to MSDF is provided. A description of the benefits of MSDF, problems and issues, levels of MSDF where fusion takes place and MSDF algorithms are presented.

2.3.1 Benefits of Multisensor Data Fusion

In general, fusion of multisensor data provides significant advantages over single source data. The advantages can be summarised as follows (Varshney, 1997; Harris *et al.*, 1998):

1. *Improved system reliability and robustness.* Multiple sensors have inherent redundancy. Due to the availability of data from multiple sensors uncertainty can be reduced, noise can be rejected and sensor failure can be tolerated.
2. *Extended coverage.* An increase in both spatial and temporal coverage of an observation is made possible by the use of multiple sensor systems. Multiple sensors can observe a region larger than the one observable by a single sensor.
3. *Increased confidence.* Joint data from multiple sensors confirm the set of hypotheses about an object or event. The confirmation can be used to exclude some hypotheses to produce a reduced set of feasible options and as a result reduce the effort required to search for the best solution.
4. *Enhanced resolution.* Multiple sensors with different resolution can result in a greater resolution than a single sensor can achieve.

2.3.2 Problems and Issues

A technique for MSDF should consider several key issues, summarised below (Harris *et al.*, 1998; Hall and Llinas, 1997):

1. *Registration/data alignment.* Each sensor provides data in its local frame. The data from different sensors must be converted into a common reference frame before combination. This problem of aligning sensor reference frames is often referred to as a registration problem.
2. *Correspondence/data association.* Once the sensors are registered, there is still a need to establish which data features in one sensor refer to the same aspect environment of the sensor.

3. *Fusion.* The fusion of data from multiple sensors or a single sensor over time can take place at different levels of representation. A useful categorisation is to consider MSDF as taking place at signal, pixel, feature and symbol levels of representation.
4. *Inference and estimation.* Once the data has been fused, it is necessary to infer the sensed data due to the inherent uncertainty in the combined measurements.
5. *Sensor Management.* Sensor management can take the form of active data gathering where the sensors are directed via feedback to specific fusion stage, physical reconfiguration of the spatial pattern of the sensors and sensor type, or algorithmic changes to the combination of data.

2.3.3 Levels of Multisensor Data Fusion

The common fused representation may range from a low-level probability distribution for statistical inference to high level logical proposition used in production rules for logical inference. Luo and Kay (1990) and Luo *et al.* (2002) divide the levels of representation of MSDF into signal, pixel, feature and symbol levels.

1. *Signal-level.* Signal level fusion deals with the combination of signals from a group of similar sensors with the aim of deriving a single composite signal, usually of the same form as the original signals but with a higher quality. The signals produced by the sensors can be modelled as random variables corrupted by uncorrelated noise, with the fusion process considered as an estimation procedure. A high degree of spatial and temporal registration between the sensed data is necessary for fusion to take place.
2. *Pixel-level.* Pixel level fusion deals with the combination of multiple images into a single image with a greater information content. The fused images can be modelled as a realisation of a stochastic process across the image, with the fusion process considered as an estimation procedure. In order for pixel-level to be feasible, the data provided by each sensor must be able to be registered at the pixel level and, in most cases, must be sufficiently similar in terms of its resolution and information content.

3. *Feature-level.* Feature level fusion deals with the combination of features derived from signals and images into meaningful internal representations or more reliable features. A feature provides for data abstraction and is created either through the attachment of some type of semantic meaning to the results of the processing of some spatial and/or temporal segment of the sensory data or through a combination of existing features. As compared to the signal and pixel-level fusion, the sensor registration requirements for feature-level fusion are less stringent, with the result that the sensors can be distributed across different platform.
4. *Symbol-level.* Symbol level fusion deals with the combination of symbols with an associated uncertainty measure, each representing some decision, into symbols representing composite decisions. A symbol derived from sensory information represents a decision that has been made concerning some aspect of the environment. The decision is usually made by matching features derived from the sensory information to a model. The Sensor registration is usually not explicitly considered in symbol-level fusion because the spatial and temporal extent of the sensory information upon which a symbol is based has already been explicitly considered in the generation of the symbol.

2.3.4 Multisensor Data Fusion Algorithms

This section presents algorithms for MSDF. Luo *et al.* (2002) classify MSDF algorithms as follows: estimation methods, classification methods, inference methods and artificial intelligence methods. Each of these methods will be discussed here and applications to AUV navigation are presented in Section 2.4.

1. *Estimation methods.* A general estimation method of fusion is to take a weighted average of redundant information provided by a group of sensors and use this as the fused value. While this method provides real-time processing capability of dynamic low-level data, the Kalman filter (KF) is generally preferred as it provides a method that is nearly equal in processing requirement and results in estimates for the fused data that are optimal in a statistical sense. Kalman

filtering is an estimation method that combines all available measurement data, plus prior knowledge about the system and measuring devices, to produce an estimate of the state in such a manner as to minimise the error statistically (Brown and Hwang, 1997). A detailed formulation of the Kalman filter is given Appendix D.

2. *Classification methods.* Classification methods involve partitioning of the multidimensional feature space (by geometrical or statistical boundaries) into distinct regions, each representing an identity class. In this method, the location of a feature vector to prespecified locations in feature space is compared. A similarity measure must be computed and each observation is compared to *a priori* classes. In the cluster analysis approach, geometrical relationships on a set of sample data in a training process are established (Bracio *et al.*, 1997). Other approaches include unsupervised or self-organised learning algorithms such as K-means clustering and the associated adaptive update rule, the Kohonen feature map (Kohonen, 1988). To fuse sensory data in an adaptive manner and allow to adjust automatically the granularity of the classifier and to maintain stability against category proliferation in the presence of drifting inputs and changing environments, adaptive resonance theory (ART) (Carpenter and Grossberg, 2003) and Fuzzy ART network (Carpenter and Grossberg, 1996) approaches can be used.
3. *Inference methods.* Bayesian inference and Dempster-Shafer evidential reasoning are the main approaches in inference methods. Bayesian inference provides formalism for MSDF that allows sensory data to be fused according to the rules of probability theory. This approach relies on the use of Bayes' rule where a relationship between the *a priori* probability of a hypothesis, the conditional probability of an observation given a hypothesis and the *a posteriori* probability of the hypothesis is provided (Hall, 1992). An immediate problem in this approach is that the required knowledge of the *a priori* probability and the conditional probability may not be always available. Also in defining these probabilities, often subjective judgements are necessary (Brooks and Iyengar, 1998). An extension to the Bayesian inference method, Dempster-Shafer evidential reasoning, overcomes these drawbacks by keeping track of an explicit probabilistic

measure of the lack of information concerning a proposition's probability. The cost of this approach is the additional time required for computation.

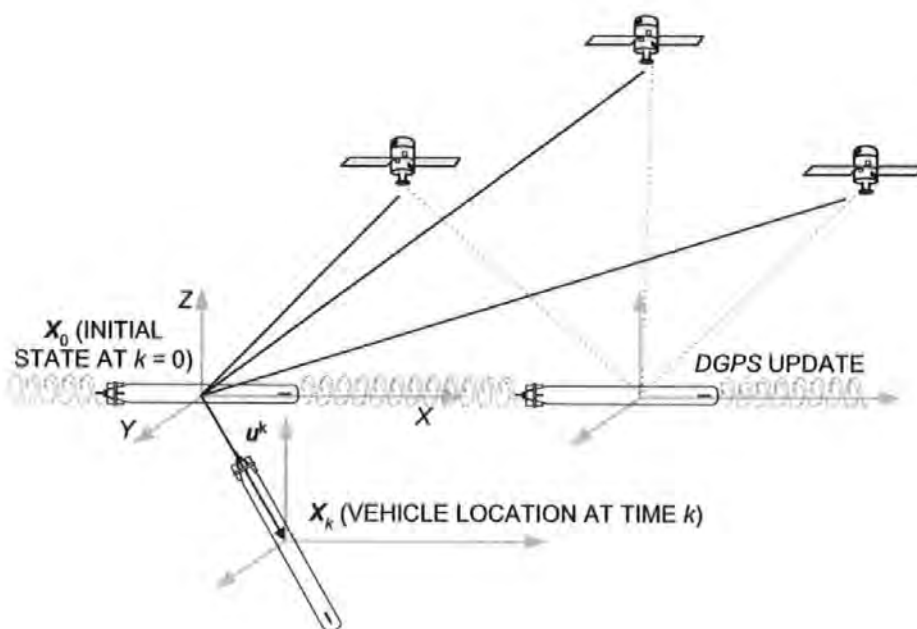
4. *Artificial intelligence methods.* Artificial intelligence is a vast, loosely defined area encompassing various aspects of pattern recognition and image processing, natural language and speech processing, automated reasoning and a host of other disciplines. Fuzzy logic and neural networks are two of the most widely used approaches in artificial intelligence methods for combining multisensor data. Fuzzy logic involves the extension of Boolean set theory and Boolean logic to a continuous-valued logic via the concept of membership functions to quantify imprecise concepts. A neural network is a method designed to mimic how biological nervous systems work. In this method, an individual neuron takes weighted input from a number of sources, performs a simple function and then produces a single output when the required threshold is reached. Neurons can be trained to represent sensor data and, through associate recall, complex combinations of the neurons can be activated in response to different sensor stimuli (Luo *et al.*, 2002).

2.4 APPLICATIONS OF MULTISENSOR DATA FUSION FOR AUTONOMOUS UNDERWATER VEHICLES

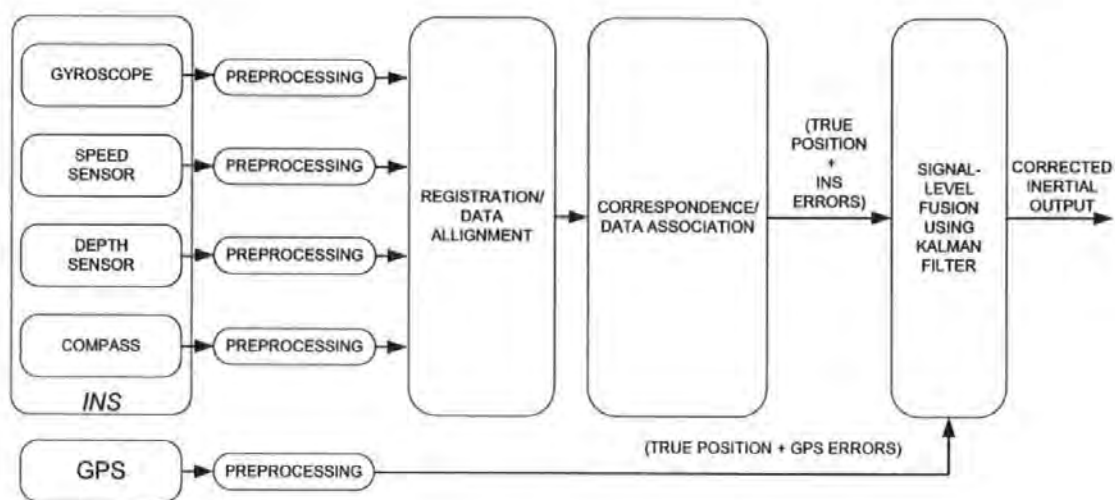
The discussion here focuses on a variety of approaches to the fusion of information from combinations of different types of sensors.

2.4.1 GPS and Inertial-Based Systems

McGhee *et al.* (1995) describe a navigation system employed by the *Phoenix* AUV using an inertial and differential DGPS navigational suite to conduct shallow water mine detection and coastal environment monitoring missions. In the course of its mission *Phoenix* combines signal-level information from a gyroscope, depth sensor, speed sensor, and a compass heading to predict its position while operating underwater. The vehicle surfaces periodically to obtain an update of its position from a DGPS fix and then submerges (Figure 2.6(a)). Problems with this setup concern



(a)



(b)

Figure 2.6: GPS/INS navigation as described by McGhee *et al.* (1995)

the time required to acquire the DGPS data and the influence of water covering the DGPS antenna during position fixing were examined in Norton (1994). The inertial navigation sensors described in McGhee *et al.* (1995) obtain accelerations and angular rates of change for the vehicle. A nine-state KF is used to process the data and to give the prediction of the vehicle position. The DGPS data is then used to update the predicted position resulting in an estimated position. The nine state KF can be divided into seven continuous-time states (three Euler angles, two horizontal velocities, and two horizontal positions) and two discrete-time states (estimated east and north current derived from the DGPS fixes). The method used to fuse sensory information discussed by McGhee *et al.* (1995) can be shown as in Figure 2.6(b).

The main problem with the KF employed in McGhee *et al.* (1995) is the need for a tuning system to prevent filter divergence. This problem can be overcome by the use of artificial intelligence (AI) techniques as have been applied in helicopters (Doyle and Harris, 1996), automobiles (Kobayashi *et al.*, 1998) and target tracking system (McGinnity and Irwin, 1997) applications. Kobayashi *et al.* (1998) wished to determine accurately the position of an automobile using DGPS. In their work, a fixed fuzzy rule based algorithm is used to tune the covariance factors of a KF. The shape and positioning of the various fuzzy sets on their respective universes of discourse having been decided by heuristic means. The main problem with the Kobayashi *et al.* (1998) methodology is the reliance on trial and error to generate the fuzzy rule based algorithms. Similar comments can also be made concerning the robot positioning work of Jetto *et al.* (1999). To overcome such drawbacks genetic algorithms (GAs) (Pham and Karaboga, 1991; Sutton and Marsden, 1997) have been used to optimise fuzzy systems. Other intelligent optimization techniques such as chemotaxis, alopex and simulated annealing have also been successfully employed in the design optimization of fuzzy control systems (Sutton *et al.*, 1996; Sutton *et al.*, 1997).

2.4.2 Acoustic-Based Systems

Atwood *et al.* (1995) have built and tested an AUV that utilises a LBL navigation system with an innovative fix-finding algorithm and commercially available hardware.

They use a spherical navigation system, in which the vehicle actively interrogates acoustic transponders and calculate ranges from round trip transit times, resulting in a greater accuracy (about 1(m)) compared to the hyperbolic method proposed by Bellingham *et al.* (1994). In this system, the vehicle can use two operating modes, master mode and transponder mode. In the first mode, the vehicle triggers the acoustic transponders, which reply with an acoustic signal. The vehicle computer can then calculate distances and, applying acoustically measured depth, a position. Using the first mode, operation over an area of $1(km)^2$ is possible. In the second operating mode, a surface vessel triggers the vehicle, which in turn interrogates the transponders. Position of the AUV can then be calculated in the surface vessel through an established GPS position and knowledge of the relative positions of the AUV and the transponders. This procedure is called the fish solution, as it lets the operator on the ship monitor vehicle progress. The second mode is developed to have operational areas as large as $10(km)^2$. In this work, Atwood *et al.* (1995) have solved the problem of fading or destructive interference of the acoustic signals produced by the transponders encountered by Bellingham *et al.* (1994). Atwood *et al.* (1995) principally combine sensor information at signal-level data.

Rendas and Lourtie (1994) combine LBL navigation with dead reckoning and calls it a *hybrid system*. The vehicle travels between deployed baseline arrays, each consisting, for example, of four transponders, and uses acoustic navigation when in range of an array. Outside the range, it uses a sonar/Doppler sensor and depth information for autonomous navigation. The distances between the arrays must be carefully planned, because the accuracy of navigation in the autonomous mode deteriorates with time, depending on the quality of the sensing systems. The transition from one mode to another takes place automatically. When the vehicle is leaving the area where a particular baseline array is located, the number of range measurements it is able to receive will gradually decrease to zero, entering, in this way, the autonomous navigation mode. On the contrary, when it approaches an area where transponders are located, it receives an increasing number of distance measurements, switching from autonomous to local navigation mode. The system uses a variable dimension Kalman filter for both navigation modes. Where there is no detectable acceleration, the filter assumes uniform motion and estimates position and linear velocity. When

there is acceleration, the filter switches to a larger order (manoeuvring model) and extends its state vector to include the accelerations. In this work, however, Rendas and Lourtie (1994) have not taken into account the analytical approximations to the error evolution during autonomous navigation to determine the layout of the baseline arrays and to derive the constraints on path planning once a layout has been decided upon. Similar to Atwood *et al.* (1995), the MSDF method used by Rendas and Lourtie (1994) is an estimation method which fuses data from the navigation sensors at signal-level.

2.4.3 Acoustic- and Visual-Based Systems

Majumder *et al.* (2000a), Majumder *et al.* (2000b) and Majumder *et al.* (2001) reported the use of sonar and underwater cameras to construct a complete environmental map for navigation. A generic, multi-layered data fusion scheme is used to combine information from the two sensors. The general principle is that all sensor information is projected into a common state-space before the extraction of seabed features. Once projection has occurred, feature extraction and subsequent processing is based on a combined description of the environment. As robust features, such as points and lines turn out to be fragile in a natural underwater environment, Majumder and co-workers found that this approach is better than extracting features from a single piece of sensor information followed by fusion. In this work, "blobs" and blob-like patches are used as scene descriptors to segregate feature information from background noise and other errors. Majumder *et al.* (2000a), Majumder *et al.* (2000b) and Majumder *et al.* (2001) discussed both the Bayesian and extended Kalman filter (EKF) approaches to map-building and localisation in autonomous navigation systems. It was shown in this work that a significant problem in applying EKF is the difficulty of modelling natural environment features in a form that can be used in an EKF algorithm. Another formidable problem is the fragility of the EKF method when faced with incorrect associations of observations to landmarks. The limitations in using this to build a feature map of landmarks describing the environment were then resolved through the use of the Bayesian approach. The fusion process can be shown as in Figure 2.7. A significant problem with this approach lies on the stability of the algorithm when the

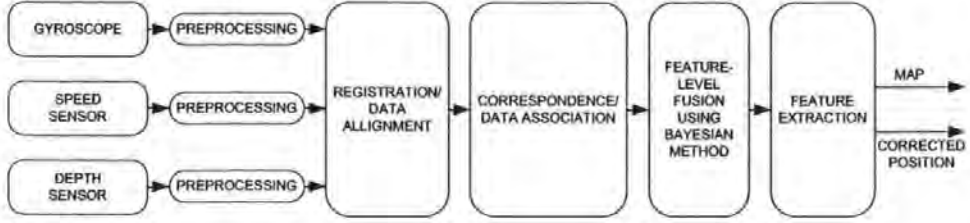


Figure 2.7: MSDF in SLAM algorithm as described by Majumder *et al.* 2000a; Majumder *et al.* 2000b; Majumder 2001 *et al.*

vehicle is run over long distances and returning around a loop to the initial vehicle location. This problem stems from the limitation in data association technique to correspond initially identified landmarks and the same landmarks viewed from the opposite side on the return visit. A potential solution to this problem is to use a probabilistic model to provide a very general description of landmarks form and shape. *Twin Burger 2*, an AUV developed by the University of Tokyo, was designed to help monitor and carry out routine maintenance work of underwater cables (Balasuriya and Ura, 1999a; Balasuriya and Ura, 1999b). In so doing, the vehicle tracks the cable visually and provides human operators with visual information about the condition of the cable accordingly. Initially the vehicle employed a visual servoing system to track the cable and to navigate the AUV. However, due to undesirable optical behaviour underwater, there were many occasions where the cable was not visible enough for the vision processor to track the cable. In addition the vehicle can lose track of the cable when there were many similar cables appearing in the image. In order to overcome these problems, a multisensor fusion technique is proposed. The proposed sensor fusion technique uses dead reckoning position uncertainty with a 2D position model of the cable to predict the region of interest in the image captured by a camera mounted on the AUV (Balasuriya and Ura, 1999a; Balasuriya and Ura, 1999b). The 2D-position model of the layout of the cable is generated by taking the position (x_i, y_i) of a few points along the cable. The 2D-position model of the cable is used to predict the most likely region of the cable in the image, which leads to a reduction in the amount of image data and a decrease in the image processing time. Additionally, due to the narrowing of the region of interest in the image, the chances of misinterpretation of similar features appearing in the image can be avoided. The 2D-position model is

also used to generate navigation commands when the vision processor cannot recognise the cable in the environment. Similar to Majumder *et al.* (2000a), Majumder *et al.* (2000b) and Majumder *et al.* (2001), the fusion process takes place at feature-level.

Scheizer and Petlevitch (1989) has reported a target detection and classification system using side scan sonar data and vision. Objects are detected by searching for highlights, textures, statistical anomalies and shadows. A neural network-based classification system is used to assist the image-processing component. The classification process does not identify objects but rather labels them as foreground, background, highlight, or shadow highlight. The level of correct classification is reported to be 95 per cent using a training set of 62 images. This technique, however, does not address the issue of feature or object identification.

2.5 CONCLUDING REMARKS

It has been suggested in this chapter, from the various examples given in AUV navigation, that information coming from a single navigation system is not sufficient to provide a good navigation capability. Therefore MSDF techniques which combine sensory information from other navigation systems to improve the navigation capability is essential. These will underpin the theoretical and practical work of this thesis that aims to design and develop an interactive navigation system that consists of several INS sensors integrated with a GPS to interact with an appropriate guidance and control system, implemented to the *Hammerhead* AUV. The next chapter presents an introduction to the vehicle, the hardware setup and sensors used within. System identification approaches and results obtained are also discussed in detail.

CHAPTER 3

THE *HAMMERHEAD* AUTONOMOUS UNDERWATER VEHICLE

This chapter presents a general introduction to the structure of the *Hammerhead* AUV, followed by a description of its hardware setup and sensor configuration used therein. A thorough discussion of system identification experiments and the results thereof are also given. This forms the foundation of the proposed techniques discussed in the following chapters.

3.1 THE EVOLUTION STRUCTURE



Figure 3.1: The DMT deployed from a chute

The *Hammerhead* AUV was built on the structure of a DMT used in training exercises by the British Admiralty in the 1960s and 1970s. The DMT was originally deployed from a chute (see Figure 3.1). The gravity force exerted on the vehicle was able to thrust the vehicle under the water and to introduce an additional force to the initial velocity. In its original configuration, the DMT was powered at 130(VDC) and was able to move at 12(knots). The hydroplanes at the rear of the vehicle were used to change the direction of the vehicle vertically, while the rudder was used to change the



Figure 3.2: (a) Rudder and rear hydroplanes, (b) front hydroplanes

direction horizontally (see Figure 3.2(a)). However, without the presence of the chute that enables the vehicle to get under the water at the beginning of an operation, the hydroplanes were unable to perform their function properly and this was observed during the initial *Hammerhead* full scale trials. It was then decided to modify the vehicle by adding a new set of hydroplanes constructed at the section adjacent to the nose of the vehicle (see Figure 3.2(b)) and to lock the rear hydroplanes. Subsequent full scale trials proved that the new structure made diving from the surface at a low speed of operation possible. The deflections for the rudder and hydroplanes being restricted to $\pm 22.5(deg)$ and $\pm 25(deg)$ respectively and were deemed sufficient for heading and depth control purposes. The new shape, as it appears now, mimic the feature of a *Hammerhead* shark and this was one of the catalysts to name the AUV as *Hammerhead*.

Another major modification to the DMT was the addition of a camera port, shown in Figure 3.3(a), to the same section where the new set of hydroplanes installed. The camera is a charge couple device (CCD) type and work together with a laser scanning unit, shown in Figure 3.3(b). This unit is used to provide an enhanced viewing below the vehicle as it performs an underwater operation. Together, the CCD camera and the laser scanning unit are used to provide navigation data and to gather underwater images for surveying purposes.

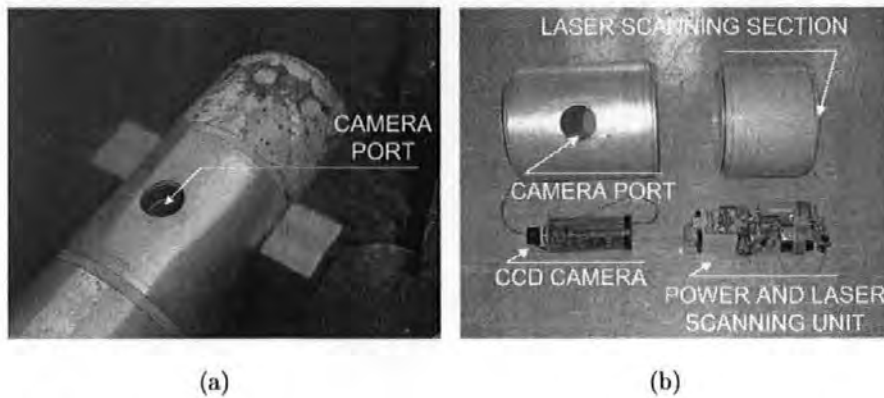


Figure 3.3: (a) Camera port, (b) camera and laser scanning unit

The physical structure of the *Hammerhead* AUV (the modified DMT) can be summarised as in Table 3.1. The schematic of the vehicle is shown in Figure 3.4(a), with the actual vehicle strapped on its trailer ready for deployment is shown in Figure 3.4(b).

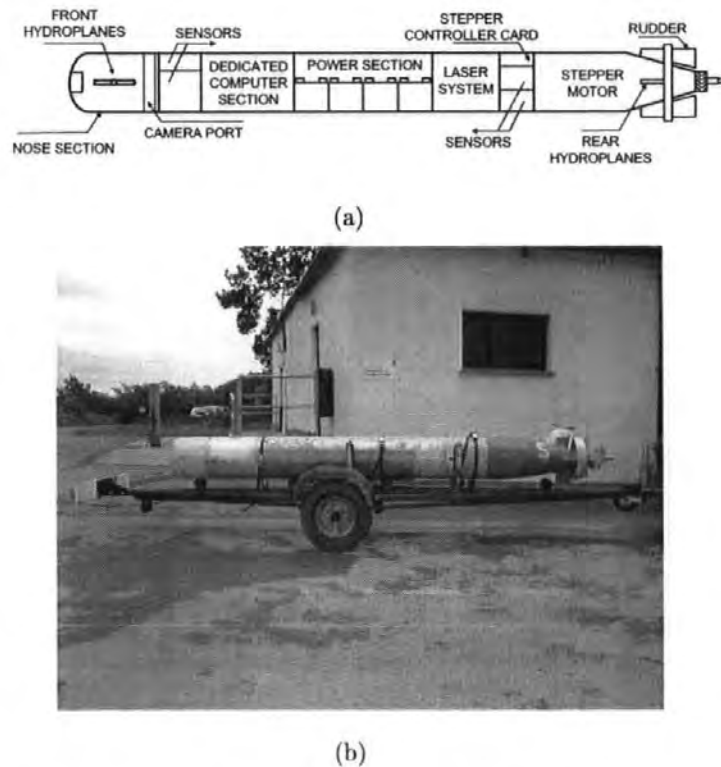


Figure 3.4: (a) The schematic of the *Hammerhead*, (b) The *Hammerhead* on its trailer

Length	3.5(<i>m</i>)
Diameter	0.35(<i>m</i>)
Weight	250(<i>kg</i>)
Weight in water	2(<i>kg</i>)
Rudder deflection	$\pm 22.5(deg)$
Hydroplane deflection	$\pm 25(deg)$
Speed	2(<i>knots</i>)
Depth Capability	100(<i>m</i>)
Battery capacity	1.6(<i>kWh</i>) or 3 – 4(<i>hours</i>)
Propulsion	Rear DC thruster 100(<i>N</i>)
Materials	Anodised aluminium hull
Launch and recovery	Submersible trailer and rigid inflated boat (RIB)

Table 3.1: The physical structure of the *Hammerhead* AUV

A discussion on the complete hardware setup and sensor configuration used in the vehicle is given in the next section.

3.2 HARDWARE SETUP

The initial hardware setup of the *Hammerhead* resembles many of those in ROVs, *i.e.*, the processing of control signals are still carried out by an off-board operator which sends appropriate control commands to the vehicle through a tether. In this particular case, however, the operator is replaced by an automatic control unit. It should also be noted that the power to the vehicle is now generated internally. Therefore, due to the addition of the tether and the remotely located control unit, the vehicle can be regarded as still running in a semi-autonomous mode at this stage.

The hardware in this setup can generally be considered to be divided into two major groups: a mobile real-time sensing unit and, a mobile real-time data-processing and control unit. The mobile sensing unit consists of the vehicle and all sensors mounted thereon: one GPS receiver unit, one IMU, one electronic compass known as TCM2

and one pressure transducer. These, with the exception of the pressure transducer, are RS232/serial compliance devices. The pressure transducer produces an analog voltage output and need to be converted to a digital form that is compatible with RS232/serial voltage levels using a MAX232 chip embedded in an analog to digital converter (ADC) card. The mobile sensing unit automatically measures the vehicle position in the Earth-centred Earth-fixed co-ordinate frame at $1(Hz)$, acceleration and angular rate at up to $50(Hz)$ in both the body and NED co-ordinate frame, orientation and depth at $8(Hz)$ in the NED co-ordinate frame.

The mobile data-processing and control unit consists of a Pentium-4 ($1.6(GHz)$) laptop. The data-processing and control softwares were developed using MATLAB with extensive use of its RS232/serial port I/O facility. The speed of data-processing is constrained by the sequential nature of MATLAB command execution and therefore limited to only $8(Hz)$. Control commands to drive the stepper motor of the rudder and hydroplanes (Figure 3.5) of the vehicle are also sent through the RS232/serial port. For this purpose, an interface which can receive an input from the mobile control unit and converts it into the stepper motor driving pulses was sought. A microcontroller based board using an ATMEL 89C2051 chip was then developed. Details of the design and development can be found in Naeem (2004).

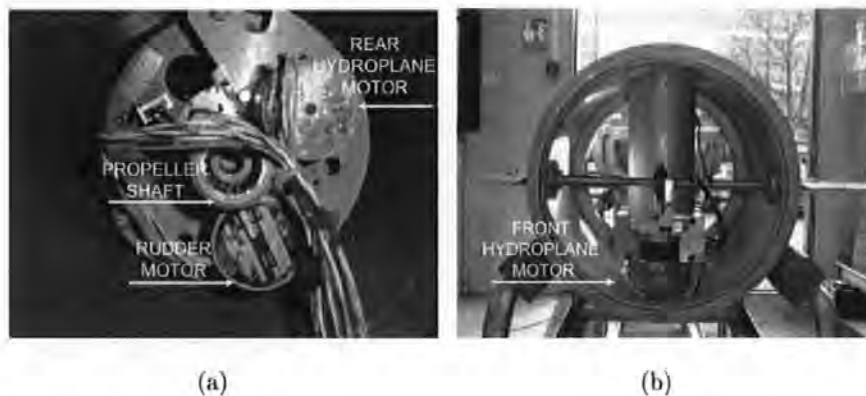


Figure 3.5: (a) Rudder and rear hydroplanes motor, (b) front hydroplanes motor

It is important to note that the laptop has only one RS232/serial port and therefore, insufficient to cater the I/O requirements of the data-processing and control unit. A QUATECH PCMCIA to $4 \times$ RS232/serial converter, shown in Figure 3.6, is included

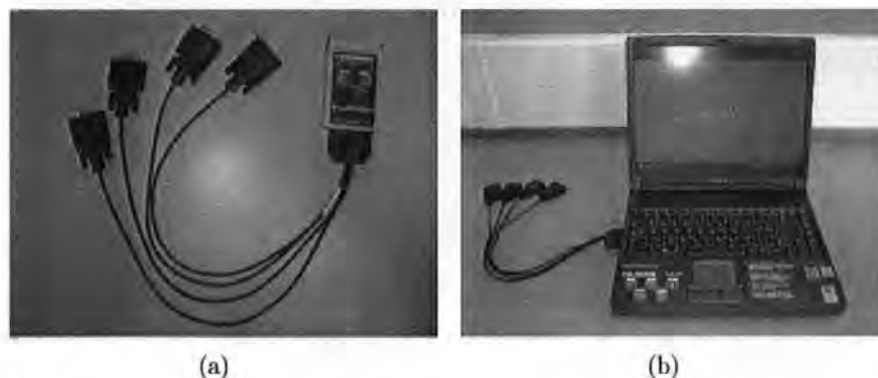
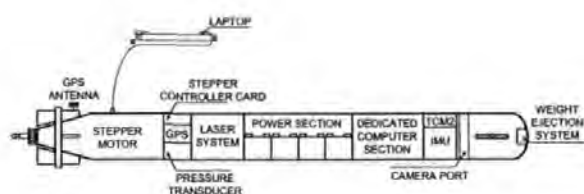


Figure 3.6: (a) PCMCIA to 4xRS232/serial converter, (b) converter-laptop connection

to acquire all sensor data. The GPS, IMU and TCM2 data can all be automatically received by the RX pin of the RS232/serial ports, while the pressure transducer require a certain 8-bits sent from the laptop to the ADC card through the TX pin to indicate the card that data are ready to be received. The control commands are sent through the TX pin of the same RS232/serial port where the RX pin used by the TCM2. More sensors can be accommodated, providing a certain or several RX pins shared/multiplexed by two or more sensors. The multiplexing process can be achieved, for instance, using a PHILIPS 74HC/HCT151 8-input multiplexer.

A 20-core thin cable is used to transmit all sensor data from the mobile real-time sensing unit to the real-time data-processing unit. This is also used to transmit the command signals from the real-time control unit to the stepper motor controller card. Two cores of the cable are connected to the leak detector unit inside the vehicle. At the other end of these two cores, a loudspeaker is connected to provide an audible leak warning. If such a case occurs, another two-cores of the cable has been dedicated to enable manual shutdown of the vehicle. In cases where the vehicle dives, due to, for instance, an imperfection in the mechanical structure of the hydroplanes, a sudden change of liquid hydrodynamic surrounds the vehicle and a strong vertical tide of water to the surface of the vehicle, an emergency weight ejection system will release a weight on the nose of the vehicle and consequently, due to inherent positive buoyancy, the vehicle will be brought back to the surface. The length and diameter of the cable are chosen carefully to minimise drag effects caused thereby that might produce a

significant disturbance to the dynamic of the vehicle. The overall hardware setup and sensor configuration mounted on the vehicle are shown in Figure 3.7(a). Figure 3.7(b) shows the mobile sensing unit, and the mobile data-processing and control unit during a full scale trial.



(a)



(b)

Figure 3.7: (a) The *Hammerhead* hardware setup and sensor configuration, (b) full scale trial

During the first half of the *Hammerhead* project, most full scale trials were conducted using the setup shown in Figure 3.7. In the second half of the project, the mobile real-time data-processing and control unit have been replaced by two on-board CPUs. This setup eliminates the need for a tether and consequently, the drag encountered by the vehicle was minimised. The two CPUs, referred to as the Host and the Navigator+Controller, work side by side to perform an autonomous mission. The Host CPU is responsible for all sensor acquisition and intervention, including failsafe emergency systems and all automated imaging parameters. To accomplish these, the Host CPU is equipped with a frame grabber, a multifunction data acquisition card and six RS232/serial ports. The programs to perform these tasks are developed in LABVIEW and are initiated via the wireless Ethernet link (IEEE802.11b wireless local area network(WLAN)) operative on an off-board laptop. The WLAN has a range of 200(m) at 11(Mbits/sec), extendable by reducing the bandwidth. The Navigator+Controller CPU receives a navigational parameter string sent by the Host CPU and perform a control action by sending appropriate pulses to the rudder and hydroplanes. The navigation and control systems are developed in MATLAB and are resident on this CPU. Figure 3.8(a) and Figure 3.8(b) respectively, show the vehicle with a 'tetherless' hardware setup and its recently conducted full scale trial.

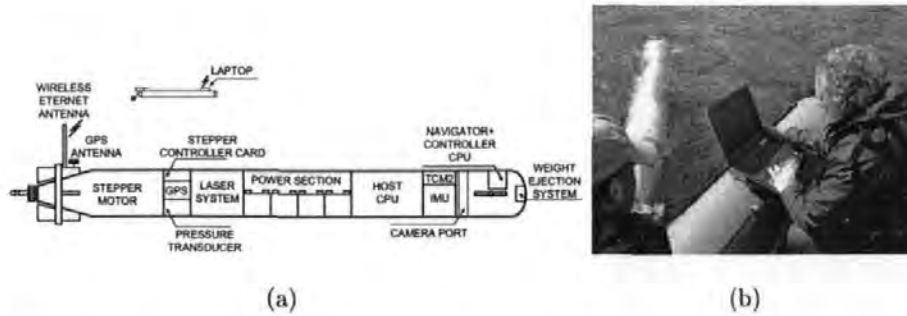


Figure 3.8: (a) The wireless *Hammerhead* hardware setup and sensor configuration, (b) full scale trial

Details of the sensors employed in both vehicles' setup are presented in the next section.

3.3 DETAILS ON THE SENSORS

The sensors used in the mobile real-time sensing unit have been discussed briefly in the previous chapters and as well in the beginning of this chapter. A more comprehensive treatment is given in this section.

3.3.1 GPS

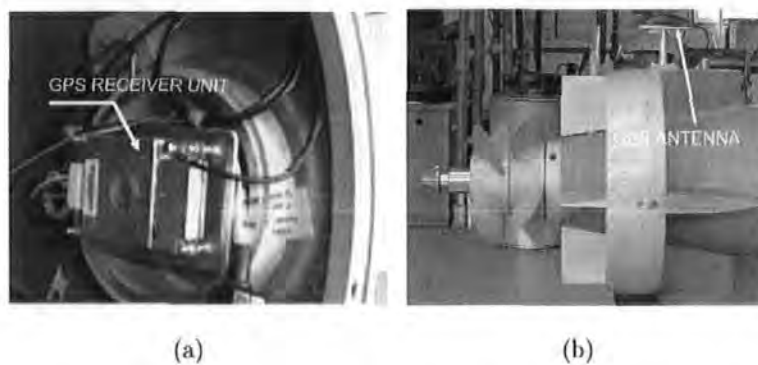


Figure 3.9: (a) Mounted GPS unit, (b) installed GPS antenna

Element	Description
\$	Start
< 1 >	Universal time coordinated (UTC) of position fix
< 2 >	Latitude
< 3 >	Latitude hemisphere, N or S
< 4 >	Longitude
< 5 >	Longitude hemisphere, E or W
< 6 >	GPS quality indicator
< 7 >	Number of satellites in use, 00 to 12
...	
hh	Check sum
< CR >	Carriage return
< LF >	Line feed

Table 3.2: *GPGGA* sentence

A GARMIN GPS 25LVS is used in the mobile real-time sensing unit. Figure 3.9(a) and 3.9(b) show the hull-mounted unit and tail-installed antenna respectively. It is designed to operate from a low voltage 3.6(VDC) to 6.0(VDC) supply and conforms to the RS232/serial standard. As the data acquired by the serial port of the mobile real-time data-processing unit, communication speed is limited to 4800 baud rate. Position accuracy is 15(m) RMS. It tracks up to 12 satellites with 1(Hz) update rate. The interface protocol design on the TX/RX is based on the national marine electronics association's (NMEA's) 0183 ASCII interface specification. The NMEA 0183 navigation information transmitted by the unit posses a common structure, which includes a message header, data fields, and a terminating carriage return and line feed as the following example :

\$GPGGA,< 1 >,< 2 >,< 3 >,< 4 >,< 5 >,< 6 >,< 7 >,...*hh < CR >< LF >

Table 3.2 provides a detailed description of the above *GPGGA* sentence. Other sentences like *GPRMC*, *GPGSA* and *PGRME* can provide more extensive information such as ambient magnetic variation, position dilution of precision, and estimated 3D position error. These, if taken into account in the proposed adaptive Kalman filtering algorithms can certainly be valuable in enhancing the overall accuracy of the *Hammerhead* navigation system. However, in this thesis, focus is given to the acquisition of the absolute positioning information, *i.e.*, the latitude and longitude, and their use in updating the position derived by the INS sensors. Chapter 6 will discuss possible

implementations of the other GPS information, especially in relation to the proposed adaptive Kalman filtering techniques.

3.3.2 TCM2 Electronic Compass

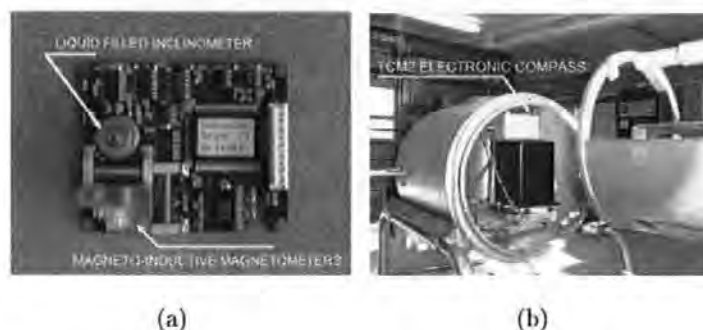


Figure 3.10: (a) TCM2 circuitry, (b) mounted TCM2

Heading Information	Values
Accuracy at tilt = 0(deg)	0.5(deg) RMS
Accuracy at 0(deg) < tilt or tilt > 0(deg)	1.0(deg) RMS
Resolution	0.1(deg)
Repeatability	$\pm 0.1(deg)$
Tilt Information	Values
Accuracy	$\pm 0.1(deg)$
Resolution	0.1(deg)
Repeatability	$\pm 0.2(deg)$
Range	$\pm 20(deg)$

Table 3.3: TCM2-20 technical specification (courtesy of Precision Navigation website www.precisionnav.com)

The TCM2 (shown in Figure 3.10) contains a combined triaxial magneto inductive magnetometer and biaxial fluid filled inclinometer. The TCM2 uses a unique technology, which enables accurate data to be obtained without having to reference the compass level to the horizon. This is made possible by an electronic gimbaling process performed by the in-built microprocessor, which takes the inclinometer data and

computes a tilt correction mathematically. The electronic gimbaling eliminates moving parts and provides more information about the environment: pitch and roll angles and 3D magnetic field measurement in addition to compass output. Table 3.3 provides the technical specification of the heading and tilt information of the TCM2.

The TCM2 also features a magnetic distortion alarm. This is to indicate magnetic anomalies that can compromise compass and magnetometer accuracy. The magnetic anomalies are detected during a continuous data sampling by an algorithm that evaluates the quality of the magnetic environment. If a significant deviation is detected between an instantaneous magnetic field information against a stored reference, the magnetic distortion alarm error flag is raised. A host magnetic reference is obtained by performing a user calibration. It was noted, that as a host, the mounting location of the TCM2 inside the *Hammerhead* vehicle produces only an infinitesimal static magnetic vector contribution to the local Earth's field. Also, the vehicle was mostly deployed in a magnetically 'benign' environment. Therefore, it was decided to reserve the magnetic distortion alarm feature only for future developments of the adaptive Kalman filtering algorithm, especially in compensating the effect of anomalies that are caused by ambient dynamic sources (see future works discussion on Chapter 6).

Element	Description
\$	Start
$C < . >$	Compass
$P < . >$	Pitch
$R < . >$	Roll
...	
hh	Check sum
$< CR >$	Carriage return
$< LF >$	Line feed

Table 3.4: TCM2 string

The TCM2 sends an ASCII data across the RS232/serial link, which takes the following standard format:

$$\$C < . > P < . > R < . > ...*hh < CR > < LF >$$

Table 3.4 provides the description of the string. For the current adaptive Kalman filtering algorithm, only the $C < . >$, $R < . >$, and $P < . >$ are used. The rest of the entries form the basis of future works.

3.3.3 Inertial Measurement Unit



Figure 3.11: (a) Watson Industries Ltd. inertial measurement unit, (b) mounted inertial measurement unit

The *Hammerhead* IMU (manufactured by Watson Industries), shown in Figure 3.11, consists of a 3-axis solid-state rate-gyro and accelerometer to measure angular measurements and linear acceleration respectively. The gyros are vibrating cylinder types while the accelerometers are made using silicon micro-machining technology. Table 3.5 provides the technical specification of the unit. The unit also sends ASCII data across the RS232/serial link. For the purpose of the present work, only angular displacement and linear acceleration data are used.

It is important to recognise that two sets of measurement co-ordinate frame are used in the unit. One set of measurement co-ordinate frame is referred to the body co-ordinate frame, while the other set is the NED co-ordinate frame. Just like in the TCM2, the term angular displacements are measurements made with respect to the Horizon. However, angular rates are measurements made with respect to the body

Angular Rates	Values
Range	$\pm 100(deg/sec)$
Accuracy	$\pm 2\%$ of scale value
Resolution	0.1 (deg/sec) ASCII format data
Tilt	Values
Range	$\pm 0.3(deg) \pm 2\%$ of scale value, $+0.5(deg)$ per g due to linear acceleration
Resolution	0.1(deg) ASCII format data
Heading	Values
Range	$+360(deg)$
Accuracy	$\pm 1(deg)$
Resolution	0.1(deg) ASCII format data
Linear Acceleration	Values
Range	$\pm 2(g)$
Accuracy	$\pm 10(mg)$
Resolution	1(mg) ASCII format data

Table 3.5: IMU technical specification

co-ordinate frame. Longitudinal, lateral and vertical acceleration are made with respect to the body co-ordinate frame, while the X-, Y-, and Z-axis acceleration are with respect to the NED co-ordinate frame.

To obtain an GPS/INS positioning, the IMU acceleration made in the body co-ordinate frame is first integrated once to obtain linear velocity:

$$[\mathbf{V}]_{BODY(t_0+\delta t)} = \int_{t_0}^{t_0+\delta t} [\dot{\mathbf{V}}]_{BODY(t)} dt + [\mathbf{V}]_{BODY(t_0)} \quad (3.1)$$

Integration of the new body velocity to determine position is preceded by a transformation from the body co-ordinate frame to the NED co-ordinate frame (see Appendix G for detail). The following substitution pertains:

$$\int_{t_0}^{t_0+\delta t} [\mathbf{V}]_{NED(t)} dt = \int_{t_0}^{t_0+\delta t} ([\mathbf{V}]_{BODY(t)})_{BODY \rightarrow NED} dt \quad (3.2)$$

The final integration to determine position is therefore:

$$[\mathbf{P}]_{NED(t_0+\delta t)} = \int_{t_0}^{t_0+\delta t} [\mathbf{V}]_{NED(t)} dt + [\mathbf{P}]_{NED(t_0)} \quad (3.3)$$

3.3.4 Pressure Transducer

The pressure transducer, shown in Figure 3.12 produces an analog signal between $0(VDC)$ to $5(VDC)$, which is equivalent to $0(bar)$ to $6(bar)$. The analog signal is converted to a RS232/serial signal using an ADC designed at the University of Plymouth. Appendix C provides the details of the design. The ADC uses an ADC0804 chip to do the conversion, an AT89C2051 microcontroller to control the ADC0804, and a MAX232 chip to convert the signals from and to RS232/serial levels for sending and receiving from the laptop. The converted signal is further processed by the data-processing unit to produce depth in meter.

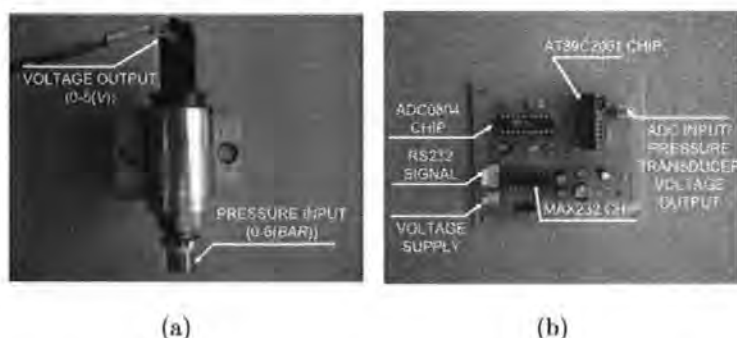


Figure 3.12: (a) Pressure transducer, (b) ADC card

The hardware setup and sensor configuration discussed herein are used to facilitate the process of building up an integrated navigation, guidance and control for the vehicle. Full scale trials conducted by the *Hammerhead* research team were designed to obtain all-important components of the integrated system. Both the navigation and, the guidance and control systems share one important component, the dynamic model of the vehicle. This can be achieved through either finding the hydrodynamic coefficients of the vehicle or through system identification (SI) methods. These approaches are discussed in the next section. The discussion is compiled from Naeem (2004). However, it should be noted that the results presented henceforth are produced from a combined effort by the author of this thesis and by Naeem (2004).

3.4 SYSTEM IDENTIFICATION FOR THE *HAMMERHEAD* AUV

To obtain a model of an AUV is an intricate task because of the nonlinear nature of the vehicle dynamics and the degrees of freedom of vehicle movement. Each of the implementation of the approaches of finding the dynamic model of an AUV, *i.e.*, mathematical modelling (based on finding the hydrodynamic coefficients of the vehicle) and system identification, is highly constrained by the physical characteristics of the vehicle and the facility available to perform the routines therein.

Mathematical modelling of AUVs is widely addressed. However, several parameters still pose uncertainties due to the difficult nature of the problem. Of prime importance in this context is the dependence of many hydrodynamic parameters and coefficients on varying velocity regimes, proximity to the sea bed, sea surface and other structures, just to mention a few. Certain model parameters can be determined analytically. Other parameters, however, will need to be identified using scaled model or full scale tank tests. For example, the *Subzero-II* vehicle based at The Institute for Sound and Vibration Research, University of Southampton, UK, has over 70 rigid body and hydrodynamic coefficients to be estimated. Twelve of these were obtained by calculation or experiments. Lack of tank test facilities prevent the evaluation of the coefficients to only four whereas the remaining coefficients used were scaled down versions of the *Ocean Voyager* vehicle, which is similar in shape to *Subzero-II* (Ahmad and Sutton, 2003).

Taking into consideration the physical characteristics of the *Hammerhead* AUV and the insufficient test tank facilities available to perform the experiments required by this type of vehicle, an alternate route using SI techniques to obtain the dynamic models of the vehicle is thus suggested and used in this thesis. SI is quite effective in providing reliable and accurate models based only on input-output data sets obtained from AUVs full scale trials. This is the main appeal of the approach and makes it quite desirable to be employed in modelling AUVs whose configuration changes frequently to suit their mission requirements. Details of the SI implemented to the *Hammerhead* AUV and the results thereof are given in the following discussion.

3.4.1 System Identification

Obtaining dynamic models of AUVs using SI approaches have been investigated before (Bossley *et al.*, 1999; Goheen and Jefferys, 1990; Ahmad and Sutton, 2003), but most of the work involved has been undertaken on identifying a model by generating data from a 6 DOF mathematical model of the vehicle. However, in this thesis, the SI is performed on actual AUV input/output data obtained from test trials. The SI of a dynamical system generally consists of the following four steps: data acquisition (DAQ) data acquisition, characterisation, identification/estimation and verification.

The first and most important step is to acquire the input/output data of the system to be identified. Acquiring data is not trivial and can be very much laborious and expensive. This involves careful planning of the inputs to be applied so that sufficient information about the system dynamics is obtained. If the inputs are not well designed, then it could lead to insufficient or even useless data. Other factors that could degrade the data quality includes the DAQ hardware involved and sampling rate. These will be discussed in detail in the following section.

The second step defines the structure of the system to be identified, for example, type and order of the differential equation relating the input to the output. This means selection of a suitable model structure, *e.g.* auto-regressive with exogeneous input (ARX), auto-regressive moving average with exogeneous input (ARMAX), output error, *etc.* If there is significant amount of noise in the data then it could be modelled separately by specifying an appropriate model type. A generic input-output linear model for a single output system can be written as (Ljung, 2001).

$$A(q)y(t) = \sum_{i=1}^{nu} \left[\frac{B_i(q)}{F_i(q)} \right] u_i(t - nk_i) + \left[\frac{C(q)}{D(q)} \right] e(t) \quad (3.4)$$

where q is the shift operator and $A(q)y(t)$ is short for

$$A(q)y(t) = \sum_{k=1}^{\infty} a(k)y(t - k) \quad (3.5)$$

and

$$A(q) = \sum_{k=1}^{\infty} a(k)q^{-k}; q^{-1}y(t) = y(t-1) \quad (3.6)$$

u and y are the input and output respectively, u_i represents the i th input and nu represents the number of input. A , B_i , C , D and F_i are polynomial in q and work as defined in Equation (3.5) and (3.6), nk denotes the time delay in the system and e is the disturbance. All the above mentioned models can be obtained from the generic model structure by substituting the appropriate values of the polynomials.

The third step is identification/estimation, which involves determining the numerical values of the structural parameters, which minimise the error between the system to be identified, and its model. Common estimation methods are least squares (LS), instrumental-variable (IV), maximum-likelihood (MLE) and the prediction-error method (PEM). A common criterion used in most optimization methods is the quadratic error function given by

$$\min_{\Gamma} J = \frac{1}{N} \sum_{i=1}^N (\hat{y}(t) - y(t))^2 \quad (3.7)$$

where \hat{y} is the predicted output from the model, y represents the actual output, N denotes the number of data points and Γ contains the coefficients to be estimated in a given model structure.

The final step, verification, consists of relating the system to the identified model responses in time or frequency domain to instil confidence in the obtained model. Residual (correlation) analysis and cross-validation tests are generally employed for model validation. The residuals ε are defined as the difference between the model output and measured output. For a perfect model, the residuals should reduce to an uncorrelated sequence e with zero mean and finite variance. Correlation based tests are employed to verify if

$$e(t) = \varepsilon(t) \quad (3.8)$$

This is achieved by verifying if the correlation functions are within the confidence intervals *i.e.*

$$\phi_{\varepsilon\varepsilon}(\kappa) = E[\varepsilon(t - \kappa)\varepsilon(t)] = \delta(\kappa) \quad (3.9)$$

$$\phi_{u\varepsilon}(\kappa) = E[u_{in}(t - \kappa)\varepsilon(t)] = 0 \quad (3.10)$$

where $\phi_{\varepsilon\varepsilon}$ and $\phi_{u_{in}\varepsilon}$ represents the autocorrelation of residuals and cross correlation of residuals and input respectively, u_{in} is the excitation signal to the system and δ is the dirac delta function defined as

$$\delta(\tau) = \begin{cases} 0 & \text{if } \tau \neq 0 \\ 1 & \text{if } \tau = 0 \end{cases}$$

If the cross correlation test in Equation (3.10) is not verified, this means that there is something in the residuals which is originating from the input and has not been properly taken care of by the model and thus the model needs further tuning.

The above-mentioned features of SI are symbolically indicated in Figure 3.13 where $d(t)$ is the external noise or disturbance to the plant. SI theory is well established and the reader is referred to Ljung (1999) for a comprehensive treatment.

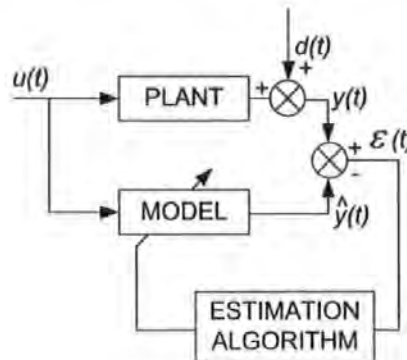


Figure 3.13: The overall system identification procedure.

3.4.2 *Hammerhead* Trials Setup for System Identification

The *Hammerhead* SI experiments (using the configuration shown in Figure 3.7) were designed to obtain the best possible data for model development. Ideally, the requirement is to have a completely noise free data which is impossible in a real world environment. The *Hammerhead* is a low speed AUV that swims at about 2(*knots*). This gives some insight about the sampling period to be chosen. Clearly, too high sampling rate in this case will give no advantage whatsoever. A sampling rate of 1(*Hz*) was thus chosen iteratively which is adequate to obtain ample dynamical information about the system. By the same token, the frequency for the input signal is chosen as 0.1(*Hz*) which was deemed sufficient to excite the interesting modes of the system.

Some common type of excitation signals used in this thesis are uniformly distributed random numbers (UDRN), Figure 3.14(a), pseudo random binary sequence (PRBS), Figure 3.14(b), and its variants such as multistep, Figure 3.14(c), and doublet input, Figure 3.14(d). The multistep inputs are suitable to obtain the step response of the vehicle with various levels of input amplitude. On the other hand, PRBS signal excites the system within a range of frequencies. The response of the vehicle to these excitation signals will be discussed in the subsequent sections. It should be re-emphasised here that the Matlab environment was used for DAQ during all SI trials. However, since Matlab DAQ abilities are limited, a sequential algorithm was developed that acquires data from various on-board sensors progressively rather than simultaneously as demonstrated in the following pseudocodes:

Step 1. send input to the control surface

Step 2. while time < specified duration

read depth sensor → read TCM2 compass → read IMU
end

Step 3. read shaft speed → read GPS

Step 4. go to Step 1

Each of the excitation signal was applied for a specified duration during which sensors data was collected. As shown in the pseudocodes above during the yaw-rudder

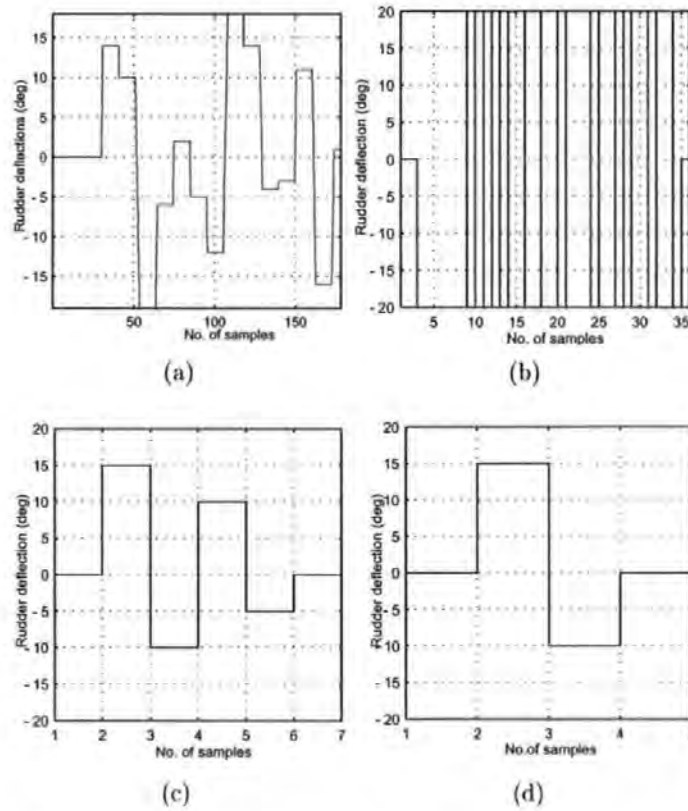


Figure 3.14: (a) UDRN input, (b) 32 length PRBS input, (c) multistep input and (d) doublet input for system identification

channel identification trials, data was acquired from the TCM2, IMU, depth sensor, GPS and a shaft speed encoder. Please note that the GPS and shaft speed encoder have been kept outside of the main loop. This is because the GPS samples at a much slower rate as compared to other sensors and therefore would reduce the overall sampling rate if it was placed inside the loop. The shaft speed encoder was employed here only to check the data validity and to make sure that the vehicle is not slowing down due to low battery power which implies a change in operating condition. The data obtained during this period was therefore successfully separated using the information from the shaft speed encoder and was not used in model identification.

The sampling frequency obtained using this algorithm was $8(Hz)$. The data was resampled afterwards at $1(Hz)$ since this frequency was found adequate to control the *Hammerhead*. Moreover, it also help smoothing the data *i.e.* acts as a low pass

filter. It was observed that during the transmission phase to the on-board actuators, no data could be acquired. This is due to the limitations of Matlab. This problem was circumvented by leaving holes during that interval which represents the missing data. In addition, since there was no feedback from the control surfaces, the transition from one input to the other is approximated as a ramp and appropriate values are inserted. The whole input/output data was later processed and the missing data was interpolated. Figure 3.15(a) shows data set with holes and Figure 3.15(b) depicts the processed data.

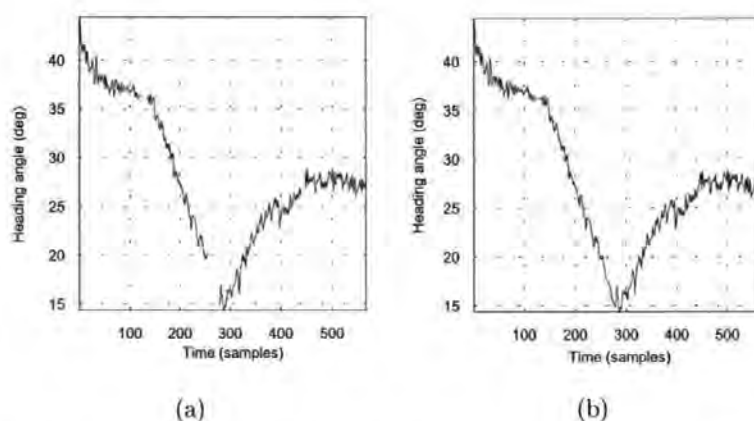


Figure 3.15: (a) Original data set and (b) interpolated data set

3.4.3 Identification Results

The procedure detailed in Section 3.4.2 was employed to acquire the rudder-yaw channel input/output data from the *Hammerhead* necessary for SI. The input to this channel is the rudder deflections and the output is the vehicle's yaw or heading angle. The heading information is available from the TCM2 and IMU, however, the results presented here are the responses obtained from the TCM2 only. In addition, the data sets shown are the original measurements at $8(Hz)$ and has not been filtered or resampled. Three trial results are shown for this channel after a PRBS, a UDRN and a multistep inputs sent to the rudder of the vehicle.

Yaw data analysis

A 32 length PRBS sequence is shown in Figure 3.16(a). The response of *Hammerhead* to this input is also depicted in Figure 3.16(b). Very useful information can be extracted from the heading data. The negative rudder deflection as seen from the figure causes the vehicle to turn clockwise while its opposite in case of a positive rudder angle. The turning radius of *Hammerhead* is an important specification and can also be estimated using this data set.

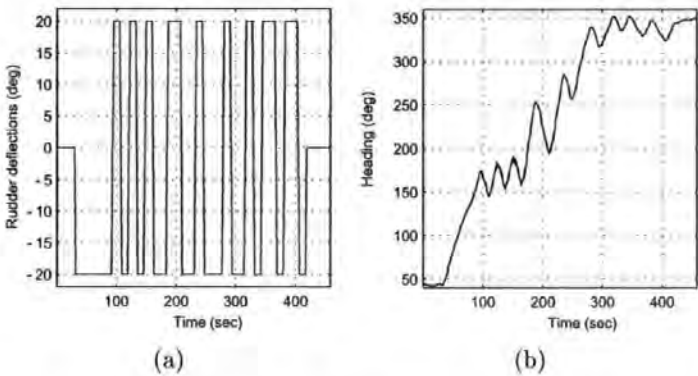


Figure 3.16: (a) PRBS input and (b) PRBS heading

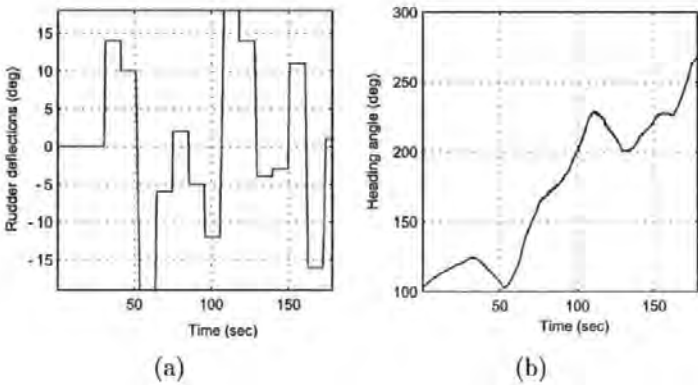


Figure 3.17: (a) UDRN input and (b) UDRN heading

The UDRN input and the vehicle's response is depicted in Figure 3.17(a) and 3.17(b). This provides several step responses of *Hammerhead* for various levels of input. Looking closely at the response plot, the vehicle course changes for a zero rudder deflection.

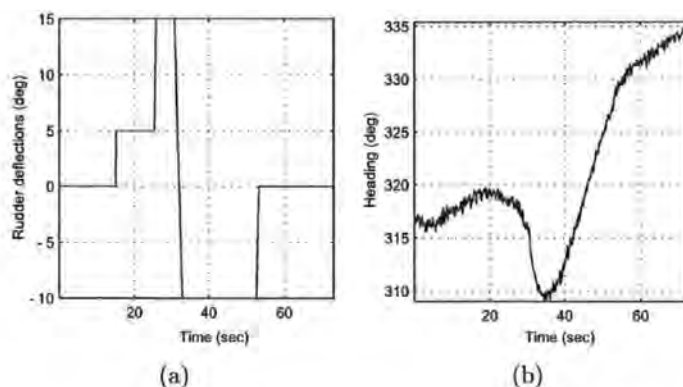


Figure 3.18: (a) multistep input, (b) multistep heading

There could be at least two possible reasons for this type of behaviour. Firstly, the surface currents can push the vehicle without any current compensation or closed loop control. Secondly, since the proximity sensors were not installed at the time of the experiments, the rudder was being initialised by observation and hence the exact position of the rudder was uncertain.

Finally, the multistep input in Figure 3.18(a) was used to excite the *Hammerhead* dynamics and the response was recorded in Figure 3.18(b). Again the vehicle heading changes for zero rudder deflection due to the reasons mentioned above.

3.4.4 Modelling of Rudder-Yaw Channel

Once suitable data sets were selected, attention was turned towards estimating a model that could best replicate the systems behaviour. All available measurements were pre-filtered and resampled at $1(Hz)$ before resuming the work on system identification. For the purpose of this work, first to tenth order state space models were identified. Their properties are shown in Appendix B. Careful observation on each individual model properties leads to the conclusion that state space models with order higher than two bring inconsequential improvement to the quality of the models. More on model properties will be discussed shortly in the next section.

The identified second order state space model is

$$x_{k+1} = \mathbf{A}x_k + \mathbf{B}u_k \quad (3.11)$$

$$z_k = \mathbf{H}x_k \quad (3.12)$$

where x , u and y are the state vector (yaw and delayed yaw), the input (rudder deflection) and the output (yaw). The matrices \mathbf{A} , \mathbf{B} and \mathbf{H} are given by

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -0.98312 & 1.9831 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} -0.003196 \\ -0.0036115 \end{bmatrix}, \mathbf{H} = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

3.4.5 Model Validation

Validation of the estimated model is the final step in an SI process. Various techniques are employed to measure the model quality and its capability to predict accurately the measured response. Correlation tests are performed to validate if all the interesting vehicle dynamics have been captured by the model. On the other hand, cross validation test is performed to gauge the predicting capacity of the model. In this test, data not used for SI is applied to the model and the degree of fit between simulated output and measured response is computed and expressed in per cent.

The correlation tests of the yaw-rudder channel model is performed and the results are shown in Figure 3.19. The cross correlation function (CCF) falls within the 95% confidence intervals indicating that there is no correlation between the input and the residuals. Higher order models could provide autocorrelation (ACF) that falls more inside the 95% confidence interval (refer to Appendix B), however the model identified previously was deemed adequate for further analysis. Next, four cross validation tests are performed for this channel and are shown in Figure 3.20. The simulated outputs as seen from the figures matches reasonably well with the measured outputs. There are some discrepancies though evident from the figures due to the effect of surface currents on different data sets during the trials. A higher order model may not give any significant improvement over the estimated model, as their difference in the degree of fit, as shown in Table 3.6, is infinitesimal. Therefore, robust controllers need to be developed for the selected model which should be able to cope with any discrepancies and disturbances.

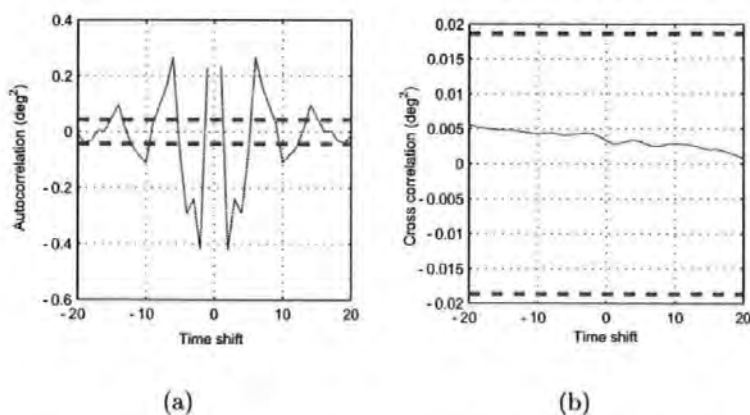


Figure 3.19: (a) Autocorrelation of residuals (b) cross correlation of residuals and the input

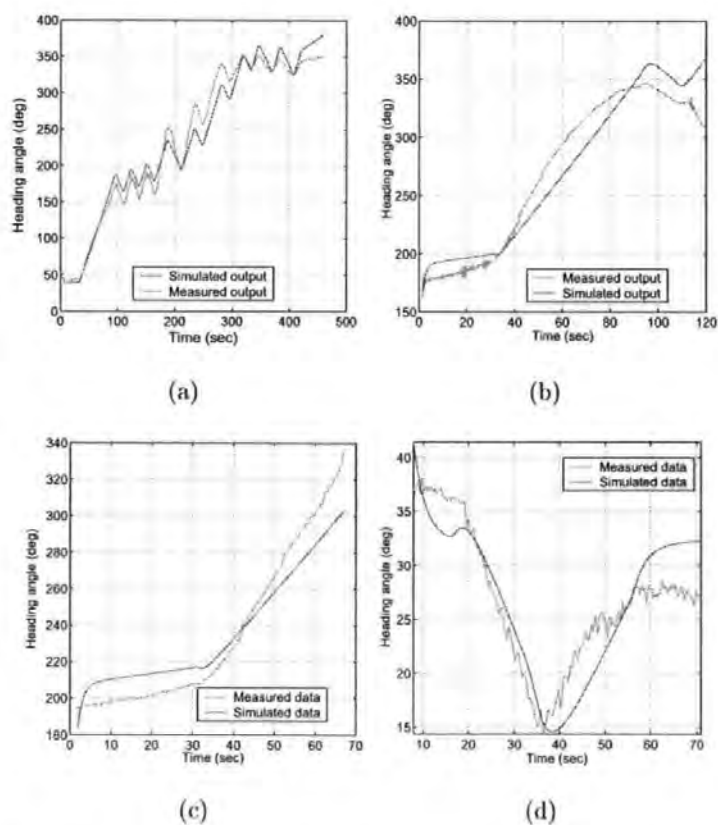


Figure 3.20: Cross validation test for yaw-rudder channel using different sets of data whose fit between the simulated and measured outputs are respectively (a) 83.188%, (b) 70.887%, (c) 70.196% and (d) 51.915%

Order	Fit
2	83.1888%
3	81.9476%
4	81.5526%
5	82.3896%
6	83.5926%
7	82.6636%
8	83.2953%
9	83.2608%
10	83.0287%

Table 3.6: The degree of fit between simulated output and measured response of the identified *Hammerhead* AUV models

3.4.6 Model Analysis

The SI approach is a black box modelling technique meaning that no physical quantities are directly involved in this process in contrast to mathematical modelling. All that is of interest is the cause and effect phenomena and then identifying the black box in between, that can reproduce the measured system output as closely as possible for the same input. Some insight can be gained into systems behaviour by analysing the estimated model. The coefficients of the model, though, do not have any direct physical interpretation but they are vital in studying the nature of the system. The numerator coefficients, for instance, provide the zeros of the plant. For many applications, the plant needs to be minimum phase, *i.e.*, all zeros must lie within the unit circle. The denominator coefficients, on the other hand, determine the pole locations in the z -plane. A pole outside the unit circle indicates an unstable system, therefore the system needs to be stabilised through closed loop control.

The pole zero plot of the rudder-yaw channel is shown in Figure 3.21(a), which clearly shows that this is a minimum phase system. However, the plant is marginally stable due to the presence of a pole on the unit circle. The step response, which is shown in Figure 3.21(b), displays a particular behaviour, which generally belongs to an integrator type of system where the output is produced by integrating the input. This behaviour is mainly caused by the presence of a pole at the axis of the unit circle ($z = 1$ or equivalent to $s = 0$), which along with the step input will form a ramp

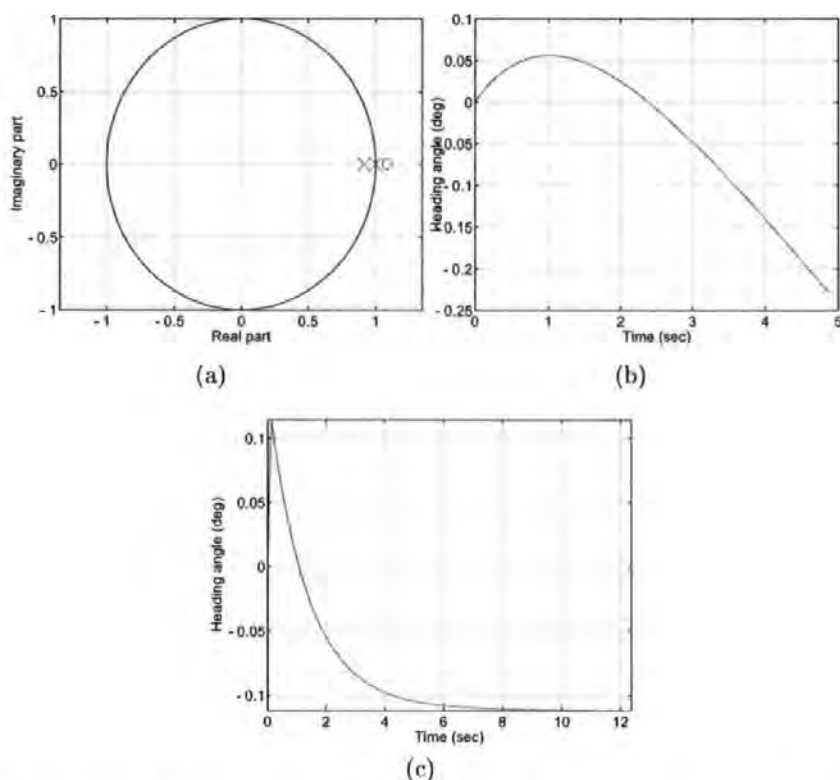


Figure 3.21: (a) Pole zero plot for the rudder-yaw channel model, (b) step response of the rudder-yaw channel model and (c) impulse response of the rudder-yaw channel model

function, whose output magnitude can increase without bound as time progresses. The actual motion of the *Hammerhead* vehicle in response to a step input can be illustrated as a vehicle moving in a circle, whose heading grows from $0(deg)$ to $360(deg)$ for one circular motion and continue growing as the number of complete circular motion (denoted as m) increases. For example for $m = 2$, the heading will grow from $360(deg)$ to $720(deg)$ (i.e., the second $0(deg)$ to $360(deg)$ circular movement). This behaviour is shown in Figure 3.22. Similar types of marginally stable AUV systems (where one pole lie on $z = 1$ or $s = 0$) can be found in the literature. The *Subzero II* (Lea, 1998), the *Aries* (Healey and Lienard, 1993; Healey and Marco, 2001) and the *Phoenix* (Ni, 2001) are to name a few. The step response of the other identified models is given in Appendix B.

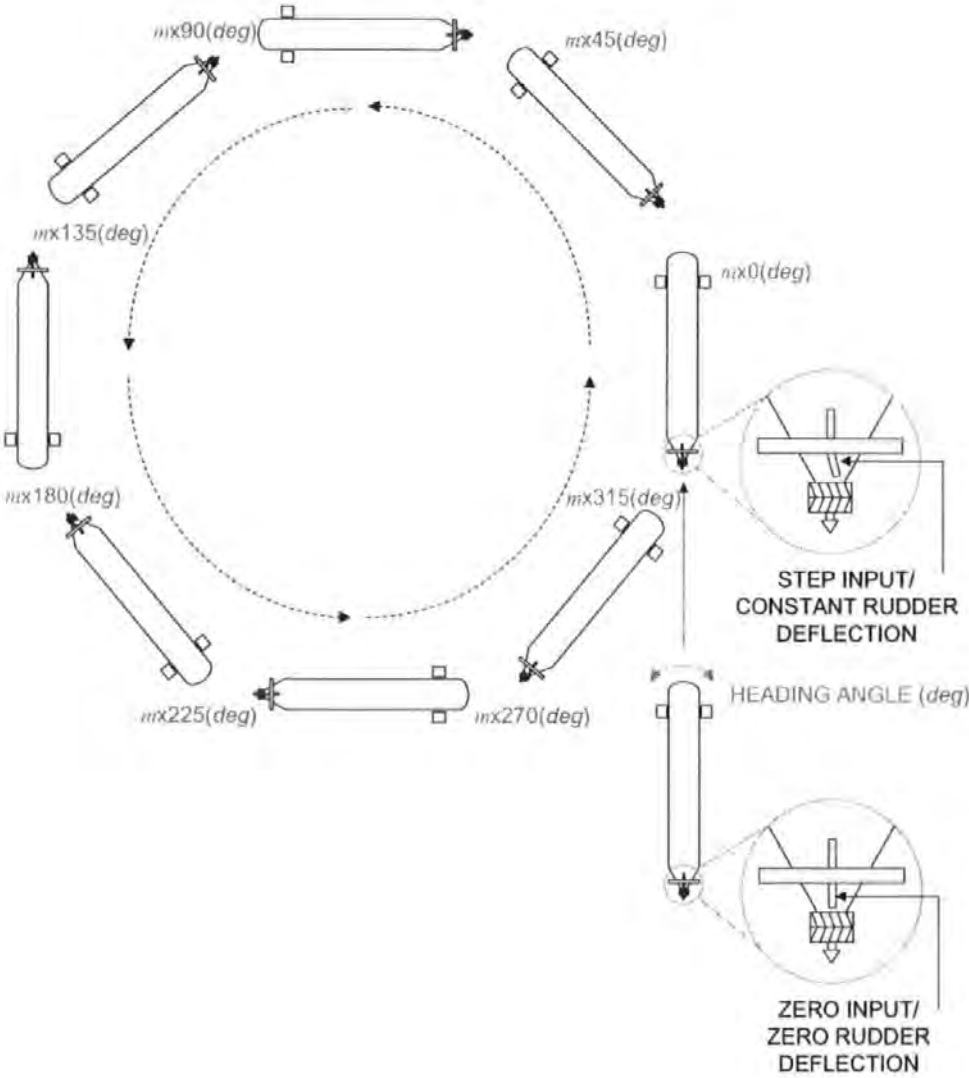


Figure 3.22: m -circular motion of the vehicle, with $m=1,2,\dots$, as a response to a step input

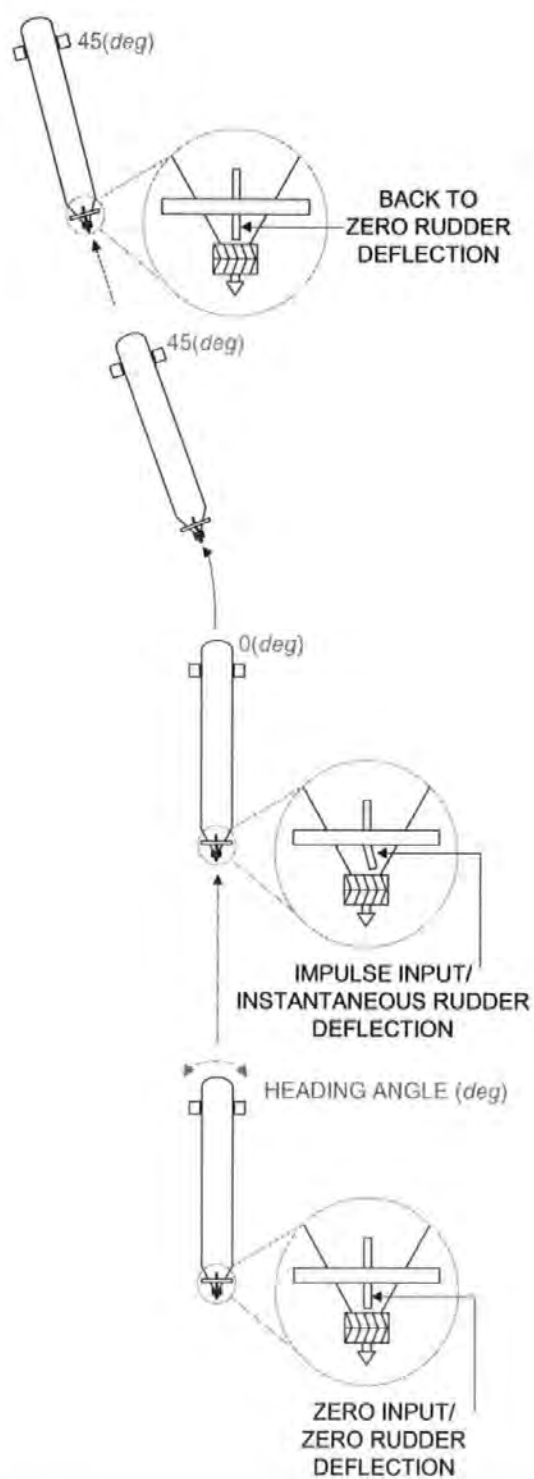


Figure 3.23: Motion of the vehicle as a response to an impulse input

The impulse response of the model is shown in Figure 3.21(c). The actual motion of the *Hammerhead* vehicle can be illustrated as shown in Figure 3.23. The vehicle moves slightly to the left in response to an instantaneous input to the rudder and maintains this heading as the rudder returns to its original position. The complete impulse response of the higher order systems is given in Appendix B.

3.5 CONCLUDING REMARKS

This chapter began with a discussion on the evolution of the structure of the *Hammerhead* AUV. It was developed on the structure of a DMT, and with subsequent modification and addition of hardware and sensor, the DMT was transformed to a semi-autonomous underwater vehicle. The hardware setup at this stage was similar to that in ROVs. Here sensor data and control action are still transferred through a tether and processed in an off-board CPU. Further development eliminated the need for a tether, and the vehicle was able to operate in a full autonomous mode. In developing an integrated navigation, guidance and control, several full scale trials were conducted for the purpose of obtaining a dynamic model of the vehicle. Taking into consideration the physical characteristics and the facilities available, it was decided to adopt SI approaches. The details of the sensors used for this purpose were discussed. Chapter 4 and 5 present respectively the results from the simulation and pseudo real-time implementation of the proposed adaptive Kalman filtering. Both employ a dynamic model of the vehicle developed using the SI approaches discussed in this chapter. Pseudo real-time implementation of the proposed techniques in Chapter 5 also makes use a set of data collected using the hardware setup described herein.

CHAPTER 4

FUZZY KALMAN FILTER MULTIOBJECTIVE GENETIC ALGORITHM: SIMULATION

4.1 INTRODUCTION

It is clear from the discussion in Chapter 2 that to achieve truly autonomous behaviour, an AUV must be able to locate itself accurately during an operating scenario using only its on-board sensors. In the past, fusing the data produced by different navigation sensors through the use of a KF has been a popular method for most navigation system mechanizations (Kayton, 1997), and this is particularly true for AUVs (Loebis *et al.*, 2002). Even in conditions where the KF does not perform very well it is often used as a benchmark for more custom-made and specialised filters. This is due to the properties of the KF which makes it very useful and easy to implement. In its basic form it allows measurement of different dimensions, observing different subsets of the system and arriving at different times and frequencies to be fused. Furthermore, it allows both measurement and process equations to be time variant and it yields the optimal state estimate when operating under Gaussian and linear conditions. However, a significant difficulty in designing a KF can often be traced to incomplete *a priori* knowledge of the process covariance matrix \mathbf{Q} and measurement noise covariance matrix \mathbf{R} . In most practical applications, these matrices are initially estimated or even unknown. Several examples of these applications are given in Chapter 2. The problem here is that the optimality of the estimation algorithm in the KF setting is closely connected to the quality of *a priori* information about the process and measurement noise (Mehra, 1970; Mehra, 1971). It has been shown that insufficiently known *a priori* filter statistics can reduce the precision of the estimated filter states or introduces biases to their estimates. In addition, incorrect *a priori* information can lead to practical divergence of the filter (Fitzgerald, 1971). From

the aforementioned it may be argued that the conventional KF with fixed \mathbf{R} and/or \mathbf{Q} should be replaced by an adaptive estimation formulation and this is the main discussion of the next sections. To maintain the clarity of the forthcoming discussion on this subject, a hands-on reference to the formulae in the basic discrete Kalman filtering algorithm is provided in Appendix D. Readers who are interested on the details of the derivation are referred to Gelb (1989), and Brown and Hwang (1997).

4.2 THE ADAPTIVE KALMAN FILTER ALGORITHM

Adaptive Kalman filtering has attracted an enormous research interest and accordingly a number of papers have been published in this area. The two major approaches that have been proposed for adaptive Kalman filtering are multiple model adaptive estimation (MMAE) (Magill, 1965; Maybeck and Hanlon, 1995; Wheaton and Maybeck, 1995; Eide, 1996; Chaer *et al.*, 1997; Chaer *et al.*, 1998; Schiller and Maybeck, 1997; Hanlon and Maybeck, 1998; Hanlon and Maybeck, 2000a; Hanlon and Maybeck, 2000b; Vazquez and Maybeck, 2004) and innovation adaptive estimation (IAE) (Jazwinski, 1969; Mehra, 1970; Mehra, 1971; Mehra, 1972; Boozer and McDaniel-Jr., 1972; Tsai and Kurz, 1983; Alspach, 1973; Kumar *et al.*, 1991; Chen and Chui, 1990; Xia *et al.*, 1994; King *et al.*, 1994; Liang *et al.*, 2004). Although the implementation of these approaches are quite different, they both share the same concept of utilising new statistical information obtained from the innovation (or residual) sequence. In both cases, the innovation Inn_k at sample time k is the difference between the real-measurement z_k , received by the filter and its estimated (predicted) value \hat{z}_k . The predicted measurement is the projection of the filter predicted states \hat{x}_k^- onto the measurement space through the measurement design matrix \mathbf{H}_k . Innovation represents additional information available to the filter as a result of the new measurement z_k . The occurrence of data with statistics different from the *a priori* information will first show up in the innovation vector. For this reason the innovation sequence represents the information content in the new observation and is considered the most relevant source of information to the filter adaptation. Interested readers can refer to Kailath (1968a), Kailath (1968b), and Kailath (1970) for a more detailed discussion on the innovation sequence and its use in linear filter theory.

In the MMAE approach, a bank of Kalman filters run in parallel (Magill, 1965; Maybeck and Hanlon, 1995; Wheaton and Maybeck, 1995; Eide, 1996; Schiller and Maybeck, 1997; Hanlon and Maybeck, 1998; Hanlon and Maybeck, 2000a; Hanlon and Maybeck, 2000b; Vazquez and Maybeck, 2004) or with a gating algorithm (Chaer *et al.*, 1997; Chaer *et al.*, 1998) under a different model for the statistical filter information matrices, *i.e.* \mathbf{Q} and \mathbf{R} . In the IAE approach, the \mathbf{Q} and \mathbf{R} themselves are adapted as measurements evolve with time. In this chapter, the IAE approach pioneered by Jazwinski (1969) and popularized by (Mehra, 1970; Mehra, 1971; Mehra, 1972) coupled with fuzzy logic techniques described in Appendix E is used to adjust the \mathbf{R} matrix of the KF.

The fuzzy logic membership functions for the IAE approach are initially established by a combination of knowledge, experience and observation and may thus not be optimal. Additionally, fine-tuning of its performance is still a matter of trial and error. Many studies have shown that genetic algorithms (described in Appendix F) have the ability to find fuzzy membership functions closer to optimal solutions and may be made to implement self-tuning and adaptive schemes (Córdoba *et al.*, 1998). However, the work in this thesis is the first known use of the multiobjective genetic algorithm proposed by Fonseca and Fleming (1995) (also described in Appendix F) for the optimization of the membership function of a fuzzy system used for the adaptation of an assumed KF measurement noise characteristic. Thus, this is considered as the major contribution of this particular study in relation to AUV technology.

4.3 FUZZY KALMAN FILTER

In this section, an on-line innovation-based adaptive scheme of the KF to adjust the \mathbf{R} matrix employing the principles of fuzzy logic is presented. The fuzzy logic is chosen mainly because of its simplicity and closeness to human reasoning. These enable a satisfactory performance being developed empirically in practice without complicated mathematics. These have motivated the interest in the topic, as testified by related

articles which have been appearing in the literature (Loebis *et al.*, 2004b; Escamilla-Ambrosio and Mort, 2001; Jetto *et al.*, 1999; Kobayashi *et al.*, 1998).

The fuzzy logic Kalman filter (FKF) proposed herein is based on the IAE approach using a technique known as covariance-matching (Mehra, 1970). The basic idea behind the technique is to make the actual value of the covariance of the innovation sequences match its theoretical value.

The actual covariance is defined as an approximation of the Inn_k sample covariance through averaging inside a moving estimation window of size M (Mohamed and Schwarz, 1999) which takes the following form:

$$\hat{\mathbf{C}}_{Inn_k} = \frac{1}{M} \sum_{j=j_0}^k Inn_k \cdot Inn_k^T \quad (4.1)$$

where $j_0 = k - M + 1$ is the first sample inside the estimation window. An empirical heuristic experiment is conducted to choose the window size M that is adequate to capture the dynamic of the Inn_k actual covariance. From experimentation it was found that a good size for the moving window in Equation (4.1) used in this thesis is 15. The value of M is dependent on the dynamic of the Inn_k and therefore can vary for different types of applications.

The theoretical covariance of the innovation sequence is defined as (Mehra, 1970):

$$\mathbf{S}_k = \mathbf{H}_k \cdot \mathbf{P}_k^- \cdot \mathbf{H}_k^T + \mathbf{R}_k \quad (4.2)$$

The logic of the adaptation algorithm using covariance matching technique can be qualitatively described as follows. If the actual covariance value $\hat{\mathbf{C}}_{Inn_k}$ is observed, whose value is within the range predicted by theory \mathbf{S}_k and the difference is very near to zero, this indicates that both covariances match almost perfectly and only a small change is needed to be made on the value of \mathbf{R} . If the actual covariance is greater than its theoretical value, the value of \mathbf{R} should be decreased. On the contrary, if $\hat{\mathbf{C}}_{Inn_k}$ is less than \mathbf{S}_k , the value of \mathbf{R} should be increased. This adjustment mechanism lends

itself very well to being dealt with using a fuzzy-logic approach (Escamilla-Ambrosio and Mort, 2001):

$$\text{IF } \langle \text{antecedent} \rangle \text{ THEN } \langle \text{consequent} \rangle \quad (4.3)$$

To implement the above covariance matching technique using the fuzzy logic approach, a new variable called \mathbf{delta}_k , is defined to detect the discrepancy between $\hat{\mathbf{C}}_{Inn_k}$ and \mathbf{S}_k . The following fuzzy rules of the kind of Equation (4.3) are used:

$$\text{IF } \langle \mathbf{delta}_k \cong 0 \rangle \text{ THEN } \langle \mathbf{R}_k \text{ is unchanged} \rangle \quad (4.4)$$

$$\text{IF } \langle \mathbf{delta}_k > 0 \rangle \text{ THEN } \langle \mathbf{R}_k \text{ is decreased} \rangle \quad (4.5)$$

$$\text{IF } \langle \mathbf{delta}_k < 0 \rangle \text{ THEN } \langle \mathbf{R}_k \text{ is increased} \rangle \quad (4.6)$$

Thus \mathbf{R} is adjusted according to,

$$\mathbf{R}_k = \mathbf{R}_{k-1} + \Delta \mathbf{R}_k \quad (4.7)$$

where $\Delta \mathbf{R}_k$ is added or subtracted from \mathbf{R} at each instant of time. Here \mathbf{delta}_k is the input to the fuzzy inference system (FIS) and $\Delta \mathbf{R}_k$ is the output.

On the basis of the above adaptation hypothesis, the FIS can be implemented using three fuzzy sets for \mathbf{delta}_k ; $N = \text{Negative}$, $Z = \text{Zero}$ and $P = \text{Positive}$. For $\Delta \mathbf{R}_k$ the fuzzy sets are specified as $I = \text{Increase}$, $M = \text{Maintain}$ and $D = \text{Decrease}$. The membership functions of these fuzzy sets which are first designed using a heuristic approach are shown in Figure 4.1(a) and 4.1(b).

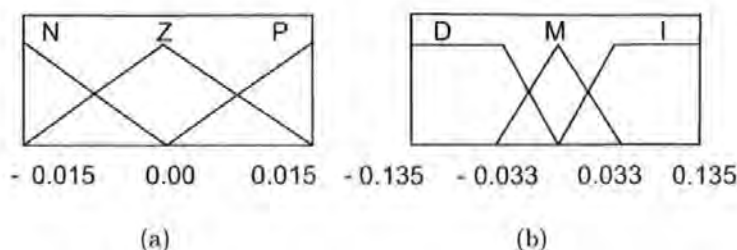


Figure 4.1: Membership function of (a) \mathbf{delta}_k and (b) $\Delta \mathbf{R}_k$

4.4 FUZZY LOGIC OBSERVER

To monitor the performance of a FKF, another FIS called the fuzzy logic observer (FLO) (Escamilla-Ambrosio and Mort, 2001) is used. The FLO assigns a weight or degree of confidence denoted as c_k , a number on the interval $[0,1]$, to the FKF state estimate. The FLO is implemented using two inputs: the values of $|\mathbf{delta}_k|$ and \mathbf{R}_k . The membership functions of these variables are shown Figure 4.2(a) and 4.2(b).

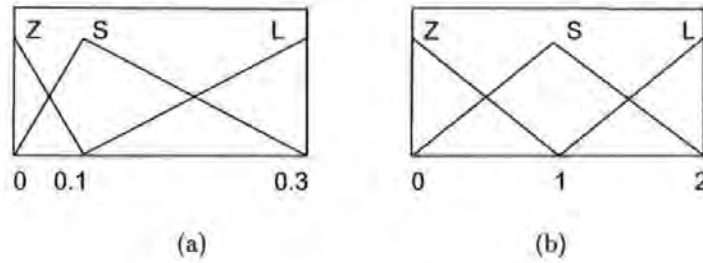


Figure 4.2: Membership function of (a) $|\mathbf{delta}_k|$ and (b) \mathbf{R}_k

The fuzzy labels for the membership functions: $Z = Zero$, $S = Small$ and $L = Large$. Three fuzzy singletons are defined for the output c_k and are labelled as $G = Good$, $AV = Average$ and $P = Poor$ with values 1, 0.5 and 0 respectively. The basic heuristic hypothesis for the FLO is as follows: if the value of $|\mathbf{delta}_k|$ is near to zero and the value of \mathbf{R}_k is near to zero, then the FKF works almost perfectly and the state estimate of the FKF is assigned a weight near 1. On the contrary if one or both of these values increases far from zero, it means that the FKF performance is degrading and the FLO assigns a weight near 0. Table 4.1 gives the complete fuzzy rule base of each FLO.

\mathbf{R}_k	Z	S	L
$ \mathbf{delta}_k $			
Z	G	G	AV
S	G	AV	P
L	AV	P	P

Table 4.1: Fuzzy rule based FLO

4.5 FUZZY MEMBERSHIP FUNCTIONS OPTIMIZATION

GAs in single- and multiobjective mode are used here to optimize the membership functions of the FKF. To translate the FKF membership functions to a representation useful as genetic material, they are parameterised with real-valued variables. Each of these variables constitutes a gene of the chromosomes (concurrent multiple search points (refer to Appendix F)) for the MOGA. Boundaries of chromosomes are required for the creation of chromosomes in the right limits so that the MOGA is not misled to some other area of search space. The technique adopted in this thesis is to define the boundaries of the output membership functions according to the furthest points and the crossover points of two adjacent membership functions. In other words, the boundaries of FKF consist of three real-valued chromosomes (*Chs*), as in Figure 4.3. The trapezoidal membership functions' two furthest points, -0.135 (D_1), -0.135 (D_2)

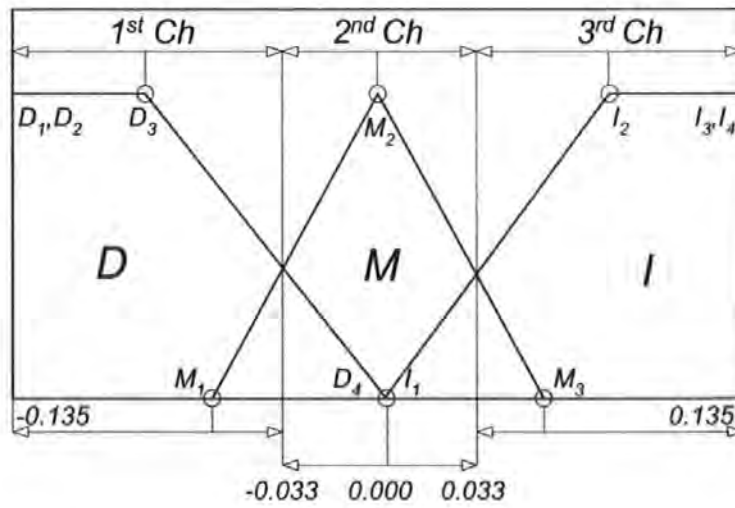


Figure 4.3: Membership function and boundaries of \mathbf{R}_k

and 0.135 (I_3), 0.135 (I_4) of FKF, remain the same in the GA's description to allow a similar representation as the fuzzy system's definition. As can be seen from Figure 4.3, D_3 and M_1 can change value in the 1st *Ch* boundary, D_4 , M_2 and I_1 in the 2nd *Ch* boundary, and finally, M_3 and I_2 in 3rd *Ch*. Table 4.2 shows the encoding used for optimization of the membership functions.

Limit	Parameter		
	D_3, M_1	D_4, M_2, I_1	M_3, I_2
Upper Limit	-0.135	-0.033	0.033
Lower Limit	-0.033	0.033	0.135

Table 4.2: FKF boundaries

4.6 FUSION OF INS SENSOR DATA

In this section, the FKF technique is applied to maintain the optimality of an AUV heading estimation process. The FLO, will then be used subsequently to fuse the estimated heading values. To this end, the *Hammerhead* dynamic model in Equation (4.8) obtained using system identification procedures discussed in the previous chapter (with further details given in Naeem (2004)), is employed. It is assumed in this model that the forward velocity of the vehicle is constant at $1(m/sec)$ and the vehicle is not at an angle of roll and pitch.

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -0.98312 & 1.9831 \end{bmatrix}; \mathbf{B} = \begin{bmatrix} -0.0031961 \\ -0.0036115 \end{bmatrix}; \mathbf{H} = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad (4.8)$$

Here yaw and delayed yaw as the component of the states. The w and v are both zero mean white noise for the process and measurement models respectively and input to the system is δ_r (rudder deflection). The initial conditions are:

$$x_0 = \begin{bmatrix} 0(rad) \\ 0(rad) \end{bmatrix}; \mathbf{P}_0 = \begin{bmatrix} 0.01(rad)^2 & 0 \\ 0 & 0.01(rad)^2 \end{bmatrix} \quad (4.9)$$

and \mathbf{Q}_k is made constant as

$$\mathbf{Q}_k = \begin{bmatrix} 0(rad)^2 & 0 \\ 0 & 0.1725 \times 10^{-7}(rad)^2 \end{bmatrix} \quad (4.10)$$

The values of \mathbf{P}_0 and \mathbf{Q}_k are determined heuristically. In real-time applications, the \mathbf{Q}_k values are dependent on temporal and spatial variations in the environment such as sea conditions, ocean current, and local magnetic variations and therefore, appropriate adjustments to the initial values of \mathbf{Q} also need to be undertaken. However,

given the fact that the *Hammerhead* AUV mostly operates in a stable environment, the problem with the **Q** adjustment is reserved for future work.

To the *Hammerhead* model described in Equation (4.8) a sinusoidal input was applied. Four yaw sensors with different noise characteristics are considered to measure the response of the vehicle. The actual value of **R** for each yaw sensor is assumed unknown but its initial value is selected as $0.01(\text{rad})^2$. The FKF algorithm optimized using a MOGA with parameters shown in Table 4.3 was then implemented with the trade-off graph shown in Figure 4.4 and simulation results shown in the next section. The MOGA parameters used herein were chosen heuristically after exhaustive tests and no significant improvement can be achieved by adjusting these values.

Parameters	Values
Number of objective functions	5
Number of generation	200
Number of individual per generation	25
Generation gap in selection operation	0.95
Rate in rate in recombination operation	0.8
Rate in mutation operation	0.09

Table 4.3: MOGA parameters

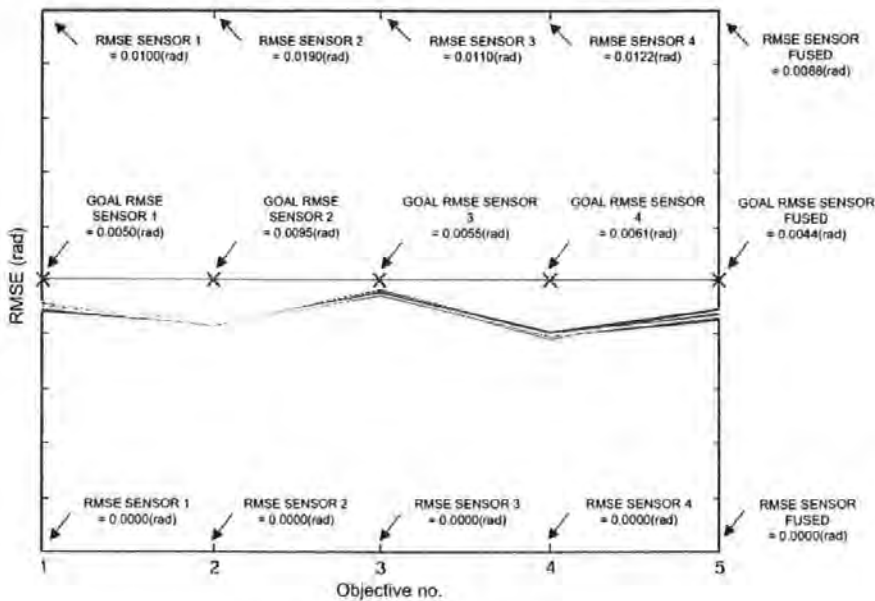


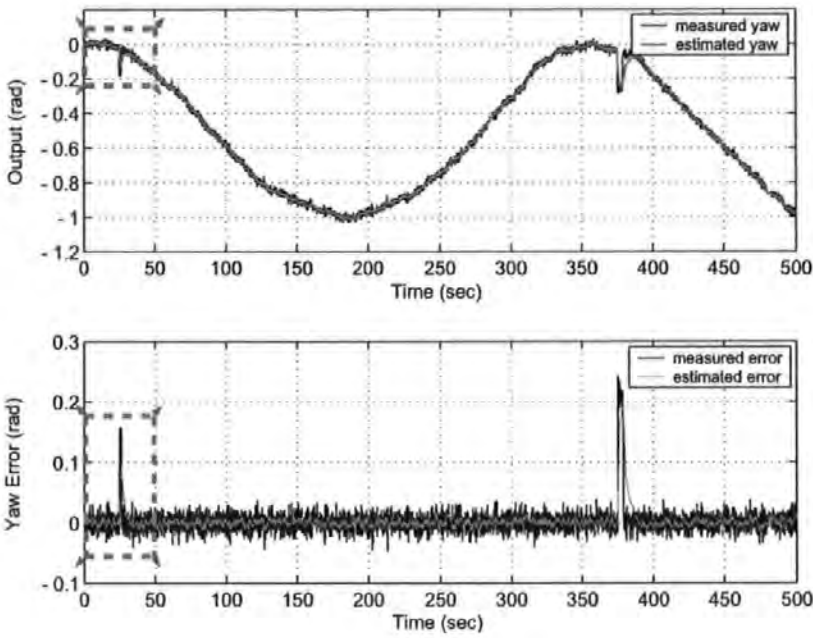
Figure 4.4: Trade-off graph for FKF search

4.7 SIMULATION RESULTS

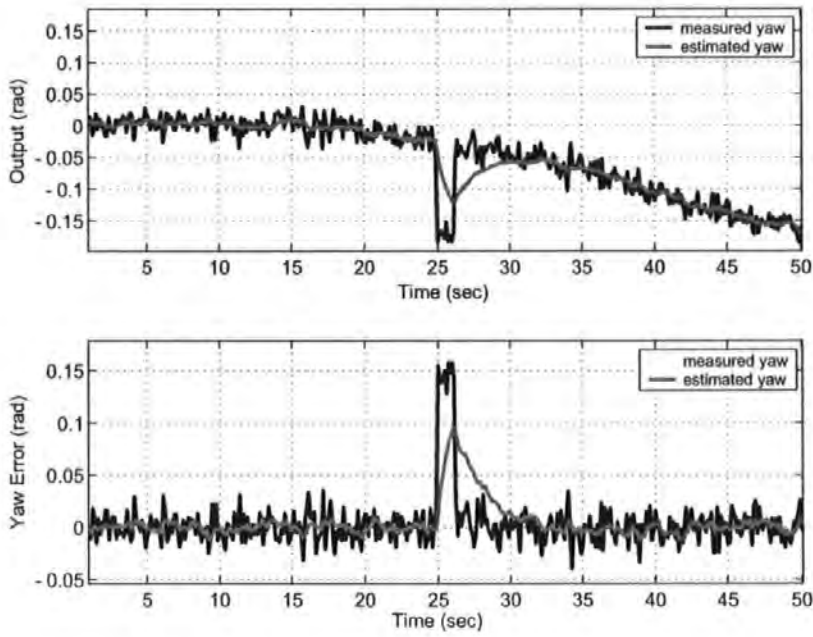
Figure 4.4 and 4.5 are the simulation results showing the response of the AUV observed by sensors with constant Gaussian noise, while Figure 4.6 and 4.7 by sensors with uniform noise increasing and decreasing with time respectively. These types of noise are included in the simulation to demonstrate the effectiveness of the proposed adaptation mechanism in general. A possible real-time scenario that can result in the noise with the characteristic shown in Figure 4.6 is the third yaw sensor located in close proximity to an electronic hardware such as the vehicle's propeller DC motor whose internal temperature increases with time and affects the sensor ambient temperature. A similar scenario can also occur when the fourth yaw sensor (Figure 4.7) is located in close proximity to another vehicle's electronic hardware such as the laser unit used in the VNS whose initial internal temperature is high and settles down after sometime. Figure 4.8 shows the values of \mathbf{R} after the FKF has been run. Figure 4.4 and 4.7 also show several peaks in the simulations of sensors 1 and 4. These are to indicate faults in the sensor. There are two types of fault defined in this simulation work, transient and persistent faults (Escamilla-Ambrosio and Mort, 2001). A transient fault happens when the sensor output increases abruptly for only a sample period of time. Persistent faults occur when the transient faults persist for a period of time. Consequently, the peaks in sensor-1 and sensor-4 simulations show the persistent and transient faults respectively. Figure 4.4(b) and Figure 4.7(b) provide a closer look on the indicated areas in Figure 4.4(a) and 4.7(a) respectively. It is clear in both cases that the algorithms have detected faults in the system and appropriate actions have been undertaken to recover the signals. Direct observation on Figure 4.4 - 4.7 shows how the proposed method has significantly reduced the level of error in the system. To fuse the estimated yaw, a centre of gravity method is used,

$$z_k = \frac{\sum_{i=1}^n \hat{z}_{k_i} c_{k_i}}{\sum_{i=1}^n c_{k_i}} \quad (4.11)$$

where \hat{z}_{k_i} is the output of the i -th FKF ($i=1,2,3,4$) and c_{k_i} is the respective weight at instant time k .



(a)



(b)

Figure 4.4: (a) Measured and estimated yaw output and error of sensor-1, (b) measured and recovered yaw output and error of sensor-1

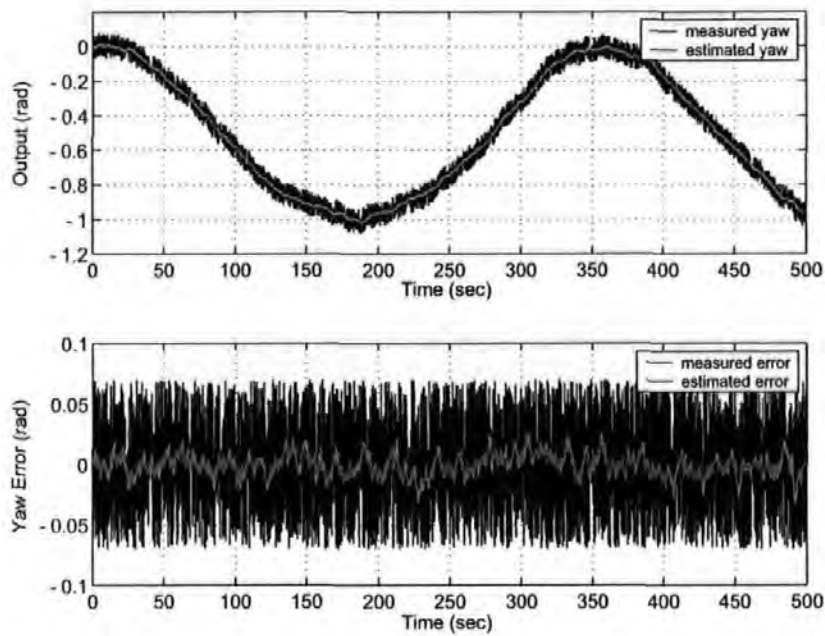


Figure 4.5: Measured and estimated yaw output and error of sensor-2

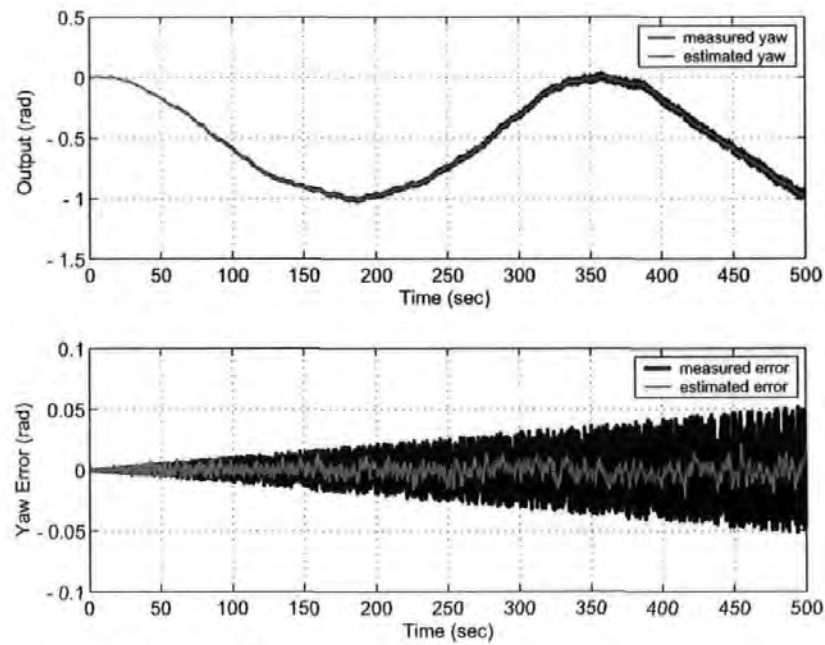
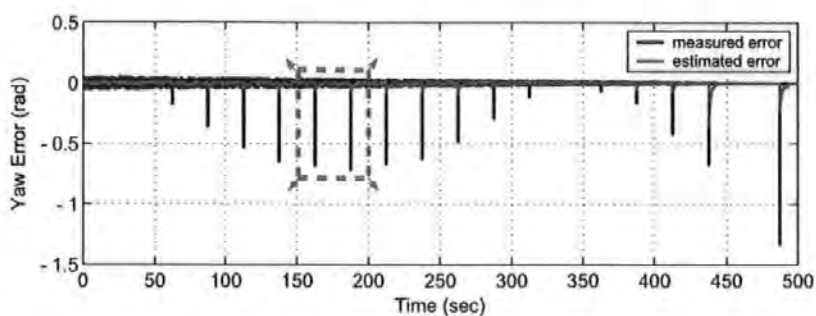
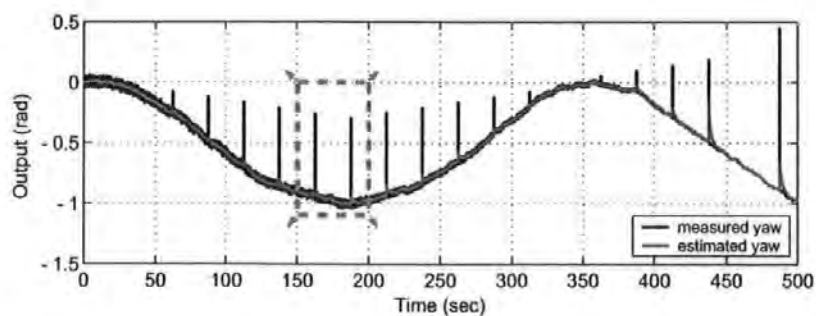
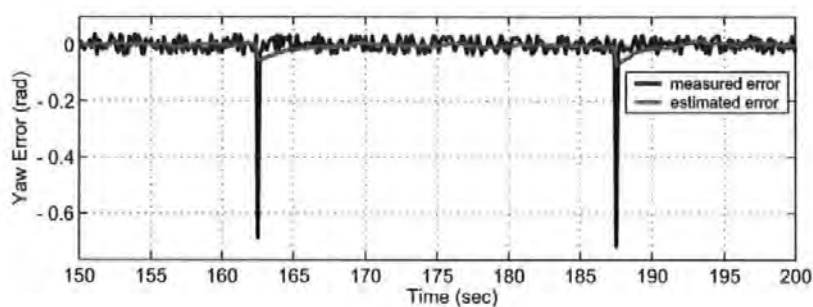
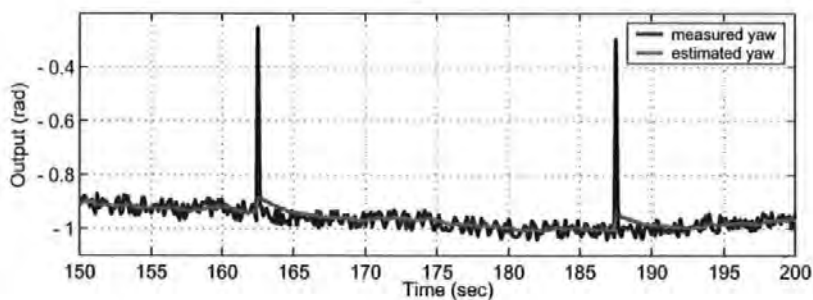


Figure 4.6: Measured and estimated yaw output and error of sensor-3



(a)



(b)

Figure 4.7: (a) Measured and estimated yaw output and error of sensor 4, (b) measured and recovered yaw output and error of sensor-4

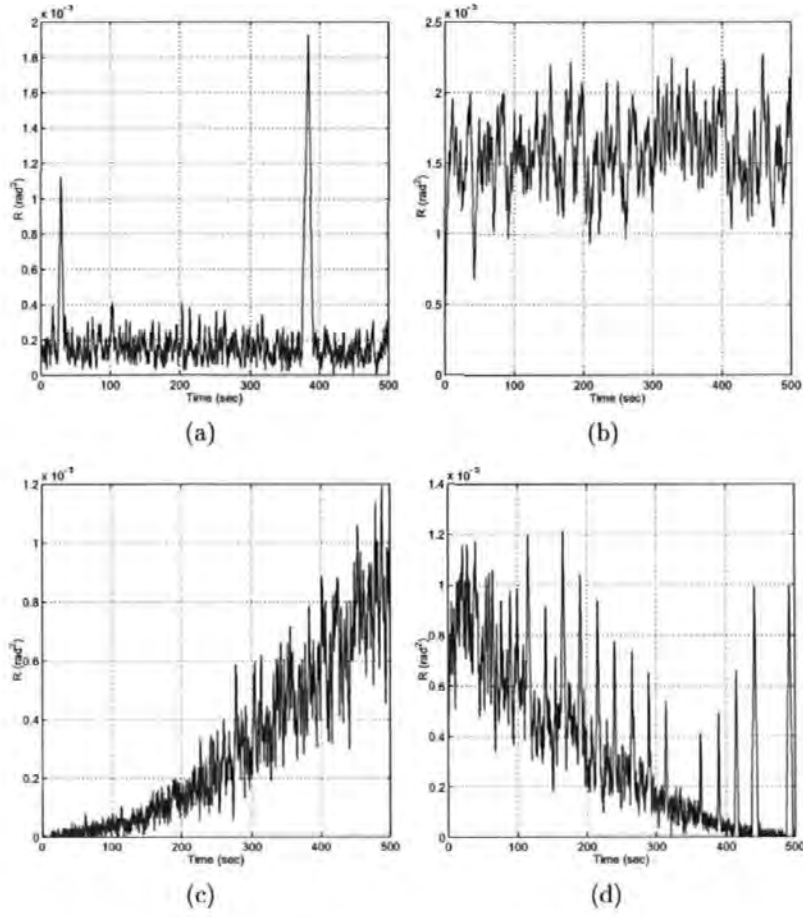


Figure 4.8: R time series for sensor (a) 1, (b) 2, (c) 3 and (d) 4

Finally, the following performance measure are adopted for comparison purposes,

$$J_{zv} = \sqrt{\frac{1}{n} \sum_{k=1}^n (za_k - z_k)^2} \quad (4.12)$$

$$J_{ze} = \sqrt{\frac{1}{n} \sum_{k=1}^n (za_k - \hat{z}_k)^2} \quad (4.13)$$

where za_k is the actual value of the yaw, z_k is the measured yaw, \hat{z}_k is the estimated yaw at an instant of time k and n = number of samples. Table 4.4 shows the comparison of performance of each individual measured sensor output with the one obtained using standard FKF and FKF optimized using MOGA (FKF-MOGA) respectively.

Sensor	$J_{zv}(rad)$	$J_{ze}(rad)$			
		GA	MOGA		
			1 st	2 nd	3 rd
Sensor-1	0.02669	0.01910	0.00469	0.00466	0.00470
Sensor-2	0.03998	0.00876	0.00877	0.00877	0.00876
Sensor-3	0.01733	0.00494	0.00498	0.00501	0.00500
Sensor-4	0.02093	0.00507	0.00487	0.00487	0.00487
Fused		0.00350	0.00372	0.00373	0.00373

Table 4.4: Comparison of performance

A further comparison is also made between the performance of the fused sensor outputs obtained using both FKF schemes. A first look on the table shows that J_{ze} of each individual sensor always outperforms J_{zv} . Most importantly is that every single J_{ze} of the fused sensor output from both FKF schemes outperforms its individual sensor counterpart. It is also clear that the MOGA optimization techniques have significantly increased the performance of sensor-1 as indicated by the corresponding J_{ze} -s. The FKF-MOGA J_{ze} -s of sensor-4 also show a similar trend of improvement, although not as good as the ones produced by sensor-1. The J_{ze} -s of sensor-2 and sensor-3 in the FKF-MOGA case are slightly inferior compared to those in the standard FKF case. This stems from the non-dominated nature of MOGA solutions for which an improvement in one objective will lead to a degradation in one or more other objectives as discussed in Appendix F.

4.8 GPS/INS SURFACE (2D) NAVIGATION

Here, the fused estimated yaw obtained previously is treated as a single imaginary yaw sensor and used by other INS sensors to transform data from a body co-ordinate to a geographical (North-East-Down/NED) co-ordinate frame (see Appendix G for details) where integration with converted GPS data is performed using a combination of FKF and EKF techniques and can be referred to as fuzzy extended Kalman filter(FEKF). Figure 4.9 shows this relationship and serves as an overall representation of the algorithms that have been discussed so far.

A continuous time model of the vehicle motion appropriate to this problem is taken

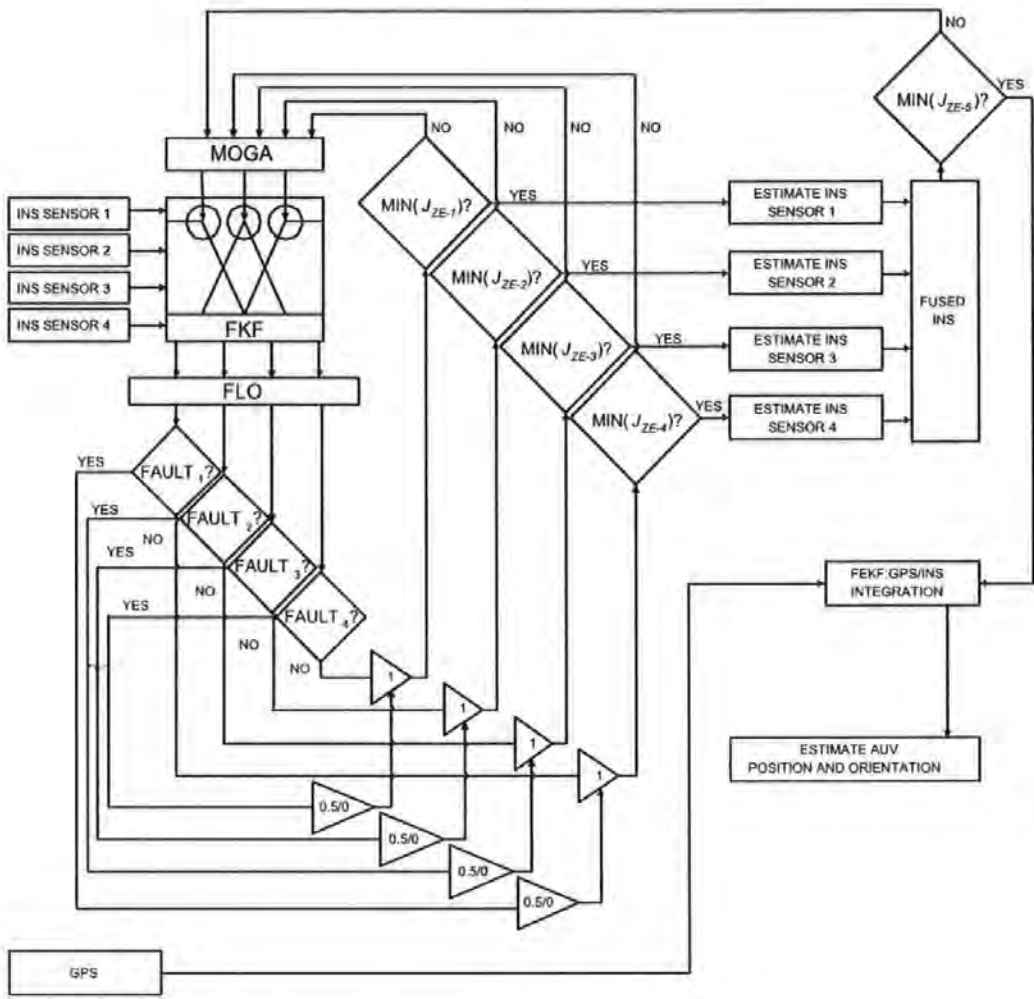


Figure 4.9: Block diagram of GPS/INS using FKF and FEKF

to be

$$\dot{X}(t) = \mathbf{F}(X(t)) + W(t) \quad (4.14)$$

$$Z(t) = \mathbf{H}(X(t)) + V(t) \quad (4.15)$$

Denoted by $X(t) = [X_{NED}(t) \ Y_{NED}(t) \ \psi_{im}(t) \ r(t) \ u(t) \ v(t)]^T$ are the model states. $X_{NED}(t)$ and $Y_{NED}(t)$ are the longitude and latitude of the AUV position converted from deg-min-sec in an Earth-centered Earth-fixed co-ordinate frame into metres in the NED co-ordinate frame, $\psi_{im}(t)$ is the yaw angle obtained from the imaginary yaw sensor, $r(t)$ is yaw rate, $u(t)$ and $v(t)$ are the surge and sway velocity respectively. In this system model, $\mathbf{F}(\cdot)$ and $\mathbf{H}(\cdot)$ are both continuous function, continuously differentiable in $X(t)$. The $W(t)$ and $V(t)$ are both zero mean white noise

for the system and measurement models respectively. The model states are related through the following kinematically based set of functions ($\mathbf{F}(X(t))$) in Equation (4.14):

$$\dot{u}(t) = 0 \quad (4.16)$$

$$\dot{v}(t) = 0 \quad (4.17)$$

$$\dot{\psi}_{im}(t) = r(t) \quad (4.18)$$

$$\dot{r}(t) = 0 \quad (4.19)$$

$$\dot{X}_{NED}(t) = u(t) \cos \psi_{im}(t) - v(t) \sin \psi_{im}(t) \quad (4.20)$$

$$\dot{Y}_{NED}(t) = u(t) \sin \psi_{im}(t) + v(t) \cos \psi_{im}(t) \quad (4.21)$$

The output measurements are related through the states by the identity matrix ($\mathbf{H}(X(t))$). To obtain an EKF with an effective state prediction equation in a simple form, the continuous time model of Equations (4.16) - (4.21) have been linearised about the current state estimates, producing:

$$\mathbf{F}_{lin}(t) =$$

$$\begin{bmatrix} 0 & 0 & -u(t) \sin \psi_{im}(t) - v(t) \cos \psi_{im}(t) & 0 & \cos \psi_{im}(t) & -\sin \psi_{im}(t) \\ 0 & 0 & u(t) \cos \psi_{im}(t) - v(t) \sin \psi_{im}(t) & 0 & \sin \psi_{im}(t) & \cos \psi_{im}(t) \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.22)$$

and $\mathbf{H}_{lin}(t)$ is an identity matrix (where \mathbf{F}_{lin} and \mathbf{H}_{lin} equivalent to \mathbf{A} and \mathbf{H}). Subsequent discretisation with period $T = 0.125(sec)$ of the linearised model results in an EKF algorithm (see Appendix D for details) only this time the \mathbf{A} is updated at every iteration. The initial conditions are:

$\mathbf{P}_0 =$

$$\begin{bmatrix} 0.01(m)^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.01(m)^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01(rad)^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01(rad/sec)^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.01(m/sec)^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.01(m/sec)^2 \end{bmatrix} \quad (4.23)$$

and \mathbf{Q} is made constant as

$$\begin{bmatrix} 10(m)^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10(m)^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.000001(rad)^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01(rad/sec)^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.01(m/sec)^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.01(m/sec)^2 \end{bmatrix} \quad (4.24)$$

As in the case of fusion of INS sensor data discussed previously, the values of \mathbf{P}_0 and \mathbf{Q} here are also determined heuristically. The initial value of \mathbf{R} is selected as

$$\begin{bmatrix} 20.18(m)^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3.3(m)^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0(rad)^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0(rad/sec)^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.000009(m/sec)^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.000016(m/sec)^2 \end{bmatrix} \quad (4.25)$$

The values of $\mathbf{R}(1,1)$ and $\mathbf{R}(2,2)$ are determined by error analysis of the output of an actual GARMIN GPS 15LW receiver over a period of several hours at the University of Plymouth testing-site with Latitude 50(deg) 22(min) 33.0552(sec) North and Longitude 004(deg) 08(min) 21.1438(sec) West. To generate the error time series shown in Figure 4.10, the *degrees – minutes – seconds* of difference between the output of the receiver and the actual known position was converted into metres, using methods available in the literature (Kennedy, 2002). $\mathbf{R}(5,5)$ and $\mathbf{R}(6,6)$ are chosen to represent the \mathbf{R} of an RDI Navigator Doppler Velocity Log at 1200(kHz)(Grenon

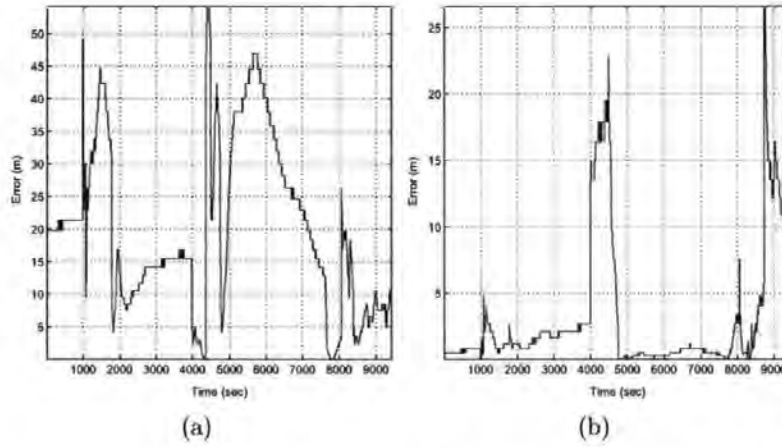


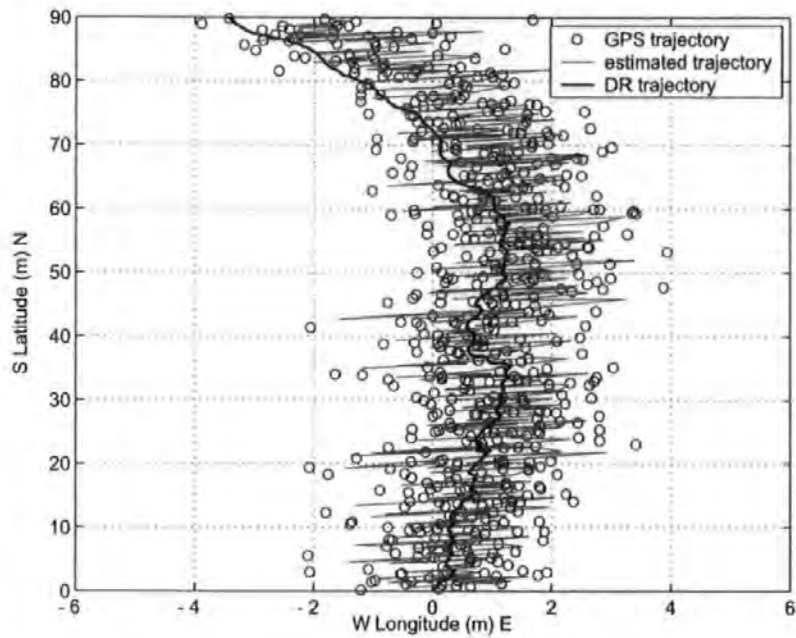
Figure 4.10: (a) Longitude error, (b) latitude error

et al., 2001). As the output from the imaginary yaw sensor assumed to be noise free, the initial values of $\mathbf{R}(3,3)$ and $\mathbf{R}(4,4)$ are selected as $0(\text{rad})^2$ and $0(\text{rad/sec})^2$ respectively.

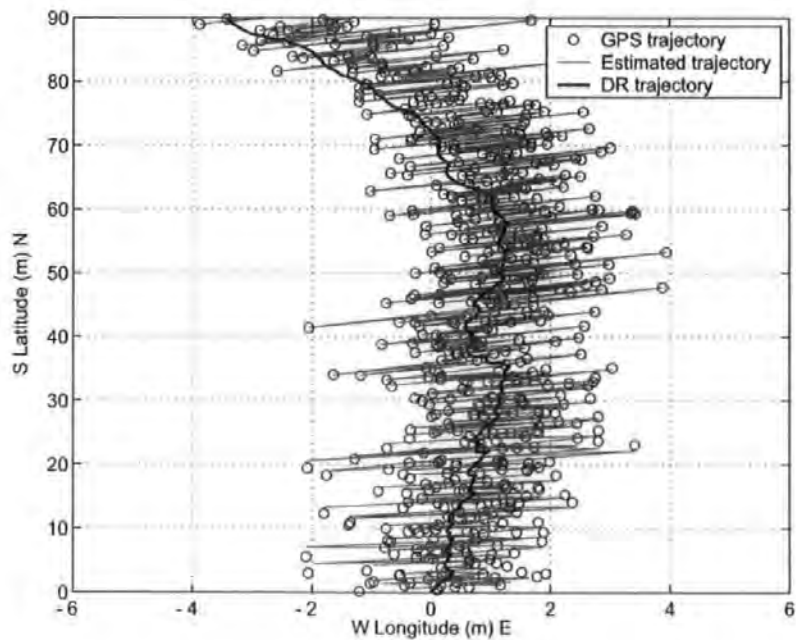
The FKF algorithm from Section 4.2 is then implemented, only this time the adaptation of the (i,i) -th element of \mathbf{R}_k is made in accordance with the (i,i) -th element of $\mathbf{\Delta R}_k$. Here a single-input-single-output (SISO) FIS as shown in Figure 4.1, is used sequentially to generate the correction factors for the elements in the main diagonal of \mathbf{R} as the following,

$$\mathbf{R}_k(i,i) = \mathbf{R}_{k-1}(i,i) + \Delta \mathbf{R}_k \quad (4.26)$$

Figure 4.11 and 4.12 are the simulation results showing the AUV trajectory operating on the surface at the start and the end of its mission. The longitude and latitude of the vehicle during the course of the mission are simulated being observed by the on-board GPS receiver with constant Gaussian noise with \mathbf{R} values lower than the assumed initial values. On the contrary, the vehicle's surge and sway velocity are simulated being observed by a Doppler velocity log (DVL) with constant Gaussian noise with \mathbf{R} values much higher than the assumed initial values. This logically will cause less weight being put on the position obtained by GPS and more on the prediction of position obtained from dead reckoning method (using DVL data) at

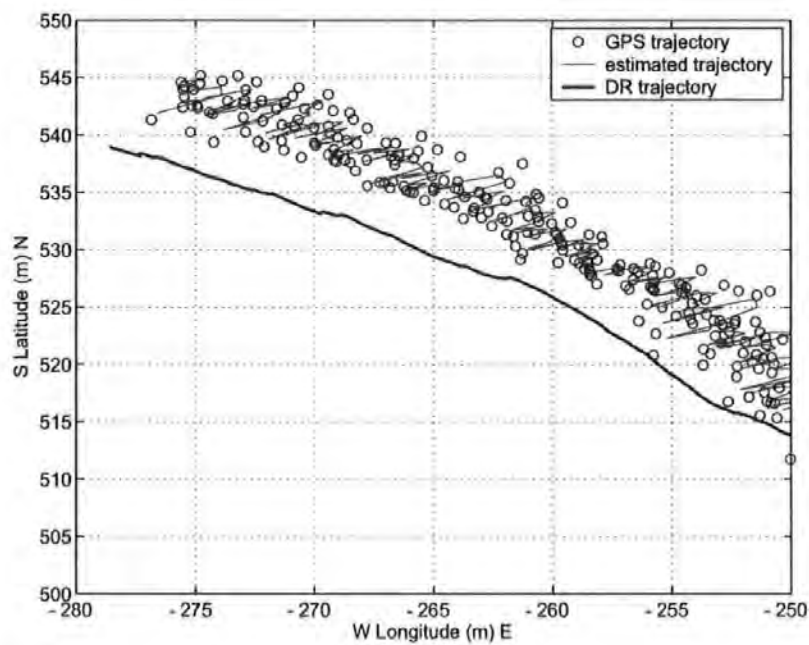


(a)

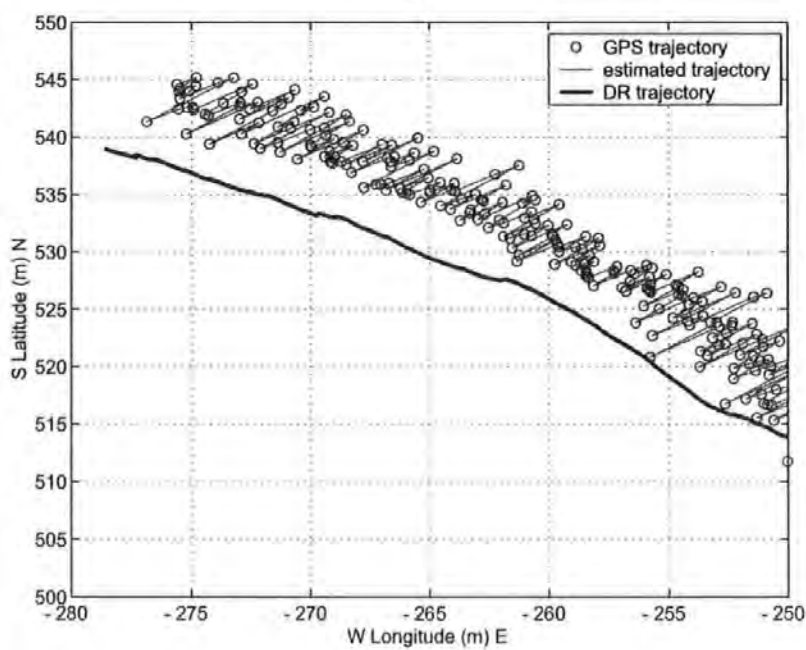


(b)

Figure 4.11: (a) Initial AUV trajectory using (a) standard EKF (b) FEKF



(a)



(b)

Figure 4.12: Final AUV trajectory using (a) standard EKF (b) FEKF

the start of the simulation and conversely towards the end. Figure 4.11(a) shows how the EKF make a relatively slower adjustments to the \mathbf{R} values compared to the adjustments made by the FEKF shown in Figure 4.11(b). It is clear here that only after a few sample of time, the FEKF has learned the true nature of the sensors noise and put more weight on the position obtained by the GPS receiver than on the prediction of position obtained by dead reckoning accordingly. Figure 4.12(a) shows how the EKF has still some 'confidence' on the dead reckoned position at the end of the simulated trajectory, while the FEKF shown in Figure 4.12(b) has put 100 per cent confidence on the GPS position. The EKF results in an estimated trajectory with RMSE of 0.6157(m) for longitude and 0.2626(m) for latitude. With the proposed FEKF adaptation, the RMSE for longitude is 0.1098(m) and 0.0158(m) for latitude.

4.9 CONCLUDING REMARKS

The problem with incomplete *a priori* knowledge of \mathbf{R} has been considered. Within this chapter, an adaptive Kalman filter approach, based on the filter innovation sequence coupled with fuzzy logic has been optimized using a MOGA and is discussed as an alternative for fusing INS sensor data and integrating GPS/INS position information. Implementation of this approach to the *Hammerhead* AUV heading model, whose responses are measured with sensors with different noise characteristics, has shown a significant result in improving the estimation of an individual KF. In Chapter 5, the principle of FKF and FEKF will be validated using real data obtained from several full scale trials. The performance of the algorithm will be tested not only on a surface (2D) scenario, but also on a surface-depth (3D) scenario.

CHAPTER 5

FUZZY KALMAN FILTER MULTIOBJECTIVE GENETIC ALGORITHM: A PSEUDO REAL-TIME IMPLEMENTATION

5.1 INTRODUCTION

Chapter 4 has shown the efficacy of the proposed FKF-MOGA technique to improve the accuracy of the *Hammerhead* integrated GPS/INS in a simulation environment. However, the real challenge lies in the application of the techniques to the real world. Real-time computations of a KF must take less time to execute than the time interval containing the total number of measurements processed by that KF. The consequences if this were not true are either only a reduced number of measurements are processed by the KF to keep up with the progression of time or the KF will insist on processing everything presented to it and gradually fall behind in the timeliness of computing its solution. The former, although maybe a suboptimal one, can still be considered as a real-time KF. The latter is certainly no longer considered as real-time KF.

The TCM2 electronic compass and the IMU on board the vehicle can produce heading data at $8(Hz)$, if acquired by MATLAB RS232/serial port I/O facility running in a Pentium-4 ($1.6(GHz)$) laptop. Running under the same hardware specification, the simulation of the FKF-MOGA with the parameters shown in Table 5.1, can only produce a result at approximately $0.88(Hz)$ or one solution at every $1.136(sec)$. These facts prevent the FKF-MOGA from being implemented even in a suboptimal level of a real-time KF. Therefore it was decided to emulate the real-time implementation of the FKF-MOGA by applying the algorithm in a simulation environment with sensor updates coming from real sensor data acquired during full scale trials. In this thesis,

Parameters	Values
Number of objective functions	5
Number of generation	5
Number of individual per generation	5
Generation gap in selection operation	0.95
Rate in rate in recombination operation	0.8
Rate in mutation operation	0.09

Table 5.1: MOGA parameters

this method is referred to as pseudo real-time application of the FKF-MOGA. Additionally, in this chapter, the algorithm is not only applied to provide an enhanced accuracy for the *Hammerhead* integrated GPS/INS for a surface (2D) mission, but also for a surface-depth (3D) mission. It will be clear from the forthcoming discussions and presented results that the proposed algorithm can perform equally well in both cases.

5.2 2D/SURFACE MISSION

This section discusses the implementation of the FKF-MOGA algorithms developed in the previous chapters for a 2D/surface mission using data acquired during a real-time experiment conducted in Roadford Reservoir, Devon, UK in July 2003.

The *Hammerhead* AUV model used in this chapter was derived using system identification techniques described in Chapter 3. The system matrix (**A**), input matrix (**B**) and output vector (**H**) (Equation (3.11) and (3.12)) are:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -0.98312 & 1.9831 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} -0.003196 \\ -0.0036115 \end{bmatrix}, \mathbf{H} = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

This model is assumed to be sufficiently accurate to represent the dynamics of the vehicle, and for this reason, any output produced by the model after being excited by an input, can be considered as an actual output value. This assumption also motivates the use of the model output as a reference in measuring the performance of the FKF-MOGA algorithm.

To test the FKF-MOGA algorithms, real data obtained from the TCM2 electronic compass and IMU (Figure 5.1(b)), as a response to the input shown in Figure 5.1(a), are fused together with two sets of simulated data. To produce the simulated data, the noise in Figure 5.2(a) and 5.2(b) are simply added to the TCM2 electronic compass and IMU real data respectively. A similar hypothetical real-time scenario as in Chapter 4, that can result in the noise with the characteristic shown in Figure 5.2(a), is adopted. In this particular scenario, the second TCM2 electronic compass (sensor-3) is located in close proximity to the propeller DC motor (Figure 3.2), whose internal temperature increases with time and affects the sensor ambient temperature. A similar scenario can also be considered to occur when the second IMU (sensor-4) is located in close proximity to the laser unit used in the VNS whose initial internal temperature is high and settles down after sometime. This particular scenario can result in the noise characteristic shown in Figure 5.2(b).

The initial condition are:

$$x_0 = \begin{bmatrix} 0(rad) \\ 0(rad) \end{bmatrix}; P_0 = \begin{bmatrix} 0.01(rad)^2 & 0 \\ 0 & 0.01(rad)^2 \end{bmatrix} \quad (5.1)$$

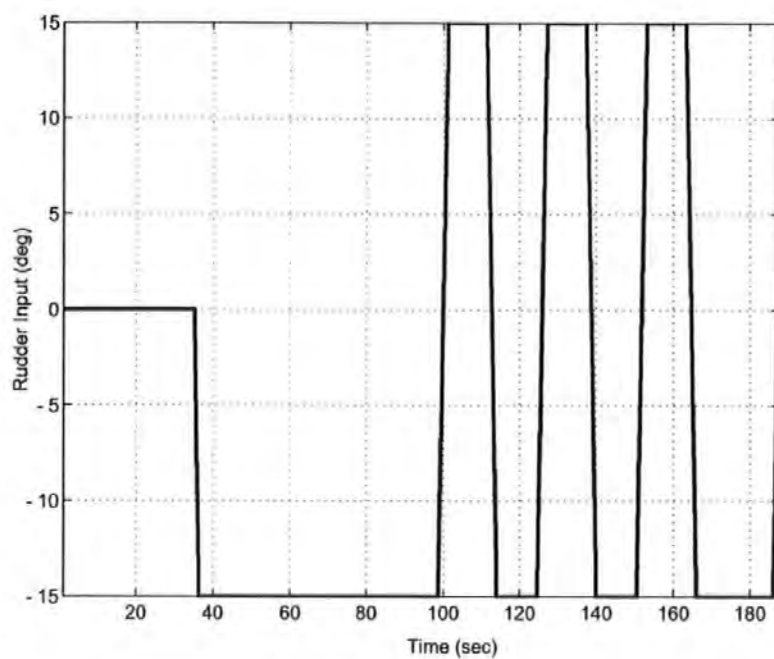
and Q_k is made constant as

$$Q_k = \begin{bmatrix} 0(rad)^2 & 0 \\ 0 & 0.1725 \times 10^{-7}(rad)^2 \end{bmatrix} \quad (5.2)$$

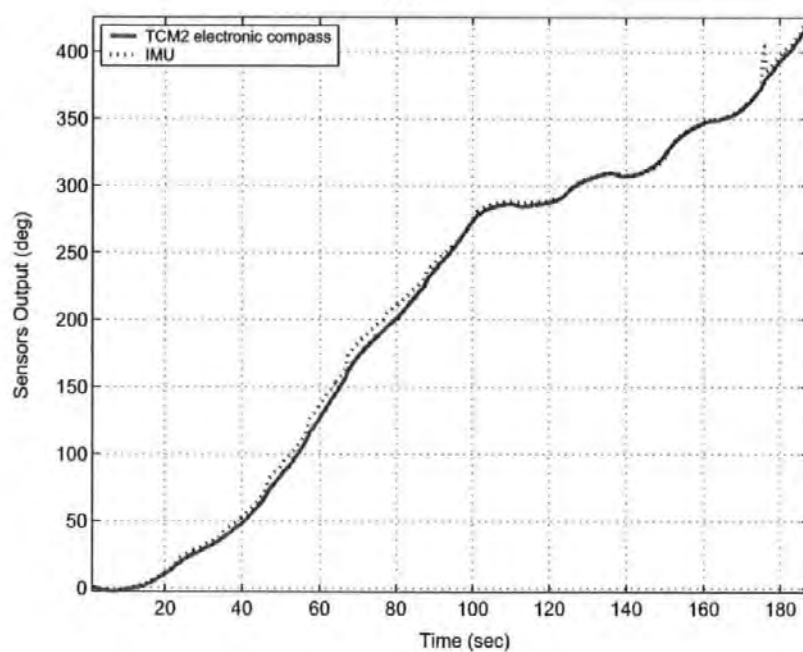
The actual value of R is assumed unknown, but its initial value is selected according to the heading accuracy of the sensors (see Table 3.3 and 3.5), *i.e.* $1(deg)^2$.

The covariance matching technique discussed in Chapter 4 is then implemented to maintain the performance of the estimation process. Fuzzy rules are kept the same as in Equation (4.4)-(4.6) as followings:

IF $\langle \delta a_k \cong 0 \rangle$ **THEN** $\langle R_k \text{ is unchanged} \rangle$

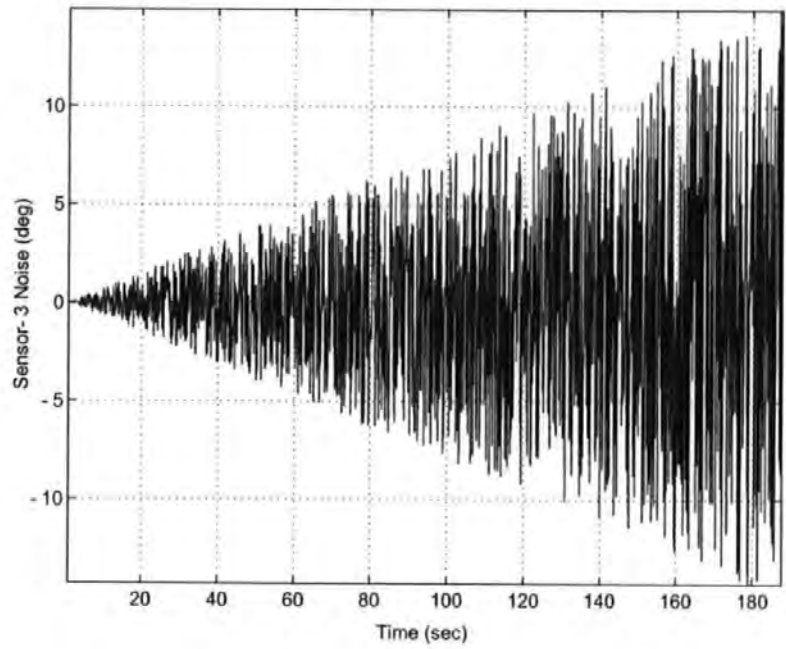


(a)

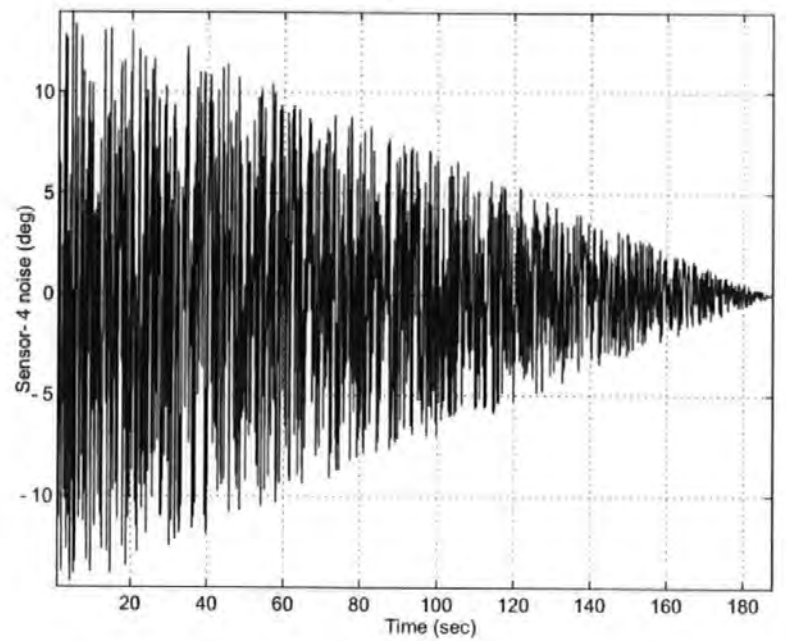


(b)

Figure 5.1: (a) Rudder input, (b) TCM2 electronic compass and IMU output



(a)



(b)

Figure 5.2: (a) Sensor-3 noise, (b) sensor-4 noise

IF $\langle \text{delta}_k > 0 \rangle$ THEN $\langle \mathbf{R}_k \text{ is decreased} \rangle$

IF $\langle \text{delta}_k < 0 \rangle$ THEN $\langle \mathbf{R}_k \text{ is increased} \rangle$

delta_k is the discrepancy between $\hat{\mathbf{C}}_{Inn_k}$ and \mathbf{S}_k , the actual and theoretical covariance of innovation Inn_k . \mathbf{R}_k is then adjusted by adding its previous value, \mathbf{R}_{k-1} with $\Delta \mathbf{R}_k$, the output of the FIS. Initial membership functions of the FIS are taken to be the same as the ones in Chapter 4 and repeated in Figure 5.3(a) and 5.3(b) for clarity. Subsequent optimization of $\Delta \mathbf{R}_k$ membership functions using MOGA with parameters shown in Table 5.2, produces membership functions shown in Figure 5.3(c). As in the previous chapter, the MOGA parameters are selected heuristically after exhaustive tests and it was observed that no significant improvement can be achieved using different set of parameters. Trade-off graphs of this particular search is shown in Figure 5.4. The goal of the objective functions, determined heuristically based on similar experiments from the previous chapters is: $[0.1565(rad) \ 0.5125(rad) \ 0.3043(rad) \ 0.1984(rad) \ 0.0770(rad)]^T$. Results of the implementation of the FKF-MOGA techniques are shown in the next section.

Parameters	Values
Number of objective functions	5
Number of generation	25
Number of individual per generation	10
Generation gap in selection operation	0.95
Rate in rate in recombination operation	0.8
Rate in mutation operation	0.09

Table 5.2: MOGA parameters

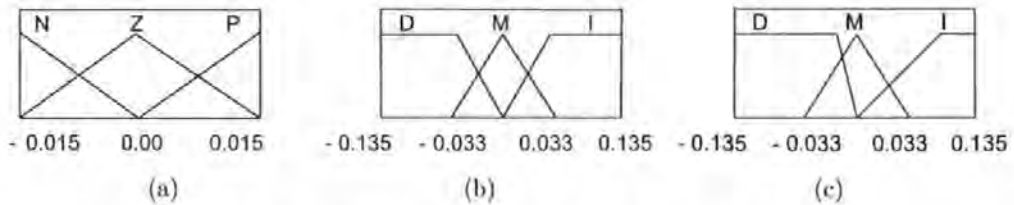


Figure 5.3: Membership function of (a) delta_k and (b) $\Delta \mathbf{R}_k$ before optimization, and (c) $\Delta \mathbf{R}_k$ after optimization

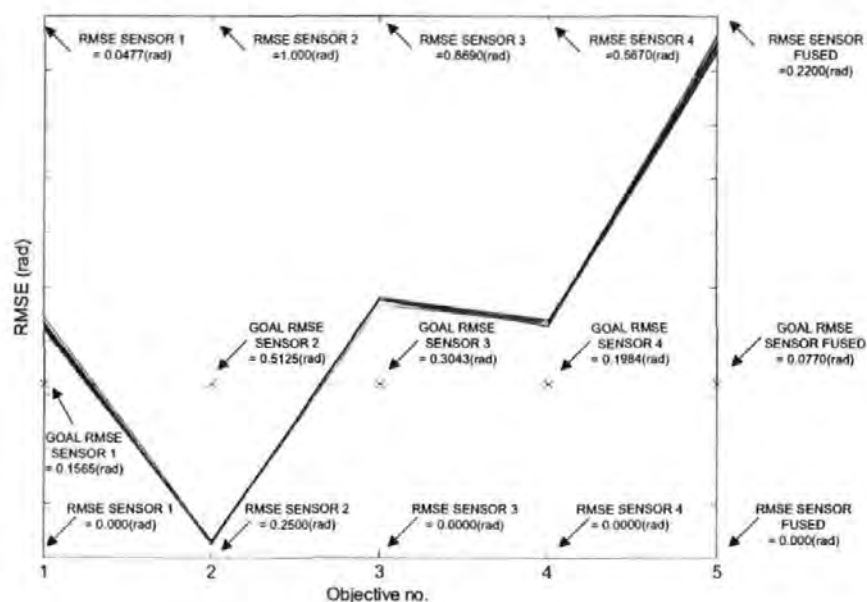


Figure 5.4: Trade-off graphs for the FKF search

5.2.1 Implementation Results

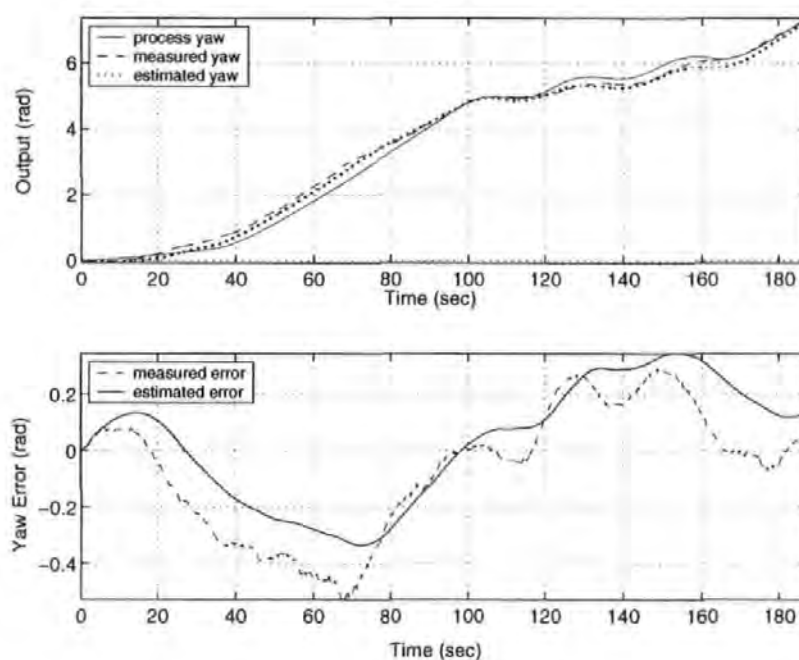


Figure 5.5: Measured and estimated yaw output and error of sensor-1

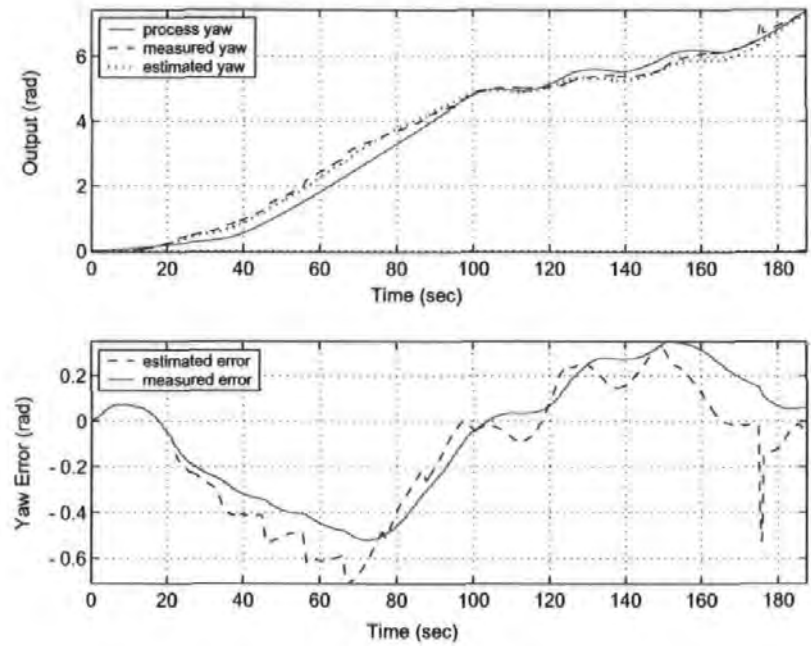


Figure 5.6: Measured and estimated yaw output and error of sensor-2

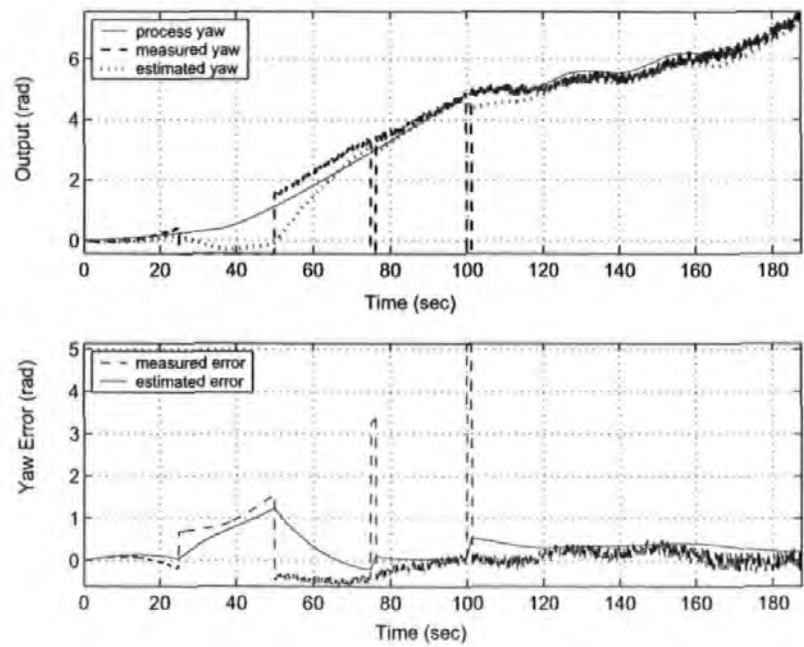


Figure 5.7: Measured and estimated yaw output and error of sensor-3

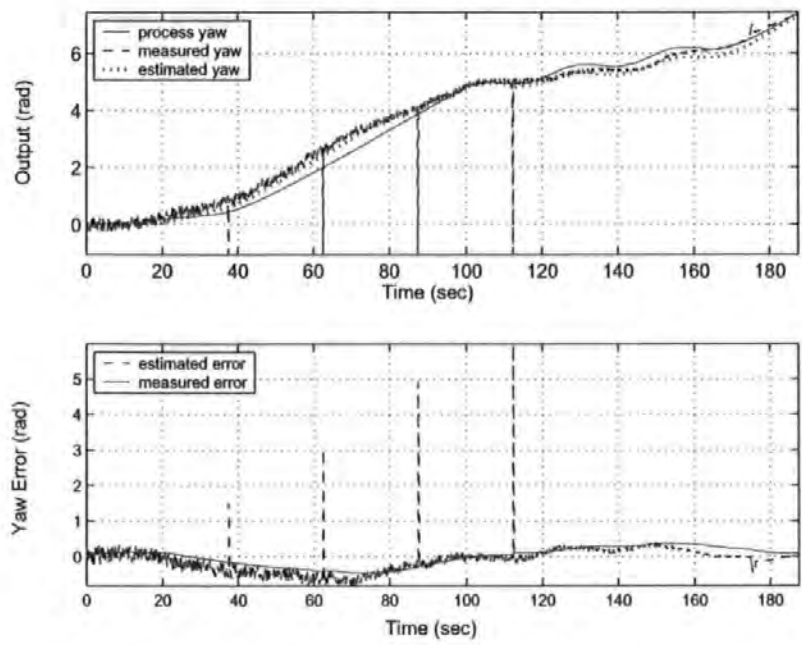


Figure 5.8: Measured and estimated yaw output and error of sensor-4

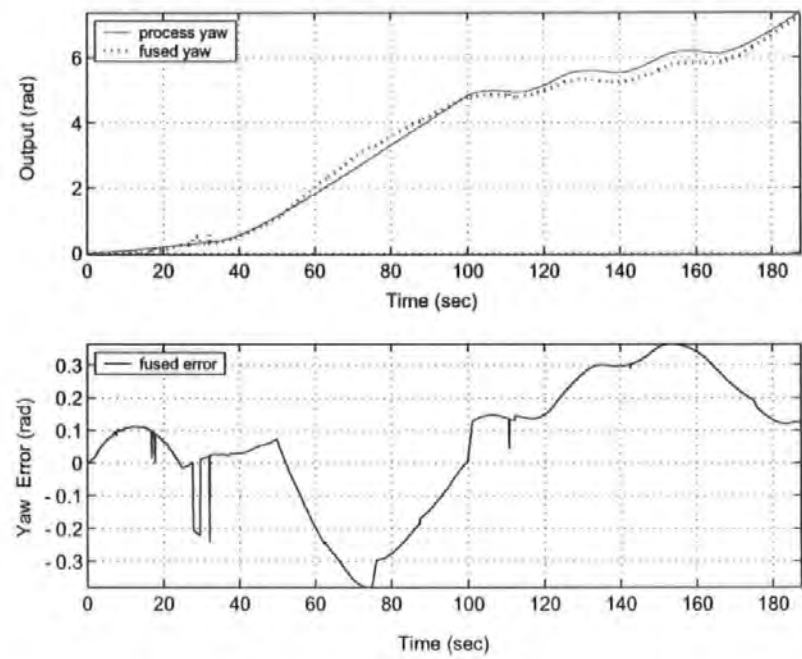


Figure 5.9: Measured and estimated yaw output and error of fused sensor

Sensor	$J_{zv}(rad)$	$J_{ze}(rad)$	
		GA	MOGA
Sensor-1	0.2340	0.2090	0.2094
Sensor-2	0.2960	0.3047	0.2761
Sensor-3	0.6558	0.4131	0.4130
Sensor-4	0.3852	0.2552	0.2551
Fused		0.2487	0.2088

Table 5.3: Comparison of performance

Figure 5.5 and 5.6 are the simulation results showing the response of the *Hammerhead* AUV observed by the TCM2 electronic compass and IMU respectively, while Figure 5.7 and 5.8, which are the output of the two former sensors added with uniform noise increasing and decreasing with time respectively. Figure 5.5 and Figure 5.6 show improvements in the level of error produced by the proposed FKF-MOGA algorithms as compared to direct measurements from the sensors. Apart from the improvements in the level of error, Figure 5.7 and 5.8 also show how the proposed algorithms has detected transient and persistent faults (see Chapter 4 for definition of these faults) in the sensors and made an appropriate recovery. Figure 5.9 shows the result of fusing the four sensors.

Table 5.3 shows the performance of the sensors, indicated by J_{zv} and J_{ze} (Equation (4.12) and (4.13)). A close look on the J_{zv} and J_{ze} , of each sensor indicates that the FKF with single objective GA optimization has improved the accuracy of the heading information of sensor-1 to sensor-4. However, the result of fusing the estimated sensor data has shown a slightly inferior performance, indicated by $J_{ze} = 0.2487(rad)$, compared to the performance of sensor-1, indicated by $J_{zv} = 0.2340(rad)$. This can be understood as a direct result of fusing a relatively accurate sensor-1, with other sensors that are less accurate. A further comparison is made between individual sensor performance of non-MOGA and MOGA case. It is clear that the individual sensor performance of the MOGA case, with the exception of sensor-1, has produced some improvements, with sensor-2 as the most noticeable one. It is clear that the improvement on sensor-2 has brought about an overall significant improvement on the quality of the estimation of the MOGA fused sensor, which is indicated by $J_{ze} = 0.2088(rad)$. As before, the fused estimated yaw is treated as a single imaginary yaw sensor and

used by other INS sensors to transform data from body co-ordinate frame to Earth-centered Earth-fixed co-ordinate frame where integration with GPS data is performed using the FEKF techniques.

In addition to Equations (4.14) - (4.21), a new output measurement matrix $\mathbf{H}(X(t))$ is defined as the following:

$$\mathbf{H}(X(t)) = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.3)$$

when GPS signal is available, and when it is not,

$$\mathbf{H}(X(t)) = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (5.4)$$

The continuous time model of Equations (4.16) - (4.21) and Equations (5.3)-(5.4) are then linearised to obtain an EKF with an effective state prediction equation in simple form, producing:

$$\mathbf{F}_{lin}(t) = \begin{bmatrix} 0 & 0 & -u(t) \sin \psi_{im}(t) - v(t) \cos \psi_{im}(t) & 0 & \cos \psi_{im}(t) & -\sin \psi_{im}(t) \\ 0 & 0 & u(t) \cos \psi_{im}(t) - v(t) \sin \psi_{im}(t) & 0 & \sin \psi_{im}(t) & \cos \psi_{im}(t) \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.5)$$

Please note this is Equation (4.22) and repeated here to maintain clarity of the discussion in this section. The $\mathbf{H}_{in}(t)$ is a matrix identical as in either Equation (5.3) or (5.4). Subsequent discretisation with period $T = 0.125(sec)$ of the linearised model results in the EKF algorithm as described in Appendix D. The initial conditions are:

$\mathbf{P}_0 =$

$$\begin{bmatrix} 0.01(m)^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.01(m)^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01(rad)^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01(rad/sec)^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.01(m/sec)^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.01(m/sec)^2 \end{bmatrix} \quad (5.6)$$

and \mathbf{Q} is made constant as

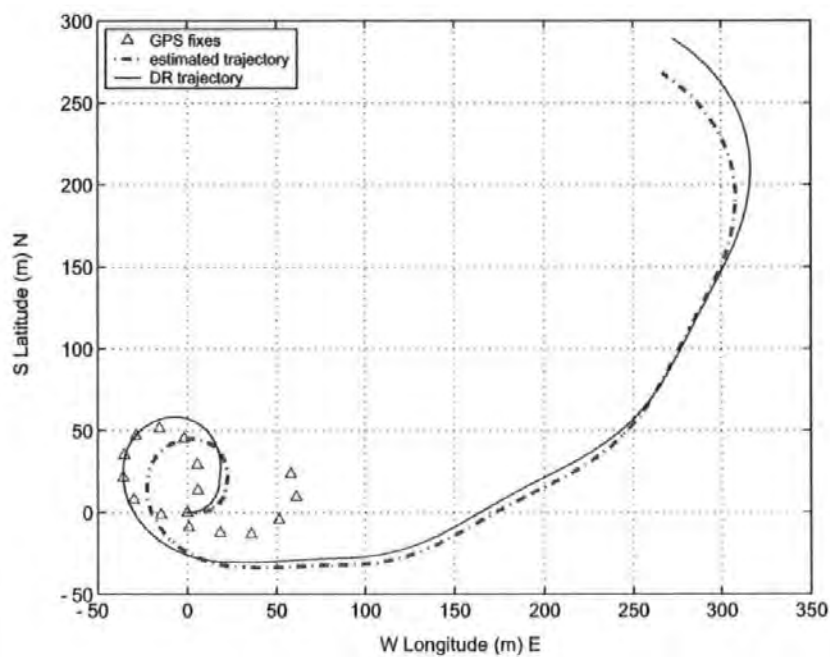
$$\begin{bmatrix} 10(m)^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10(m)^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0175(rad)^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1(rad/sec)^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1(m/sec)^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1(m/sec)^2 \end{bmatrix} \quad (5.7)$$

The actual value of \mathbf{R} is assumed unknown but its initial value is selected as:

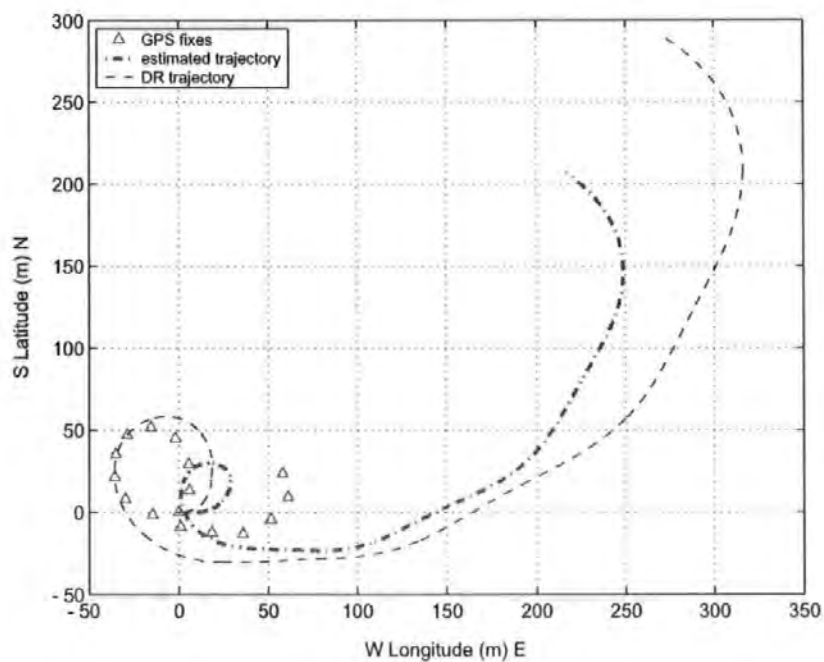
$$\begin{bmatrix} 1000(m)^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1000(m)^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0873(rad)^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0175(rad/sec)^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2(m/sec)^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2(m/sec)^2 \end{bmatrix} \quad (5.8)$$

The FEKF algorithm is then implemented to the diagonal element of \mathbf{R}_k .

Figure 5.10(a) shows the *Hammerhead* AUV trajectory obtained using GPS, dead reckoning using INS sensors (through double integration of the accelerometer data



(a)



(b)

Figure 5.10: (a) AUV trajectory obtained using GPS, INS sensors (dead reckoning method) and GPS/INS using EKF without adaptation, (b) AUV trajectory obtained using GPS, INS sensors (dead reckoning method) and GPS/INS using EKF with adaptation

with respect to time) and integrated GPS/INS. As the initial value of \mathbf{R} for both $X_{NED}(t)$ and $Y_{NED}(t)$ is $1000(m^2)$, the standard EKF algorithm puts less weight on the position obtained by GPS and more on the prediction of position obtained from dead reckoning method (using INS sensor data). Figure 5.10(b) shows that the matrix has been adjusted accordingly and more weight is given to the GPS data, and therefore the estimated trajectory in the integrated GPS/INS is "pulled" a little bit further to the GPS trajectory. However, big discrepancies can still be appreciated between the integrated GPS/INS estimate with respect to the GPS fixes. There are several explanations to this erratic behaviour. The first possibility is that it is caused by the poor level of accuracy of the low-cost GPS being used in this particular application. It is important to note that the proposed algorithm has detected a persistent high actual covariance ($\hat{\mathbf{C}}_{Innk}$) for both X_{NED} and Y_{NED} throughout the trajectory. This results in insufficient weight being given to the GPS fixes in the FEKF and more on the position obtained by the dead reckoning. The second possibility is that the GPS receiver did not lock into a sufficient number of satellites with a sufficiently small value of position dilution of precision (PDOP) that can provide the required level of accuracy. The use of a differential global positioning system (DGPS) receiver or a GPS receiver with a wide area augmentation system (WAAS) or a European geostationary navigation overlay service (EGNOS) capability can be considered as a way forward to alleviate this problem.

5.3 3D/SURFACE-DEPTH MISSION

Many missions performed by AUVs require the vehicle to operate not only on the surface of the sea, but also at a particular depth. Examples of such AUVs and their specific missions can be found in Chapter 2. The *Hammerhead* AUV is also designed to be able to dive to a certain depth and perform a particular mission, such as tracking underwater cables for maintenance purposes or landmark recognition for an underwater absolute positioning system as proposed in Loebis *et al.* (2003a). To carry out these missions, the *Hammerhead* AUV is equipped with underwater image acquisition techniques (Dalglish *et al.*, 2003), coupled with a laser stripe illumination methodology (LSI) developed previously by Cranfield University (Tetlow and

Allwood, 1995) to provide an enhanced viewing of the seabed below the vehicle, and a depth controller developed by Naeem (2004).

The concept of 3D pseudo real-time navigation system enhanced by the proposed techniques is demonstrated in this section. The real data used herein are those generated by the individual TCM2 electronic compass and IMU, their respective simulated counterparts, and their overall fused values. Further real-time experiments are considered to be imperative and must be conducted before a full-scale pseudo real-time implementation of the proposed techniques can be undertaken. This, however, due to the amount of time required to do so and to analyse the data produced thereby, is considered to be suitable for the future work of the *Hammerhead* AUV.

The mission scenario adopted in this section is designed to mimic the actual cable-tracking or landmark recognition that will be performed in the future by the *Hammerhead* vehicle. This involves acquiring GPS/INS data on the surface and subsequently finding the estimated trajectory before sending the vehicle to a certain depth. Once the vehicle is under the water, the GPS signals are completely blocked and the GPS/INS navigation system is replaced by a pure dead reckoning navigation system. During this period, the underwater image acquisition algorithms continuously observing the area beneath the vehicle to find a cable to be tracked or underwater landmarks to be identified and used as underwater absolute position fixes. In conditions where sufficient illumination is available in identifying those objects, produced either by the LSI or natural ambient light, the vehicle is then controlled to maintain its current depth. Otherwise, the depth controller algorithm will act accordingly and send the vehicle further down until sufficient illumination is obtained. After a certain period of time the vehicle is sent back to the surface to obtain GPS fixes that are used to reset the drift or the accumulated error produced by the dead reckoning navigation system.

Figure 5.11, 5.12 and 5.13 show the simulated surge, sway and heave of the vehicle respectively, obtained after integrating the corresponding body co-ordinate frame acceleration data. The true values of the surge and sway are, respectively, defined as $1.3(m/sec)$ and $\pm 0.1(m/sec)$. The heave values are defined into five parts. The first and last parts are to show the heave of the vehicle when it is operating on the

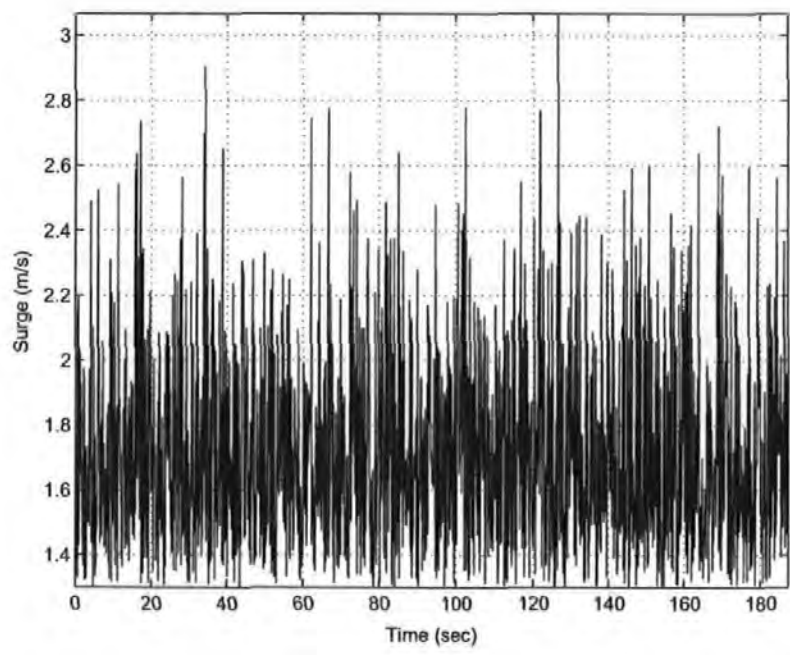


Figure 5.11: Surge

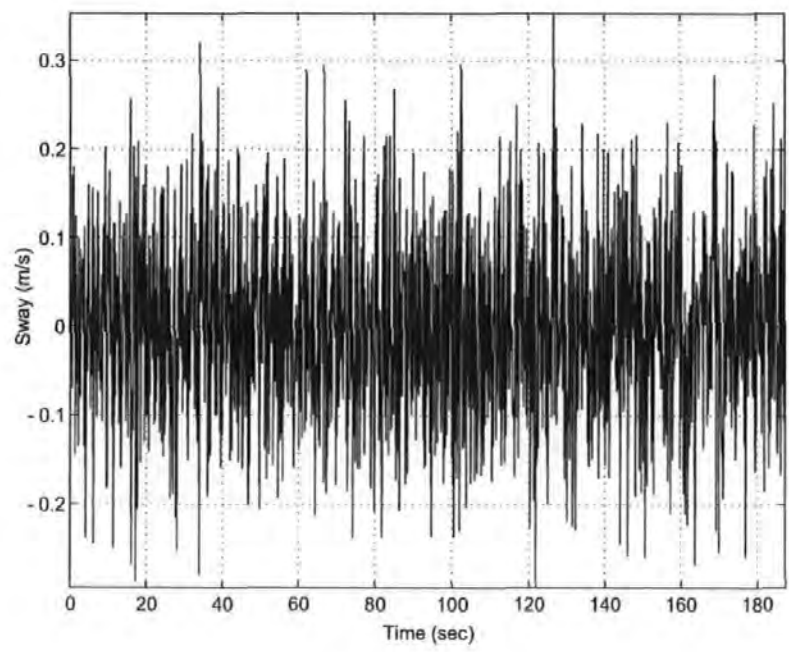


Figure 5.12: Sway

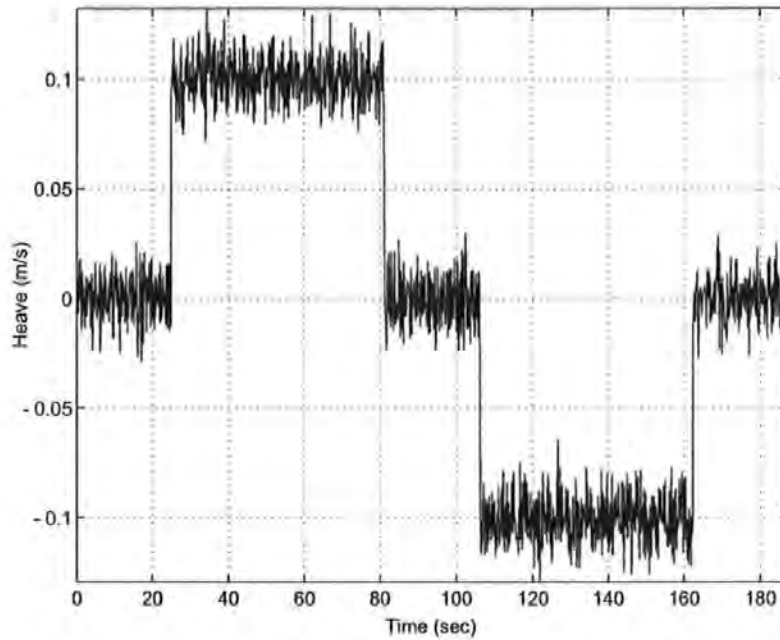


Figure 5.13: Heave

surface, *i.e.*, true values are assumed to be $0(m/sec)$. The second part is to show the heave of the vehicle as it is descending to a certain depth, defined here as $-0.1(m/sec)$. Once the vehicle reaches this, the depth controller is employed to maintain the depth of the vehicle. Consequently, the true heave during this period is defined to be $0(m/sec)$. Finally, the vehicle is sent back to the surface, and the heave during ascending period is defined to be $0.1(m/sec)$. It is clear that the errors added to these true values will contribute to the total drift suffered by the dead reckoning navigation system in finding the position of the vehicle when it is operating under the water.

Figure 5.14 and 5.15 show, respectively, pitch and pitch rate of the vehicle during this particular mission. Figure 5.14 shows the pitch of the vehicle when it is on the surface, descending, at a constant depth, ascending (and back) to the surface. Figure 5.15 shows the corresponding pitch rate. It is assumed that the roll of the vehicle is constant and stable at $0(rad)$ during the course of the mission. For clarity, the yaw of the individual TCM2 electronic compass and IMU, their respective simulated counterparts, and their overall fusion values will be repeated in the forthcoming discussion. These are presented along with their associated yaw rate values.

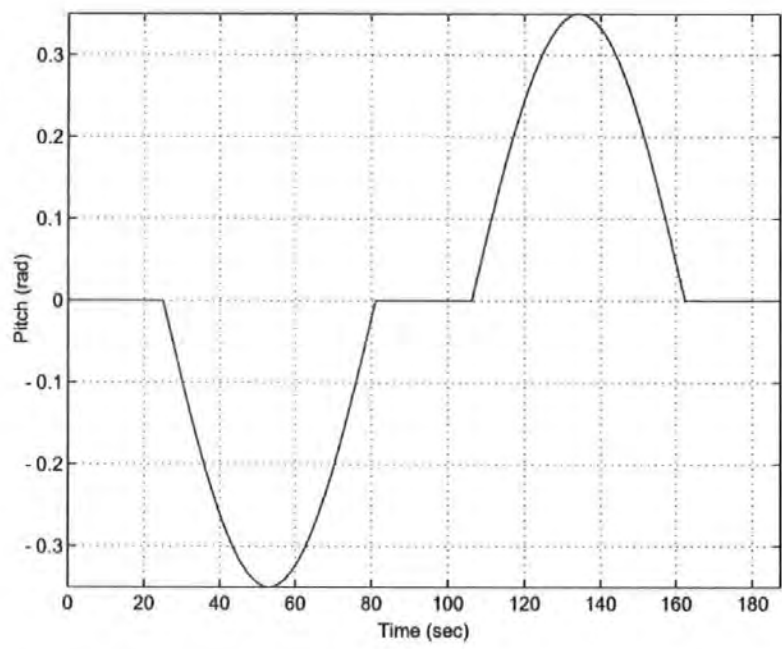


Figure 5.14: Pitch

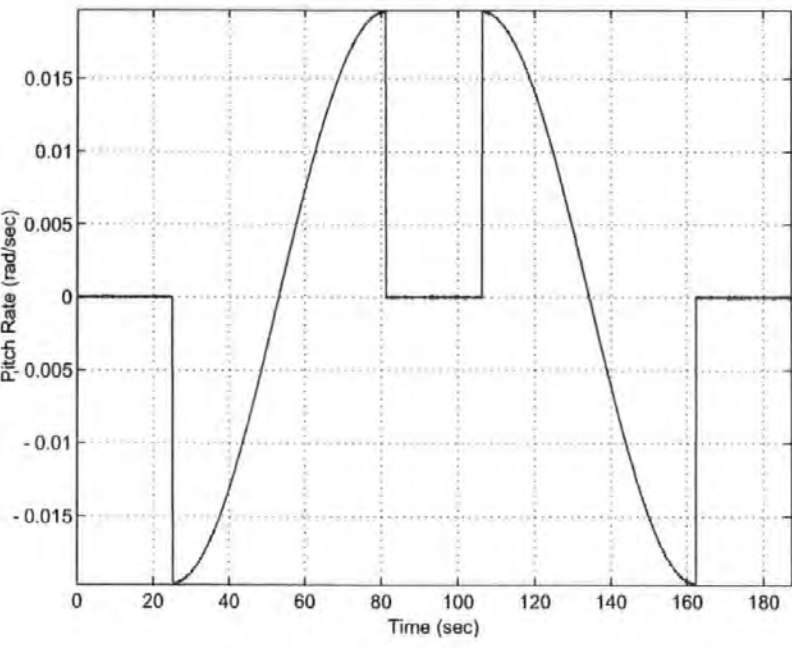


Figure 5.15: Pitch rate

A continuous time model of the vehicle motion appropriate to this problem is taken to be as in Equations (4.14) and (4.15), with $X(t)$, the state vector, is equal to $[X_{NED}(t) \ Y_{NED}(t) \ Z_{NED}(t) \ \theta(t) \ q(t) \ \psi_{im}(t) \ r(t) \ u(t) \ v(t) \ w(t)]^T$. $X_{NED}(t)$, $Y_{NED}(t)$ are the longitude and latitude of the AUV position converted from deg-min-sec in the Earth-centered Earth-fixed co-ordinate frame into metres in the NED co-ordinate frame, $Z_{NED}(t)$ is the depth of the vehicle, $\theta(t)$ is the pitch, $q(t)$ is the pitch rate, $\psi_{im}(t)$ is the yaw angle obtained from the imaginary yaw sensor, $r(t)$ is yaw rate, $u(t)$, $v(t)$ and $w(t)$ are the surge, sway and heave velocity respectively. The model states are related through the following kinematically based set of functions (refer to Appendix G for body co-ordinate frame to Geographical/North-East-Down co-ordinate frame transformation):

$$\dot{X}_{NED}(t) = u(t) \cos \psi_{im}(t) \cos \theta(t) - v(t) \sin \psi_{im}(t) + w(t) \cos \psi_{im}(t) \sin \theta(t) \quad (5.9)$$

$$\dot{Y}_{NED}(t) = u(t) \sin \psi_{im}(t) \cos \theta(t) + v(t) \cos \psi_{im}(t) + w(t) \sin \psi_{im}(t) \sin \theta(t) \quad (5.10)$$

$$\dot{Z}_{NED}(t) = -u(t) \sin \theta(t) + w(t) \cos \theta(t) \quad (5.11)$$

$$\dot{\theta}(t) = q(t) \quad (5.12)$$

$$\dot{q}(t) = 0 \quad (5.13)$$

$$\dot{\psi}_{im}(t) = r(t) \quad (5.14)$$

$$\dot{r}(t) = 0 \quad (5.15)$$

$$\dot{u}(t) = a_{x_{BODY \rightarrow NED}}(t) \quad (5.16)$$

$$\dot{v}(t) = a_{y_{BODY \rightarrow NED}}(t) \quad (5.17)$$

$$\dot{w}(t) = a_{z_{BODY \rightarrow NED}}(t) \quad (5.18)$$

where $a_{x_{BODY \rightarrow NED}}(t)$, $a_{y_{BODY \rightarrow NED}}(t)$ and $a_{z_{BODY \rightarrow NED}}(t)$ are the acceleration of the vehicle acquired in the body co-ordinate frame and transformed subsequently to the NED co-ordinate frame.

The output measurements are related through the states by an identity matrix $\mathbf{I}_{10 \times 10}$ when the vehicle is operating on the surface. When the vehicle is operating under the water, pure dead reckoning is used. Linearisation about the current estimates of the continuous time model of Equation (5.9)-(5.18), producing $\mathbf{F}_{lin}(t)$ with the following

non-zero components (readers interested in the techniques of linearisation in Kalman filtering are referred to Brown and Hwang (1997)):

$$\mathbf{F}_{lin[1,4]}(t) = -u(t) \cos \psi_{im}(t) \sin \theta(t) + w(t) \cos \psi_{im}(t) \cos \theta(t)$$

$$\mathbf{F}_{lin[1,6]}(t) = -u(t) \sin \psi_{im}(t) \cos \theta(t) - v(t) \cos \psi_{im}(t) - w(t) \sin \psi_{im}(t) \sin \theta(t)$$

$$\mathbf{F}_{lin[1,8]}(t) = \cos \psi_{im}(t) \cos \theta(t)$$

$$\mathbf{F}_{lin[1,9]}(t) = -\sin \psi_{im}(t)$$

$$\mathbf{F}_{lin[1,10]}(t) = \cos \psi_{im}(t) \sin \theta(t)$$

$$\mathbf{F}_{lin[2,4]}(t) = -u(t) \sin \psi_{im}(t) \sin \theta(t) + w(t) \sin \psi_{im}(t) \cos \theta(t)$$

$$\mathbf{F}_{lin[2,6]}(t) = u(t) \cos \psi_{im}(t) \cos \theta(t) - v(t) \sin \psi_{im}(t) + w(t) \cos \psi_{im}(t) \sin \theta(t)$$

$$\mathbf{F}_{lin[2,8]}(t) = \sin \psi_{im}(t) \cos \theta(t)$$

$$\mathbf{F}_{lin[2,9]}(t) = \cos \psi_{im}(t)$$

$$\mathbf{F}_{lin[2,10]}(t) = \sin \psi_{im}(t) \sin \theta(t)$$

$$\mathbf{F}_{lin[3,4]}(t) = -u(t) \cos \theta(t) - w(t) \sin \theta(t)$$

$$\mathbf{F}_{lin[3,8]}(t) = -\sin \theta(t)$$

$$\mathbf{F}_{lin[3,10]}(t) = \cos \theta(t)$$

$$\mathbf{F}_{lin[4,5]}(t) = 1$$

$$\mathbf{F}_{lin[6,7]}(t) = 1$$

with $\mathbf{H}_{lin}(t) = \mathbf{I}_{10 \times 10}$. The FEKF algorithm is then implemented after subsequent discretisation with period $T = 0.125(\text{sec})$. The initial conditions are $X_0 = 0\mathbf{I}_{10 \times 10}$ and $\mathbf{P}_0 = 0.01\mathbf{I}_{10 \times 10}$, and \mathbf{Q} is made constant as with the following components:

$$\mathbf{Q}_{[1,1]} = 0.01(m)^2$$

$$\mathbf{Q}_{[2,2]} = 0.01(m)^2$$

$$\mathbf{Q}_{[3,3]} = 0.01(m)^2$$

$$\mathbf{Q}_{[4,4]} = 0.000001(rad)^2$$

$$\mathbf{Q}_{[5,5]} = 0.01(rad/sec)^2$$

$$\mathbf{Q}_{[6,6]} = 0.000001(rad)^2$$

$$\mathbf{Q}_{[7,7]} = 0.01(rad/sec)^2$$

$$\mathbf{Q}_{[8,8]} = 0.01(m/sec)^2$$

$$\mathbf{Q}_{[9,9]} = 0.01(m/sec)^2$$

$$\mathbf{Q}_{[10,10]} = 0.01(m/sec)^2$$

The initial value of \mathbf{R} is selected as:

$$\mathbf{R}_{[1,1]} = 10(m)^2$$

$$\mathbf{R}_{[2,2]} = 10(m)^2$$

$$\mathbf{R}_{[3,3]} = 5(m)^2$$

$$\mathbf{R}_{[4,4]} = 0.000001(rad)^2$$

$$\mathbf{R}_{[5,5]} = 0.000001(rad/sec)^2$$

$$\mathbf{R}_{[6,6]} = 0(rad)^2$$

$$\mathbf{R}_{[7,7]} = 0(rad/sec)^2$$

$$\mathbf{R}_{[8,8]} = 2(m/sec)^2$$

$$\mathbf{R}_{[9,9]} = 2(m/sec)^2$$

$$\mathbf{R}_{[10,10]} = 2(m/sec)^2$$

Figure 5.17 shows the result of implementing the proposed FEKF algorithm to the 3D/surface-depth mission described in the beginning of the section using the yaw produced by sensor-1 (shown in Figure 4.4 and repeated in Figure 5.16 for clarity). Figure 5.18 present a closer look on the start and the end of the mission. Figure 5.20, 5.23, 5.26 and 5.29 present the results of implementing the proposed FEKF algorithm using, respectively, the yaw data produced by sensor-2 (Figure 5.19), sensor-3 (Figure 5.22), sensor-4 (Figure 5.25) and the fusion results (Figure 5.28). Closer looks on the start and end of mission produced by implementing the proposed algorithm using sensor-2, sensor-3, sensor-4 and the fusion results respectively, are shown in Figure 5.21, 5.24, 5.27, 5.30.

It is clear, that the initial GPS/INS surface trajectory using the yaw produced by the individual sensor contain an unexpected drift in vertical direction. This is a direct result of assuming the measurement noise in this direction as being higher than its corresponding process noise. The values of the measurement and process covariance

matrices are indicated respectively as $\mathbf{R}_{[3,3]} = 5(m)^2$ and $\mathbf{Q}_{[3,3]} = 0.01(m)^2$. Consequently, the EKF algorithm puts more confidence on the process, *i.e.*, integrating the $\dot{Z}_{[NED]}$, than the measurement of depth from the pressure transducer. It is clear, as indicated by Equation (5.11), that integrating the value of $\dot{Z}_{[NED]}$ consequently integrates the noise in the $u(t)$, ψ_{im} , w and θ . This in turn produces an accumulation of error and needs to be reset to $Z_{NED} = 0(m)$, right before the vehicle dives. The reset mechanism can easily be seen in Figure 5.18(a), 5.21(a), 5.24(a) and 5.27(a).

Once the vehicle is below the surface, the depth controller and the underwater image acquisition algorithms will work side by side to find objects of interest and to maintain a constant depth thereafter for a specific period of time. This is shown in Figure 5.17, 5.20, 5.23 and 5.26. Particular emphasis is placed on Figure 5.23 and 5.26. The two trajectories were produced using the yaw from sensor-3 and sensor-4, which contain persistent and transient faults respectively. It can be observed from Figure 5.23, the first persistent fault (shown in Figure 5.22) introduces a significant error into the vehicle's underwater navigation system. Although the second and the third persistent faults did not occur as long as the first one, they still indeed exacerbate the overall accuracy and reliability of the system. It can also be observed from Figure 5.26 that the fault in sensor-4 did not affect the underwater navigation system significantly as it is only transient in nature. This 'glitch' only produced an infinitesimal amount of error in Euler angle computation and subsequently in the underwater DR process.

The vehicle is then sent back to the surface to obtain GPS fixes used to reset the drift produced by the dead reckoning process during the underwater mission. This particular mechanism is shown in Figure 5.18(b), 5.21(b), 5.24(b) and 5.27(b). A similar case a DR error also occurs at this stage. Although the depth has been reset to $0(m)$, the EKF algorithm soon puts more confidence on the vertical DR process and consequently produces an estimate of depth larger than $0(m)$. This also happens to the horizontal (X_{NED} and Y_{NED}) estimation process. As the measurement covariance matrices for both the longitude and latitude are $\mathbf{R}_{[1,1]} = \mathbf{R}_{[2,2]} = 10(m)^2$, the estimation process put more weight on the DR processes, which are assumed to have $\mathbf{Q}_{[1,1]} = \mathbf{Q}_{[2,2]} = 0.01(m)^2$ process covariance matrices.

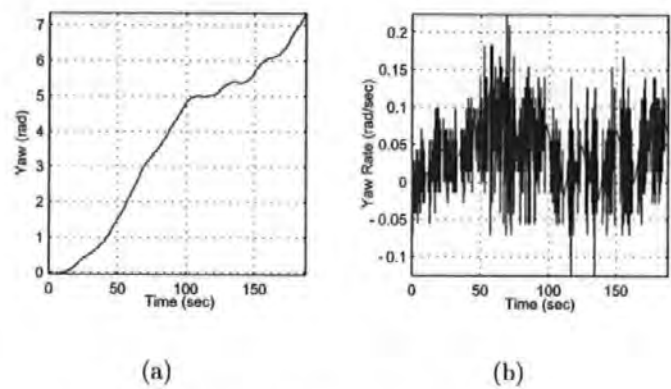


Figure 5.16: (a) Yaw sensor-1, (b) yaw rate sensor-1

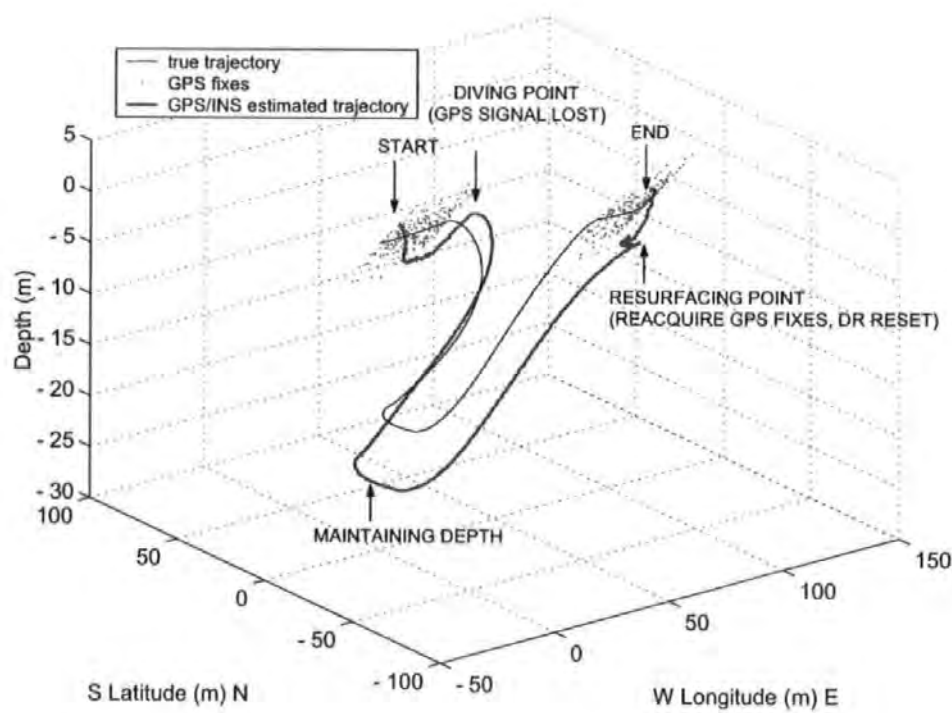
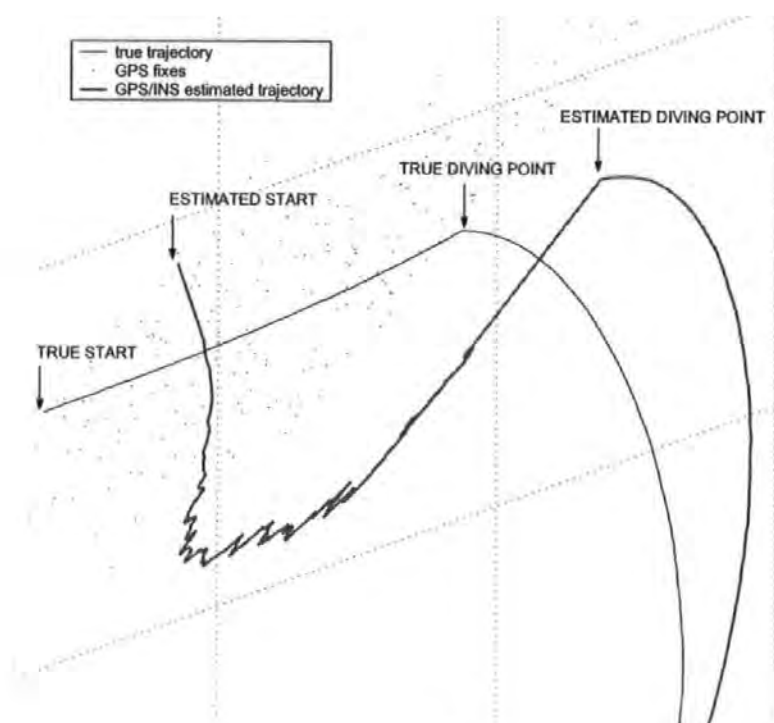
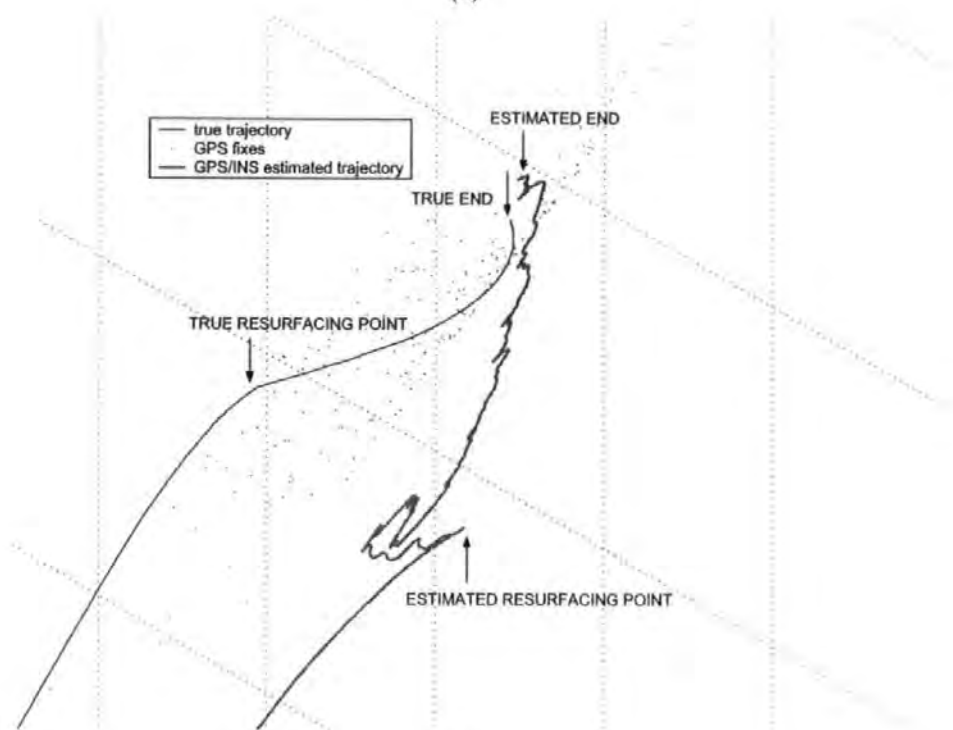


Figure 5.17: True trajectory, GPS fixes and GPS/INS using yaw produced by sensor-1 only



(a)



(b)

Figure 5.18: (a) Initial true trajectory, GPS fixes and GPS/INS estimated trajectory using yaw produced by sensor-1 only, (b) final true trajectory, GPS fixes and GPS/INS estimated trajectory using yaw produced by sensor-1 only

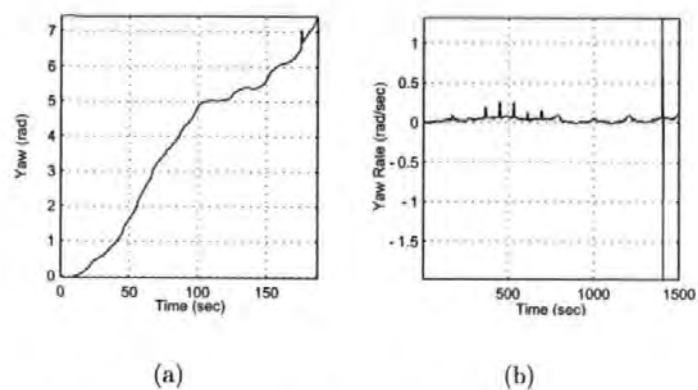


Figure 5.19: (a) Yaw sensor-2, (b) yaw rate sensor-2

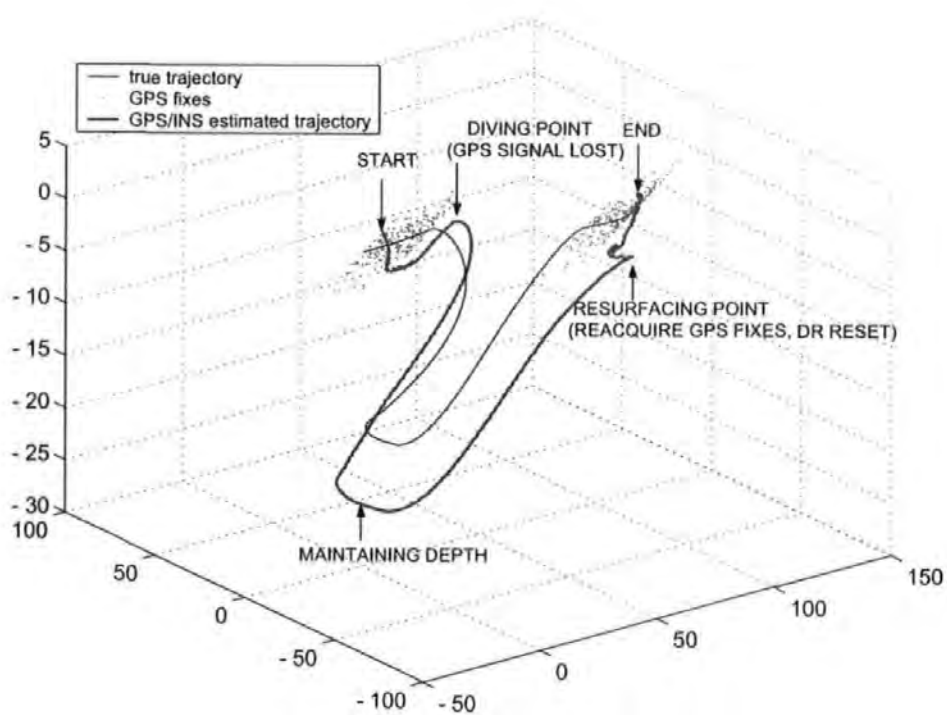


Figure 5.20: True trajectory, GPS fixes and GPS/INS using yaw produced by sensor-2 only

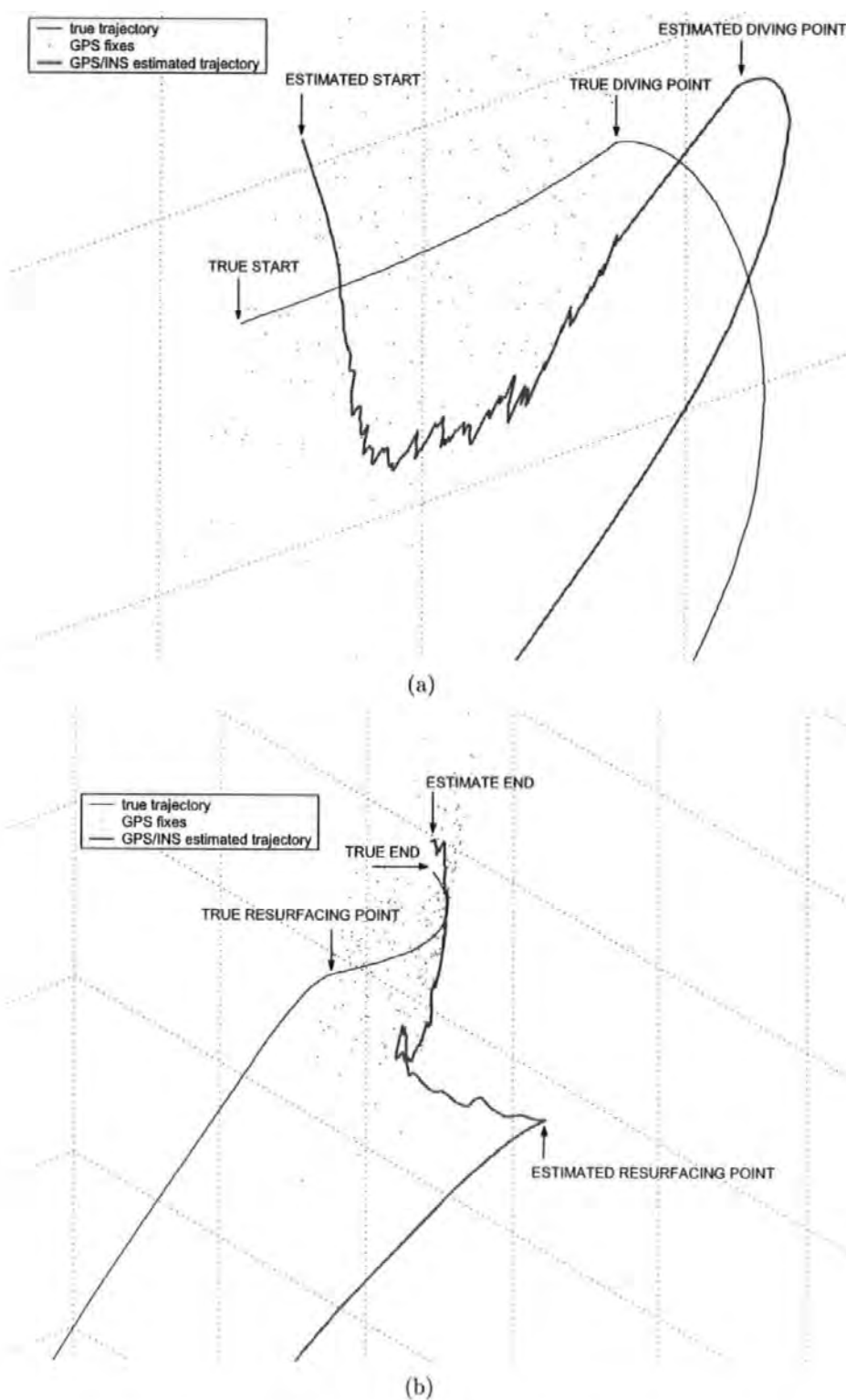


Figure 5.21: (a) Initial true trajectory, GPS fixes and GPS/INS estimated trajectory using yaw produced by sensor-2 only, (b) final true trajectory, GPS fixes and GPS/INS estimated trajectory using yaw produced by sensor-2 only

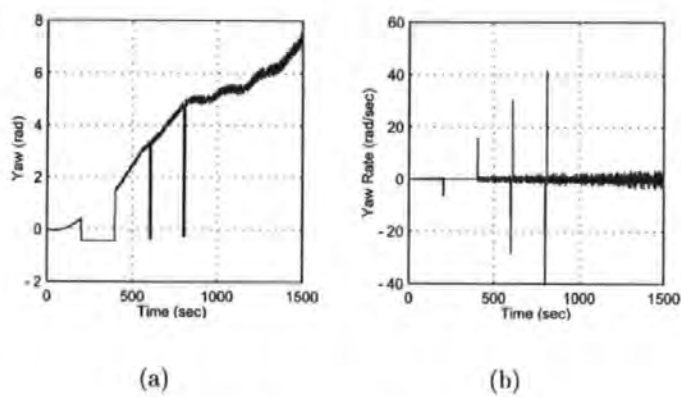


Figure 5.22: (a) Yaw sensor-3, (b) yaw rate sensor-3

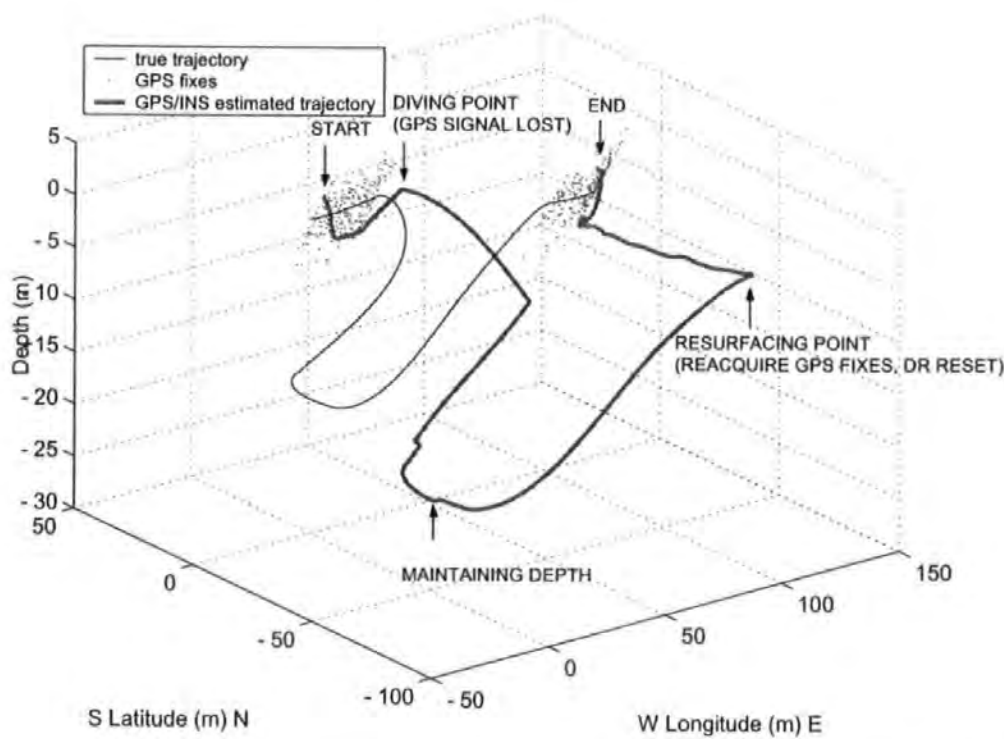


Figure 5.23: True trajectory, GPS fixes and GPS/INS using yaw produced by sensor-3 only

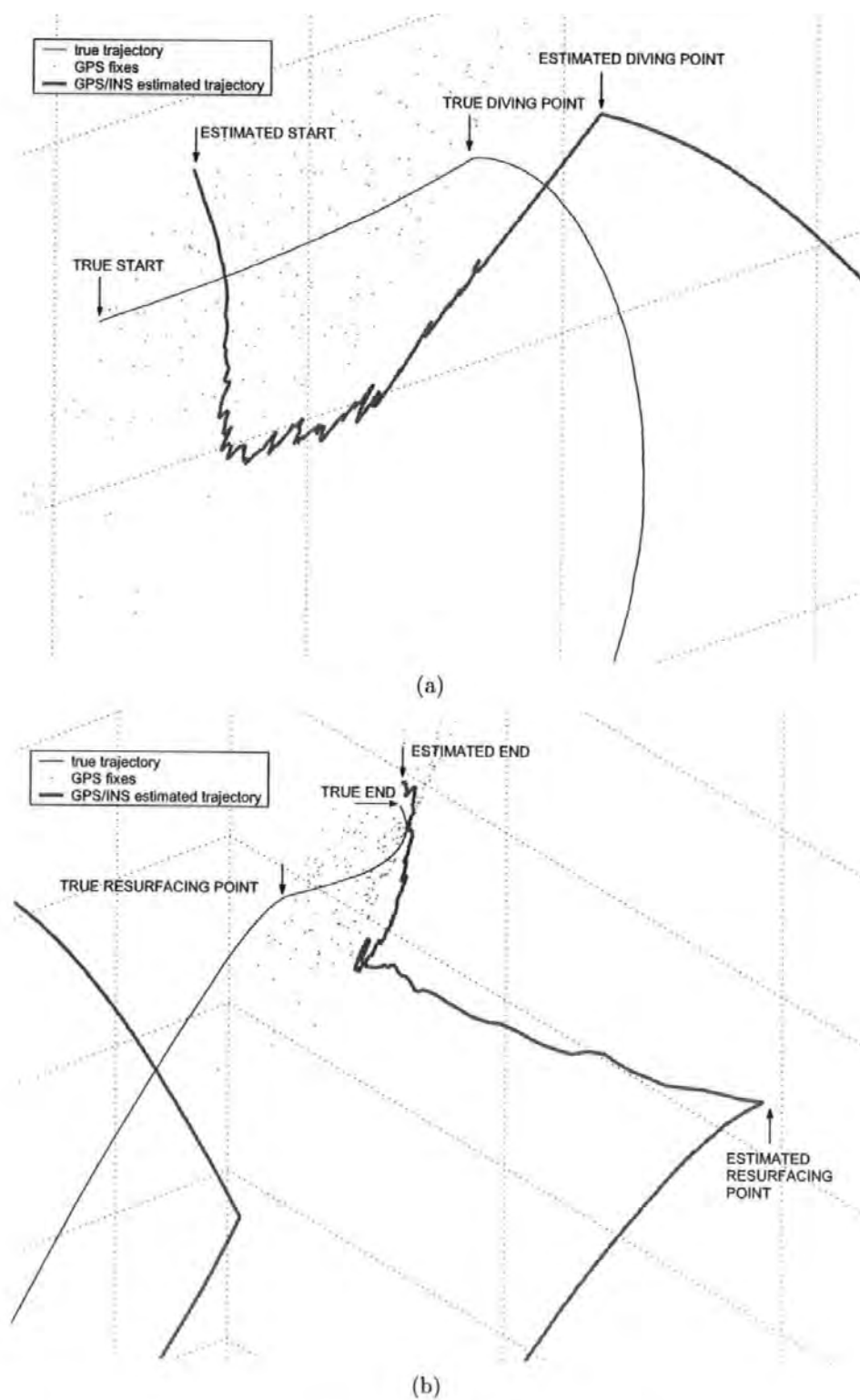


Figure 5.24: (a) Initial true trajectory, GPS fixes and GPS/INS estimated trajectory using yaw produced by sensor-3 only, (b) final true trajectory, GPS fixes and GPS/INS estimated trajectory using yaw produced by sensor-3 only

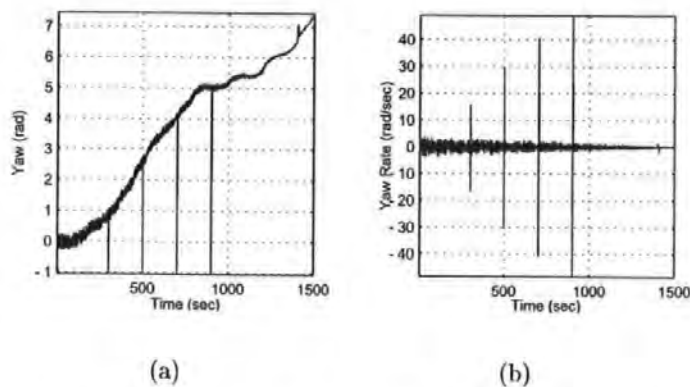


Figure 5.25: (a) Yaw sensor-4, (b) yaw rate sensor-4

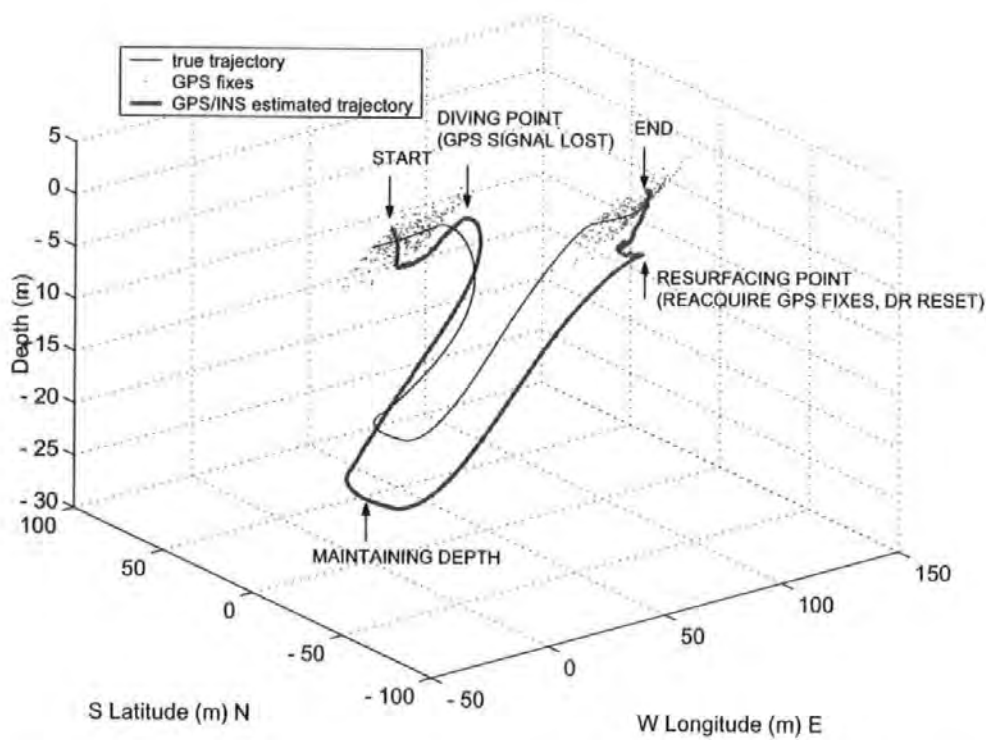


Figure 5.26: True trajectory, GPS fixes and GPS/INS using yaw produced by sensor-4 only

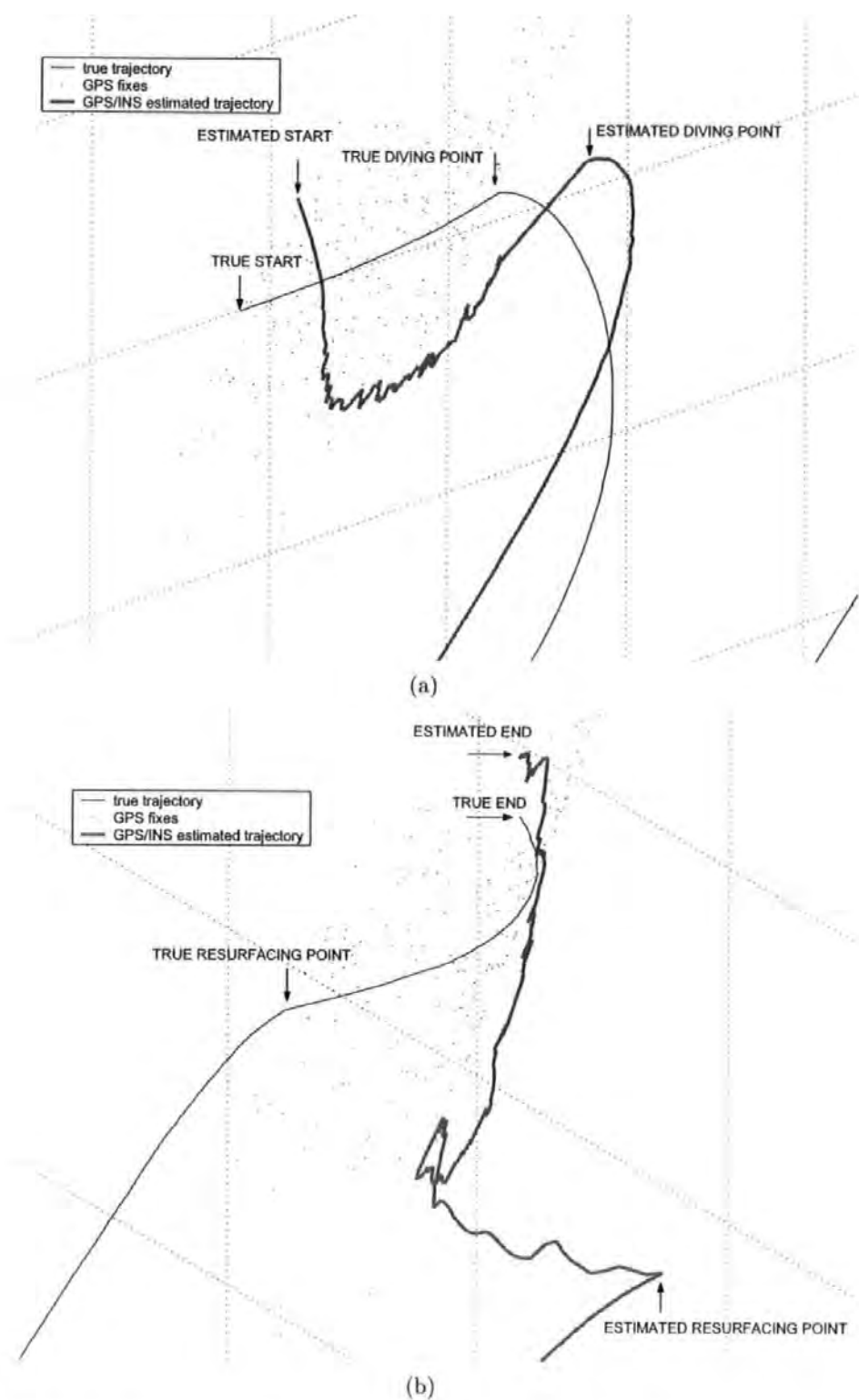


Figure 5.27: (a) Initial true trajectory, GPS fixes and GPS/INS estimated trajectory using yaw produced by sensor-4 only, (b) final true trajectory, GPS fixes and GPS/INS estimated trajectory using yaw produced by sensor-4 only

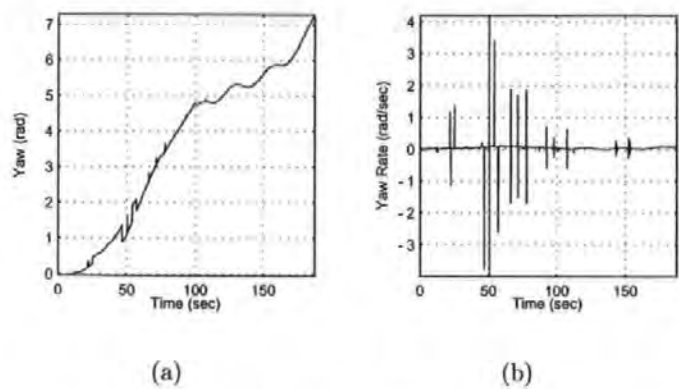


Figure 5.28: (a) Yaw sensor fused, (b) yaw rate sensor fused

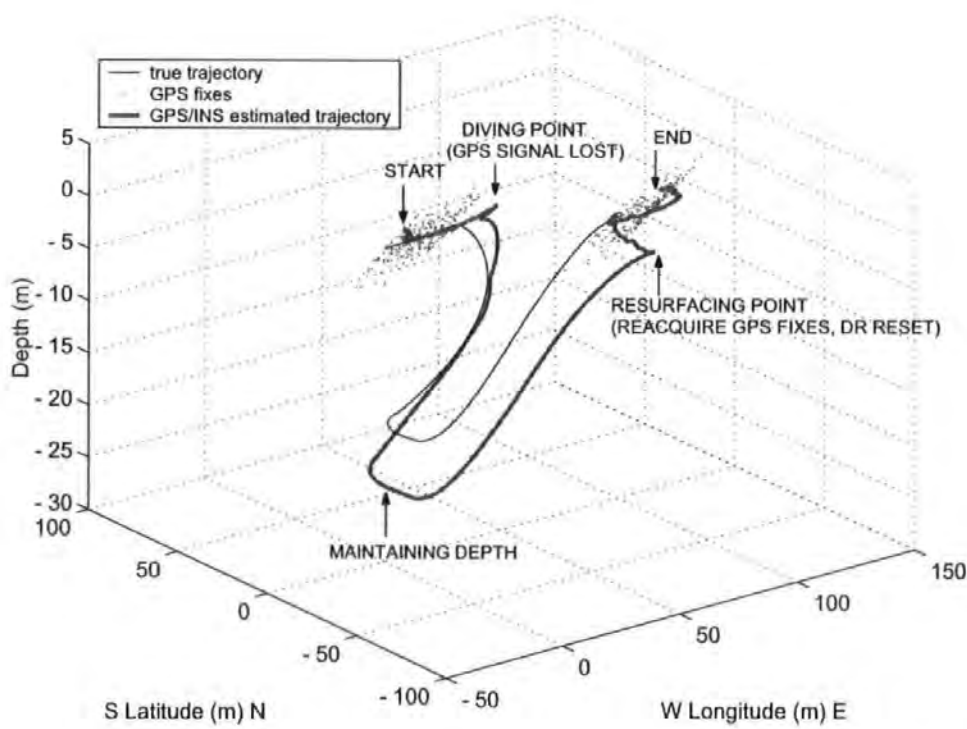


Figure 5.29: True trajectory, GPS fixes and GPS/INS using yaw produced by fused sensor

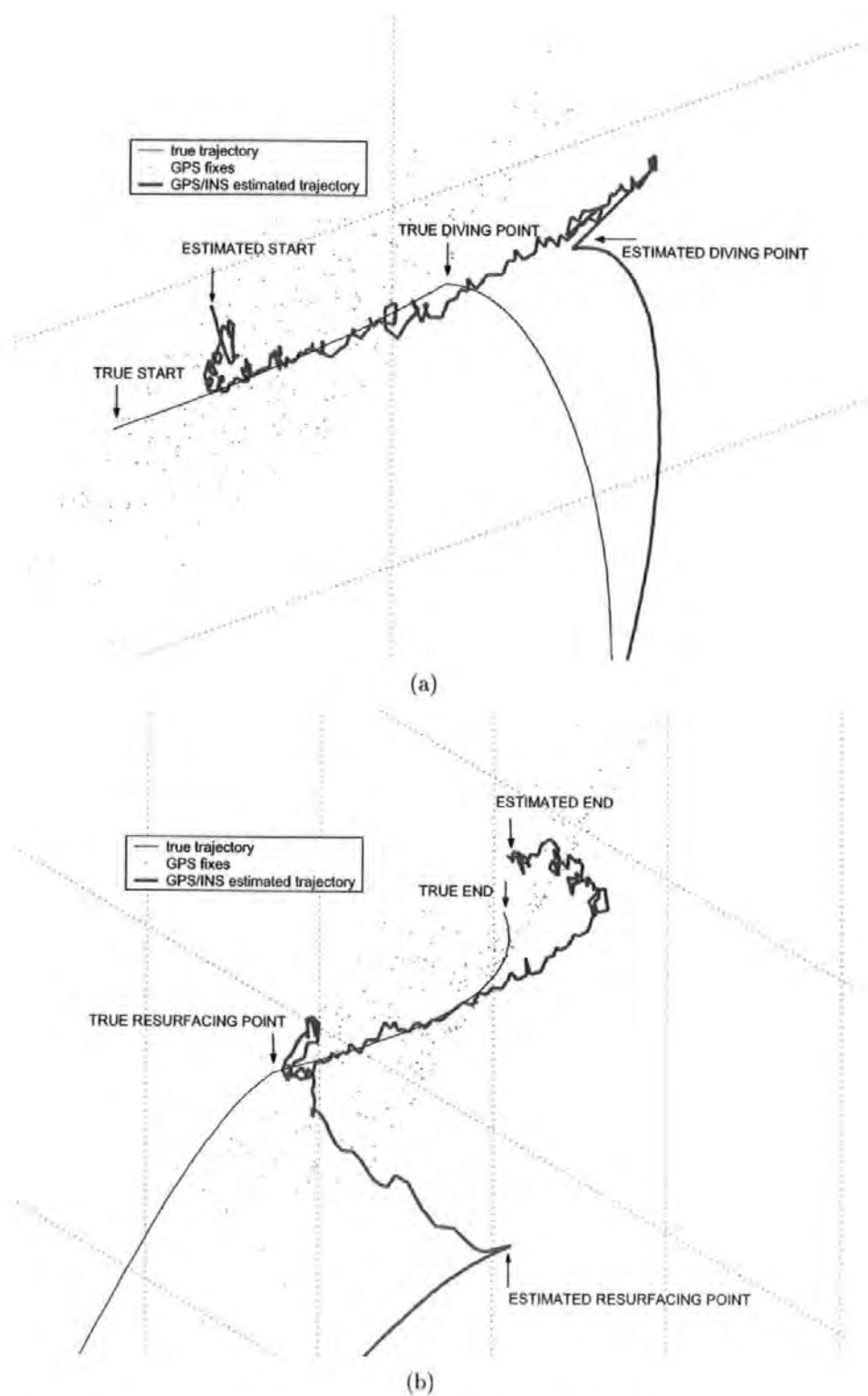


Figure 5.30: (a) Initial true trajectory, GPS fixes and GPS/INS estimated trajectory using yaw produced by fused sensor, (b) final true trajectory, GPS fixes and GPS/INS estimated trajectory using yaw produced by fused sensor

Figure 5.29 shows the trajectory produced using the fused yaw sensor and with the values of \mathbf{R} adjusted by FEKF. It is clear as presented in Figure 5.30(a), that as the assumed values of \mathbf{R} for the longitude and latitude quite low, $10(m)^2$, compared to the true ones, which are simulated to be $225(m)^2$ and $100(m)^2$, the FEKF estimation process put initial weight more on the GPS fixes than the dead reckoning solutions. However, as the filter learns the true nature of \mathbf{R} of these quantities, the FEKF makes an appropriate adjustment by putting more weight on the dead reckoning solution than on the GPS fixes. It can also be observed how the filter learns the true value of $\mathbf{R}_{[3,3]}$, which is simulated to be $0.0001(m^2)$. This time the vehicle is not estimated to have depth larger than $0(m)^2$, as in the case with the trajectory using yaw data produced by sensor-1 - sensor-4. Figure 5.30(a) also shows the work of the reset mechanism. As before, once the vehicle is below the surface, the depth controller and the underwater image acquisition algorithms will work side by side to find objects of interest and to maintain a constant depth thereafter for a specific period of time (see Figure 5.29). The vehicle is then sent back to the surface to obtain GPS fixes used to reset the drift produced by the dead reckoning process during the underwater mission. This particular mechanism is shown in Figure 5.30(b). It is also clear here how the FEKF has learned the true nature of the \mathbf{R} values. It can be observed from Figure 5.30(b) how the FEKF algorithm puts extra confidence on the GPS fixes right after the vehicle reaches the surface. Soon afterwards however, the algorithm recognises the high level of noise inherent in the acquired GPS signals and put less confidence thereon. Small discrepancies still exist between the true end and the estimated end of the mission. However, it is clear that without the FEKF, the estimated end could easily coincide with the last GPS fix and cause a significant position error.

5.4 CONCLUDING REMARKS

This chapter extends the implementation of the adaptive Kalman filtering approach from pure simulation in Chapter 4 to pseudo real-time herein. The set of data used here were gathered during a real-time experiment of the actual *Hammerhead* vehicle. The sensors used for this purpose were discussed with special emphasis given to their

characteristics and how the physical behaviour of their surrounding environments can possibly affect these.

Two navigation scenarios have been considered to validate the proposed approach: 2D/surface and 3D/surface-depth scenario. In both scenarios, the data from the TCM2 electronic compass and IMU are fused with two other simulated sensors before being used in transforming data from the body to the NED co-ordinate frame, where integration between the INS and GPS data occurs. In the first scenario, as the vehicle operates on the surface only, the GPS data is available periodically and the proposed estimation process takes place between the GPS fixes. In the second scenario, the GPS fixes are available continuously when the vehicle operates on the surface, and the proposed estimation algorithm blends these data with the position solution produced by the dead reckoning method to find the best estimates of the vehicle's position. In this scenario, the vehicle uses only dead reckoning method during an underwater mission and the accumulated errors produced thereby is reset by GPS fixes the next time the vehicle gain access to their signals. It has been shown in both scenarios that the proposed algorithm has produced a significant improvement in accuracy and reliability of the navigation system of the vehicle.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 CONCLUSIONS

This thesis focuses on the investigation of multisensor data fusion method utilising a synergistic combination of fuzzy logic, genetic algorithms (both in single and multiobjective mode), and Kalman filtering techniques to provide enhanced accuracy of the *Hammerhead* AUV integrated GPS/INS. This work is the first known use of this particular hybrid technique and is thus considered as a major contribution in relation to AUV technology. It has been shown how the method is able to provide a significant improvement over the conventional Kalman filtering techniques in their capacity as estimators in an integrated GPS/INS.

The following conclusions can be drawn with respect to the adaptive Kalman filter (which is here in this thesis also referred to as FKF/FEKF) developed in this research:

- The replacement of the widely used standard form of Kalman filters with adaptive ones for GPS/INS applications should be considered for the following reason:
 - the requirement to have a complete *a priori* knowledge of the filter statistics, represented by the \mathbf{R} and \mathbf{Q} matrices, are relaxed in the adaptive Kalman filtering approach. Although the focus of the work in the thesis has been placed to the adaptation of the \mathbf{R} matrix, with the \mathbf{Q} made constant, adaptive Kalman filtering is still able to produce the anticipated enhancement on the overall GPS/INS solutions.
 - results from pure simulation and pseudo real-time implementation show

that the adaptive Kalman filtering techniques outperform their standard counterparts in various situations, *e.g.* when the vehicle operates on the surface with continuous or periodical access to the GPS signals, before and after the vehicle is sent to a certain depth to perform a particular mission.

- Numerical complexity introduced by the soft computing techniques to the standard Kalman filtering in making up the adaptivity of the filter is quite substantial. Therefore it is decided to implement the proposed adaptation mechanisms to a set of simulated data and to a set of real data collected during a real-time experiment. In both cases, the proposed adaptive filtering performs equally well. It should be noted however, that the algorithms are highly implementable in real time provided superior computing power is at one's disposal.
- The effectiveness of adaptive Kalman filtering, depends largely on several important factors such as the number of satellites that are currently being locked on to by the GPS receiver and their PDOP.
- The size of the sliding window over which the actual covariance of the innovation is computed plays an important part in the overall performance of the filter. It is determined empirically in such away so that it is large enough to capture the dynamic of slowly-varying covariance values or small enough to capture the dynamic of fast-varying covariance values.
- The membership functions found by MOGA depends largely on the parameters defined thereon. The larger the values of certain parameters can lead to a higher numerical complexity. The trade-off between the numerical complexity and the solutions that can be produced by MOGA in this thesis have been made cautiously so as to sufficiently good solutions can still be found without having to go through a complex numerical computation.
- Despite the fact that solutions produced by the single objective GA and MOGA are comparable in some cases, the latter is still preferable to the former as it can be designed to direct the optimization process to satisfy a certain number of conditions without compromising the overall performance of the algorithms.

It has also been observed that improvements in one or more objectives of can lead to a degradation in other objectives. This stems from the non-dominated nature of the MOGA solutions.

6.2 FUTURE WORK

The work in this thesis focuses on the adaptation of \mathbf{R} . Consequently, the next logical step is to make the value of \mathbf{Q} adaptive. The \mathbf{Q} in this thesis are determined carefully using an iterative simulation process.

To use other sensor outputs, like the magnetic distortion alarm from the TCM2 electronic compass, PDOP and number of satellites from the GPS as part of the adaptation of the covariance matrices .

Chapter 2 discusses numerous algorithms for combining the information from various sensors and navigation aids for use in AUV navigation system. These have also been enriched by the work in this thesis. However, relatively little analytical or quantitative work has been undertaken to establish a rationale for sensor selection. Nor has much work been done to quantify the relative contributions that individual navigation sensors make to the performance of various navigation systems performance. As the usage of AUVs become more and more common as scientific and military exploration platforms, a tool is therefore considered to be necessary in this particular area of navigation system. The navigation systems that will be examined are similar to the previously developed *Hammerhead* AUV navigation system discussed in Chapters 4 and 5 and in Loebis *et al.* (2003b), Loebis *et al.* (2003c), Loebis *et al.* (2004a), Loebis *et al.* (2004b) in that they each utilize GPS position fixes and information from INS sensors. The differences between the systems lay primarily in *which* INS sensors the system utilize and the *accuracy* of the various sensor measurements. For example, many results are obtained for a system utilizing GPS position fixes, an accelerometer and a gyroscope. This sensor set was chosen as it is frequently sufficient to achieve moderate accuracy in an AUV navigation system. Here, the performance of this set of sensors is examined for various accelerometer and gyroscope performance levels and various GPS position fix accuracies. Other results are obtained for a system utilizing GPS position fixes, an accelerometer, a gyroscope and an electronic compass. The

performance of this system is examined for various GPS position fix accuracies and for a range of compass errors. Still other results can be obtained for a system operating in a littoral water utilizing GPS position fixes, an accelerometer, a gyroscope, and a velocity estimator (Loebis *et al.*, 2003a) from the VNS. The quantitative results directly divulge that individual navigation sensor error parameters have on navigation system performance. These quantitative results should therefore be beneficial for identifying the most *cost-effective* navigation system designs. ***It is important to note that although the analysis techniques in this work were developed for AUVs, they can effortlessly be applied to other autonomous vehicles, which are employed in the aerospace, underground and land environments. Thus the post-processing algorithms are valuable as generic practical tools for all types of navigation system design.***

To achieve the aim of the proposed future work, the following steps need to be undertaken:

- Step 1. To develop sensor error models using first and second order Markov processes.*
- Step 2. To derive sensitivity analysis equations for a Kalman filter and the corresponding adaptation mechanism using soft computing methodology*
- Step 3. To derive sensitivity analysis equations for a Kalman smoother and the corresponding adaptation mechanism using soft computing methodology*
- Step 4. To validate the post-processing analysis tool by undertaking full scale trials.*

It is felt that major contributions to knowledge will be forthcoming from this post-processing analysis tool.

APPENDIX A

PUBLICATIONS

1. Loebis, D., Naeem, W., Sutton, R. and Chudley, J. (2004). The Navigation, Guidance and Control of the *Hammerhead* Autonomous Underwater Vehicle, (To appear in: *Advances in Unmanned Marine Vehicle* (Roberts, G. N. and Sutton, R. (Ed)). Peter Peregrinus Ltd., Herts.)
2. Loebis, D., Sutton, R., Chudley, J. and Naeem, W. (2004). Adaptive Tuning of a Kalman Filter via Fuzzy Logic for an Intelligent AUV Navigation System. *Control Engineering Practice*, **12** (12), pp. 1531-1539.
3. Loebis, D., Sutton, R. and Chudley, J. (2004). A Fuzzy Kalman Filter Optimized Using a Multiobjective Algorithm for Enhanced Autonomous Underwater Vehicle Navigation. *Proceedings of the Institution of Mechanical Engineers Part M*, **218** (M1), pp. 53-69.
4. Loebis, D., Sutton, R. and Chudley, J. (2004). A Soft Computing Method for an AUV Navigation System with Pseudo-Real-Time Applicability. *Proc. 2004 IFAC Conference on Control Applications in Marine Systems*, Ancona, Italy, pp. 421-426.
5. Loebis, D., Sutton, R., Chudley, J., Dalglish, F. R. and Tetlow, S. (2004). The Application of Soft Computing Techniques to an Integrated Navigation System of an AUV. *Proc. 5th IFAC Symposium on Intelligent Autonomous Vehicles*, Lisbon, Portugal, MA-3-2 (CD-ROM Preprints)
6. Loebis, D., Chudley, J. and Sutton, R. (2003). A Fuzzy Kalman Filter Optimized Using a Genetic Algorithm for Accurate Navigation of an Autonomous Underwater Vehicle. *Proc. 6th IFAC Conference on Manoeuvring and Control of Marine Craft*, Girona, Spain, pp. 19-24.

7. Loebis, D., Dalglish, F. R., Sutton, R., Tetlow, S., Chudley, J., and Alwood, R. L. (2003). An Integrated Approach in the Design of a Navigation System for an AUV. *Proc. 6th IFAC Conference on Manoeuvring and Control of Marine Craft*, Girona, Spain, pp. 329-334.
8. Loebis, D., Chudley, J. and Sutton, R. (2003). A Fuzzy Kalman Filter for Accurate Navigation of an Autonomous Underwater Vehicle. *A Proceedings Volume from the IFAC Workshop on Guidance and Control of Underwater Vehicles (ISBN: 0080442021)*, Newport, South Wales, UK, pp. 157-162.
9. Loebis, D., Sutton, R. and Chudley, J. (2002). Review of Multisensor Data Fusion Techniques and Their Application to Autonomous Underwater Vehicle Navigation. *Journal of Marine Engineering and Technology*, A1, pp. 3-14. (This was given the Stanley Gray Award for the most worthy Offshore Technology paper in the journal during 2001/2002).



Adaptive tuning of a Kalman filter via fuzzy logic for an intelligent AUV navigation system

D. Loebis*, R. Sutton, J. Chudley, W. Naeem

Marine and Industrial Dynamic Analysis Research Group, School of Engineering, The University of Plymouth, Drake Circus, Plymouth PL4 8AA, UK

Received 17 August 2003; accepted 27 November 2003

Abstract

This paper describes the implementation of an intelligent navigation system, based on the integrated use of the global positioning system (GPS) and several inertial navigation system (INS) sensors, for autonomous underwater vehicle (AUV) applications. A simple Kalman filter (SKF) and an extended Kalman filter (EKF) are proposed to be used subsequently to fuse the data from the INS sensors and to integrate them with the GPS data. The paper highlights the use of fuzzy logic techniques to the adaptation of the initial statistical assumption of both the SKF and EKF caused by possible changes in sensor noise characteristics. This adaptive mechanism is considered to be necessary as the SKF and EKF can only maintain their stability and performance when the algorithms contain the true sensor noise characteristics. In addition, fault detection and signal recovery algorithms during the fusion process to enhance the reliability of the navigation systems are also discussed herein. The proposed algorithms are implemented to real experimental data obtained from a series of AUV trials conducted by running the low-cost *Hammerhead* AUV, developed by the University of Plymouth and Cranfield University.

© 2004 Elsevier Ltd. All rights reserved.

Keywords: Autonomous underwater vehicles; Navigation; Sensor fusion; Kalman filters; Extended Kalman filters; Fuzzy logic

1. Introduction

The development of autonomous underwater vehicles (AUVs) for scientific, military and commercial purposes in applications such as ocean surveying (Størkersen, Kristensen, Indreeide, Seim, & Glancy, 1998), unexploded ordnance hunting (Wright et al., 1996) and cable tracking and inspection (Asakawa, Kojima, Kato, Matsumoto, & Kato, 2000) requires the corresponding development of navigation systems. Such systems are necessary to provide knowledge of vehicle position and attitude. The need for accuracy in such systems is paramount: erroneous position and attitude data can lead to a meaningless interpretation of the collected data or even to a catastrophic failure of an AUV.

A growing number of research groups around the world are developing integrated navigation systems utilising inertial navigation system (INS) and global positioning system (GPS) (Gade & Jalving, 1999;

Grenon, An, Smith, & Healey, 2001; Yun et al., 1999). However, few of these works make explicit the essential need for fusion of several INS sensors that enable the users to maintain the accuracy or even to prevent a complete failure of this part of the navigation system, before being integrated with the GPS. Kinsey and Whitcomb (2003), e.g. use a switching mechanism to prevent a complete failure of the INS. Although simple to implement, the approach may not be appropriate to use to maintain a certain level of accuracy.

Several estimation methods have been used in the past for multisensor data fusion and integration purpose (Loebis, Sutton, & Chudley, 2002). To this end, simple/extended Kalman filter (SKF/EKF) and their variants have been popular methods in the past and interest in developing the algorithms has continued to the present day. However, a significant difficulty in designing a SKF/EKF can often be traced to incomplete a priori knowledge of the process covariance matrix (Q) and measurement noise covariance matrix (R). In most practical applications, these matrices are initially estimated or even unknown. The problem here is that the optimality of the estimation algorithm in the SKF/EKF

*Corresponding author. Tel.: +44-1752232633; fax: +44-1752232638.

E-mail address: d.loebis@plymouth.ac.uk (D. Loebis).

setting is closely connected to the quality of a priori information about the process and measurement noise (Mehra, 1970, 1971). It has been shown that insufficiently known a priori filter statistics can reduce the precision of the estimated filter states or introduces biases to their estimates. In addition, incorrect a priori information can lead to practical divergence of the filter (Fitzgerald, 1971). From the aforementioned, it may be argued that the conventional SKF/EKF with fixed R and/or Q should be replaced by an adaptive estimation formulation as discussed in the next section.

2. The adaptive tuning of Kalman filter algorithm

In the past few years, only a few publications in the area of adaptive Kalman filtering can be found in the literature. The two major approaches that have been proposed for adaptive Kalman filtering are multiple model adaptive estimation (MMAE) and innovation adaptive estimation (IAE). Although the implementation of these approaches are quite different, they both share the same concept of utilising new statistical information obtained from the innovation (or residual) sequence. In both cases, the innovation inn_k at sample time k is the difference between the real measurement z_k , received by the filter and its estimated (predicted) value \hat{z}_k . The predicted measurement is the projection of the filter predicted states \hat{x}_k onto the measurement space through the measurement design matrix H_k . Innovation represents additional information available to the filter as a result of the new measurement z_k . The occurrence of data with statistics different from the a priori information will first show up in the innovation vector. For this reason, the innovation sequence represents the information content in the new observation and is considered the most relevant source of information to the filter adaptation. Interested readers can refer to (Kailath, 1968a, b, 1970) for a more detailed discussion of the innovation sequence and its use in linear filter theory.

In the MMAE approach, a bank of Kalman filters runs in parallel (Magill, 1965; Hanlon & Maybeck, 2000) or with a gating algorithm (Chao, Bishop, & Ghough, 1997) under a different model for the statistical filter information matrices, i.e. Q and R . In the IAE approach (Mehra, 1970, 1971), the Q and R matrices themselves are adapted as measurements evolve with time. In this paper, the IAE approach coupled with fuzzy logic techniques with membership functions designed using heuristic methods is used to adjust the R matrix of both the SKF and EKF. The proposed algorithms in this paper are implemented using a set of experimental data obtained from the Hammerhead AUV trials conducted in July 2003 at Roadford Reservoir, Devon, UK. Initial work using purely simulated data on the proposed algorithms can be found in Loebis, Sutton,

and Chudley, 2003b and Loebis, Chudley, and Sutton, 2003a.

2.1. Fuzzy simple Kalman filter

In this section, an on-line innovation-based adaptive scheme of the SKF to adjust the R matrix employing the principles of fuzzy logic is presented. The fuzzy logic is chosen mainly because of its simplicity. This motivates the interest in the topic, as testified by related papers which have been appearing in the literature (Kobayashi, Cheok, Watanabe, & Muneka, 1998; Jetto, Longhi, & Vitali, 1999; Loebis et al., 2003a, b). The fuzzy logic simple Kalman filter (FSKF) proposed herein and fuzzy logic extended Kalman filter (FEKF) discussed in Section 4 are based on the IAE approach using a technique known as covariance matching (Mehra, 1970). The basic idea behind the technique is to make the actual value of the covariance of the innovation sequences match its theoretical value.

The actual covariance is defined as an approximation of the inn_k sample covariance through averaging inside a moving estimation window of size M (Mohamed & Schwarz, 1999) which takes the following form:

$$\hat{C}_{inn_k} = \frac{1}{M} \sum_{j=j_0}^k inn_k inn_k^T, \quad (1)$$

where $j_0 = k - M + 1$ is the first sample inside the estimation window. An empirical experiment is conducted to choose the window size M . From experimentation, it was found that a good size for the moving window in (1) is 15.

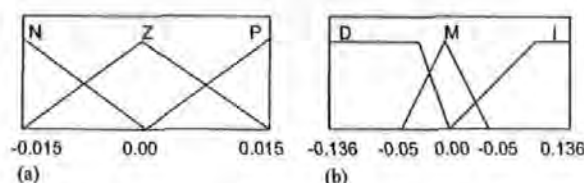
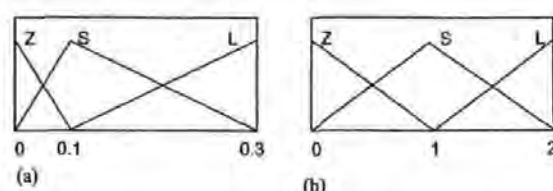
The theoretical covariance of the innovation sequence is defined as

$$S_k = H_k P_k^- H_k^T + R_k. \quad (2)$$

The logic of the adaptation algorithm using covariance matching technique can be qualitatively described as follows. If the actual covariance value \hat{C}_{inn_k} is observed, whose value is within the range predicted by theory S_k and the difference is very near to zero, this indicates that both covariances match almost perfectly and only a small change is needed to be made on the value of R . If the actual covariance is greater than its theoretical value, the value of R should be decreased. On the contrary, if \hat{C}_{inn_k} is less than S_k , the value of R should be increased. This adjustment mechanism lends itself very well to being dealt with using a fuzzy-logic approach based on rules of the kind:

$$\text{IF } \langle \text{antecedent} \rangle \text{ THEN } \langle \text{consequent} \rangle, \quad (3)$$

where antecedent and consequent are of the form $\chi \in O_i$, $\kappa \in L_i$, $i = 1, 2, \dots$, respectively, where χ and κ are the input and output variables, respectively, and O_i and L_i are the fuzzy sets.

Fig. 1. Membership function of: (a) δ_k , and (b) ΔR_k .Fig. 2. Membership function of: (a) $|\delta_k|$, and (b) R_k .

To implement the above covariance matching technique using the fuzzy logic approach, a new variable called δ_k , is defined to detect the discrepancy between \hat{C}_{Inn_k} and S_k . It is important to note that in this particular application, \hat{C}_{Inn_k} and S_k are constrained to be diagonal matrices. The following three fuzzy rules of the kind (3) are used:

IF $\langle \delta_k \approx 0 \rangle$ THEN $\langle R_k$ is unchanged \rangle , (4)

IF $\langle \delta_k > 0 \rangle$ THEN $\langle R_k$ is decreased \rangle , (5)

IF $\langle \delta_k < 0 \rangle$ THEN $\langle R_k$ is increased \rangle . (6)

Thus R is adjusted according to

$$R_k = R_{k-1} + \Delta R_k, \quad (7)$$

where ΔR_k is added or subtracted from R at each instant of time. Here δ_k is the input to the fuzzy inference system (FIS) and ΔR_k is the output.

On the basis of the above adaptation hypothesis, the FIS can be implemented using three fuzzy sets for δ_k ; N =Negative, Z =Zero and P =Positive. For ΔR_k the fuzzy sets are specified as I =Increase, M =Maintain and D =Decrease. The membership functions of these fuzzy sets which are designed using a heuristic approach are shown in Fig. 1.

2.2. Sensor fault diagnostic and recovery algorithm

In addition to the adaptation procedure, the FSKF has been equipped with the sensor fault diagnostic and recovery algorithm as proposed by Escamilla-Ambrosio & Mort (2001). The basic idea behind this algorithm is that the amplitude of the actual value of the Inn_k and its theoretical value ($\sqrt{S_k}$) for a sensor without any fault must be around 1, but it increases abruptly if a transient or persistent fault is present in the measurement data. For this purpose a variable $InnC_k$ is defined as

$$InnC_k = \frac{|Inn_k|}{\sqrt{S_k}}. \quad (8)$$

Thus, if the value of $InnC_k$ is greater or equal than a threshold (α) then a transient fault is declared and Inn_k is assigned a value of 0. If $InnC_k$ is still greater than α for an instant of time, the persistent fault is declared and Inn_k is assigned a value of $\sqrt{S_k}$ multiplied by a random

Table 1
Fuzzy rule base FLO

$ \delta_k $	R_k		
	Z	S	L
Z	G	G	AV
S	G	AV	P
L	AV	P	P

number. From experimentation, it was found that the good value of α is 1.2.

2.3. Fuzzy logic observer

To monitor the performance of a FSKF, another FIS called the fuzzy logic observer (FLO) (Escamilla-Ambrosio & Mort, 2001) is used. The FLO assigns a weight or degree of confidence denoted as c_k , a number on the interval $[0,1]$, to the FSKF state estimate. The FLO is implemented using two inputs: the values of $|\delta_k|$ and R_k . The membership functions of these variables were found using a heuristic method that produced a non-symmetrical shape for $|\delta_k|$ and a symmetrical shape for R_k are shown in Fig. 2.

The fuzzy labels for the membership functions: Z =Zero, S =Small and L =Large. Three fuzzy singletons are defined for the output c_k and are labelled as G =Good, AV =Average and P =Poor with values 1, 0.5 and 0, respectively. The basic heuristic hypothesis for the FLO is as follows: if the value of $|\delta_k|$ is near to zero and the value of R_k is near to zero, then the FSKF works almost perfectly and the state estimate of the FSKF is assigned a weight near 1. On the contrary, if one or both of these values increases far from zero, it means that the FSKF performance is degrading and the FLO assigns a weight near 0. Table 1 gives the complete fuzzy rule base of each FLO.

3. Fusion of INS sensor data

In this section, the FSKF algorithm is applied to the linear heading model of the *Hammerhead* AUV. Fig. 3(a) shows the vehicle before leak testing and

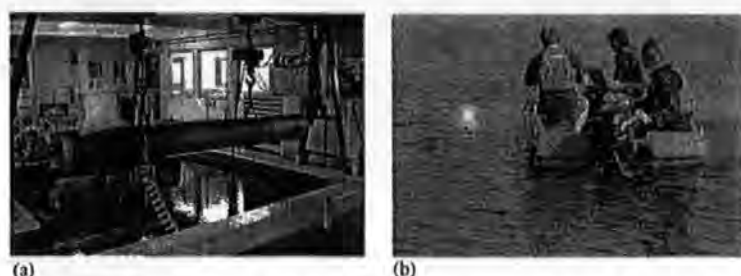


Fig. 3. The *Hammerhead* AUV: (a) before leak testing and ballasting, and (b) during a system identification trial.

ballasting in the laboratory test tank and Fig. 3(b) shows the vehicle during a heading system identification trial. Inputs to the rudder of the vehicle are sent by the user from a laptop through an umbilical cable. Thus in reality it was travelling in a semi-autonomous model for that specific trial. The drag effect of the cable is considered to be negligible. An electronic compass and an inertial measurement unit (IMU) on board the vehicle are used to capture the corresponding responses.

The system matrix (A), input matrix (B), and output vector (H) of the linear discrete state space model (see Appendix A) obtained from the trial data are, respectively, $A = \begin{bmatrix} 0 & 1 \\ -0.98312 & 1.9831 \end{bmatrix}$, $B = \begin{bmatrix} -0.0031961 & -0.0036115 \end{bmatrix}$, $H = \begin{bmatrix} 1 & 0 \end{bmatrix}$, with the yaw and delayed-yaw as the component of the states.

This model is assumed to be sufficiently accurate to represent the dynamics of the vehicle, and for this reason, any output produced by the model after being excited by an input, can be considered as an actual output value. This assumption also motivates the use of the model output as a reference in measuring the performance of the FSKF algorithms.

To test the FSKF algorithms, real data obtained from the electronic compass and IMU (Fig. 5), as a response to the input shown in Fig. 4, are fused together with two sets of simulated data. To produce the simulated data, the noise in Figs. 6(a) and (b) are simply added to the electronic compass and IMU real data, respectively. A possible real-time scenario that can result in the noise with the characteristic shown in Fig. 6(a) is that the second electronic compass is located in close proximity to the propeller DC motor whose internal temperature increases with time and affects the sensor ambient temperature. A similar scenario can also be considered to occur when the second IMU is located in close proximity to the front hydroplane stepper motor whose initial internal temperature is high and settles down after sometime. This particular scenario can result in the noise characteristic as shown in Fig. 6(b). The initial condition of the states are $\begin{bmatrix} 0 & 0 \end{bmatrix}^T$, $P_0 = 0.01I_2$ (see Appendix A) and $Q = \text{diag}(0, 0.1725 \times 10^{-7})$. The actual value of R for each sensor is assumed unknown, but its initial value is selected as 1. Simulation results are shown in the next section.

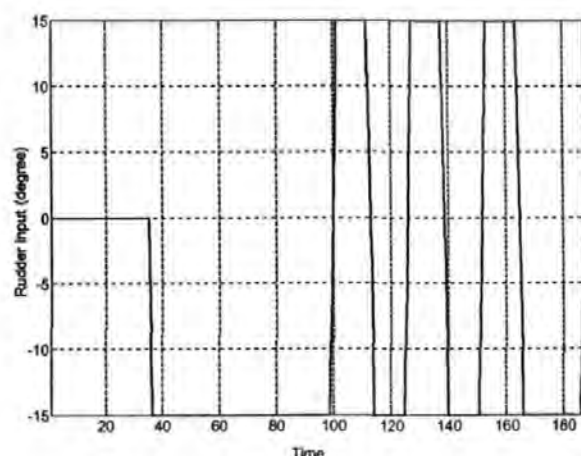


Fig. 4. Input rudder.

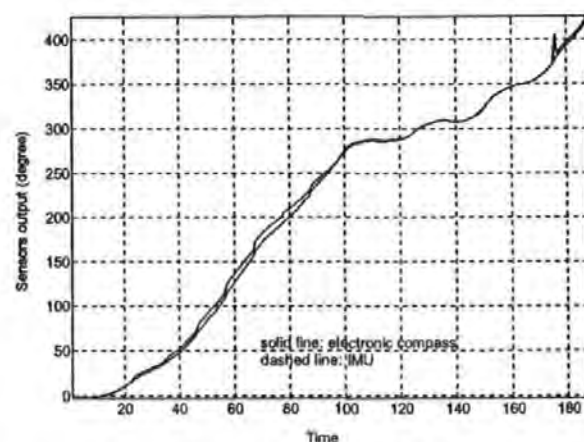


Fig. 5. Real electronic compass and IMU output.

3.1. Simulation result

Figs. 7 and 8 are the simulation results showing the response of the *Hammerhead* AUV observed by electronic compass and IMU, respectively, while Figs. 9 and 10 by sensor 3 and 4, which are the output of the two former sensors added with uniform noise increasing and

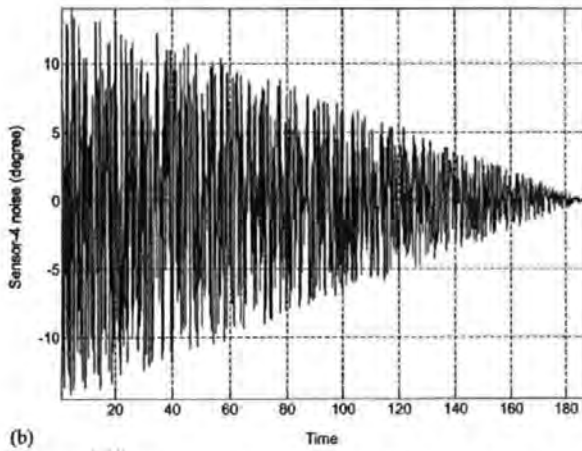
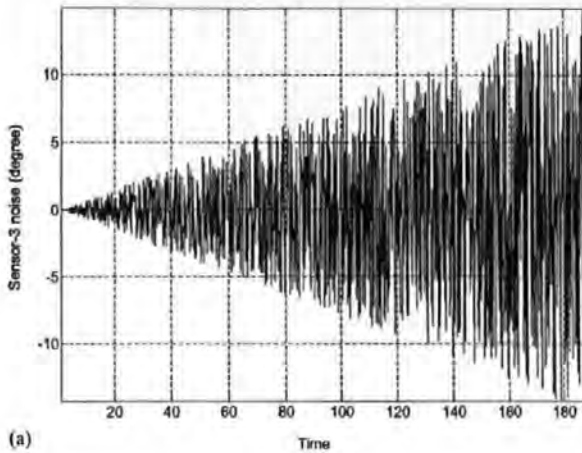


Fig. 6. (a) Added noise profile to the electronic compass data, and (b) to the IMU data.

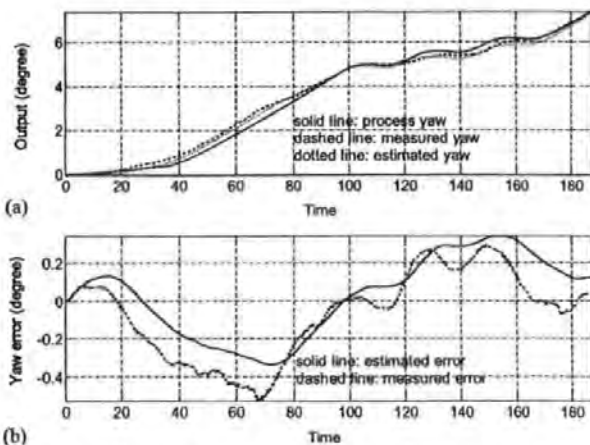


Fig. 7. (a) Process, measured and estimated yaw output, and (b) measured and estimated yaw error of electronic compass.

decreasing with time, respectively. Figs. 7 and 8 show improvements in the level of error produced by the proposed FSKF algorithms as compared to direct

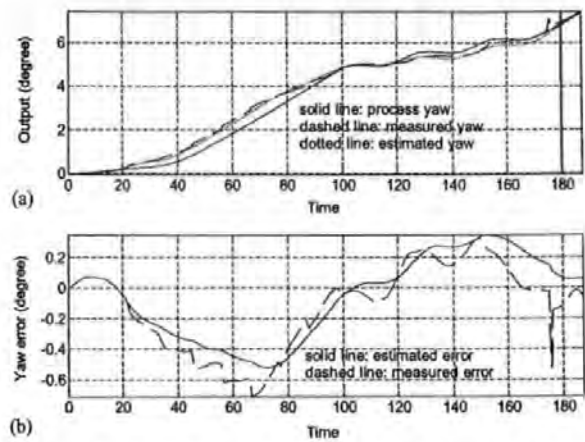


Fig. 8. (a) Process, measured and estimated yaw output, and (b) measured and estimated yaw error of IMU.

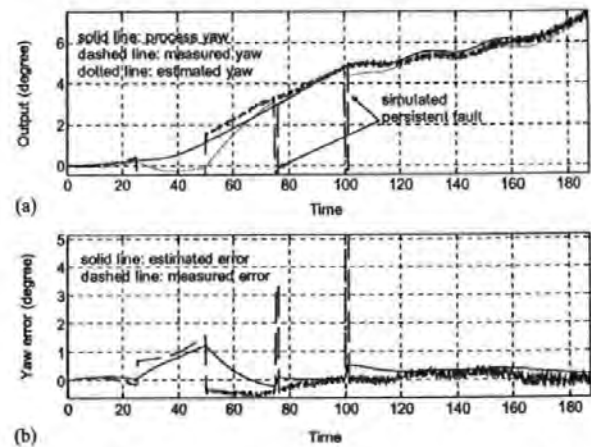


Fig. 9. (a) Process, measured and estimated yaw output, and (b) measured and estimated yaw error of sensor 3 (electronic compass + simulated noise).

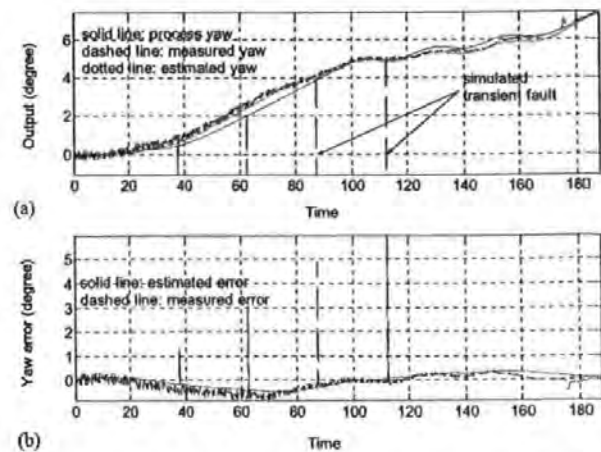


Fig. 10. (a) Process, measured and estimated yaw output, and (b) measured and estimated yaw error of sensor 4 (IMU + simulated noise).

measurements from the sensors. Apart from the improvements in the level of error, Figs. 9 and 10 also show how the proposed algorithms have detected transient and persistent faults in the sensors (see Section 2.2) and made an appropriate recovery.

To fuse the estimated yaw, a centre of gravity method is used

$$\hat{z}_k = \frac{\sum_{i=1}^n \hat{z}_{k_i} c_{k_i}}{\sum_{i=1}^n c_{k_i}}, \quad (9)$$

where \hat{z}_{k_i} is the output of the i th FSKF ($i = 1, 2, 3, 4$) and c_{k_i} is the respective weight at instant time k . Fig. 11 shows the comparison of the actual and the fused estimated yaw. It is clear, by comparing Fig. 11 and Figs. 7–10 that an improvement has been achieved by fusing the estimated yaw.

Finally, the following performance measure are adopted for comparison purposes:

$$J_{zv} = \sqrt{\frac{1}{n} \sum_{k=1}^n (z_{ak} - z_k)^2}, \quad (10)$$

$$J_{ze} = \sqrt{\frac{1}{n} \sum_{k=1}^n (z_{ak} - \hat{z}_k)^2}, \quad (11)$$

where z_{ak} is the actual value of the yaw, z_k is the measured yaw, \hat{z}_k is the estimated yaw at an instant of time k and n the number of samples (Table 2).

A close look on the J_{zv} and J_{ze} on Table 2 of each sensor indicates that the FSKF has improved the accuracy of the heading information. The result of fusing the estimated data has shown a further improvement. A slight offset shown by the final fusion result might be caused by an inaccurate model of the process noise (see Appendix A) and its covariance (Q). Adaptation of these parameters is the topic of a future investigation. It should also be noted that from a theoretical point of view, the analysis of the stability of

Table 2
Comparison of performance

Sensor	Performance	
	J_{zv} (deg)	J_{ze} (deg)
Electronic compass	13.4050	12.1170
IMU	17.3507	15.8216
Electronic compass + noise	37.5725	23.6664
IMU + noise	22.0702	14.6159
Sensor fusion		11.9650

the FSKF needs to be investigated. However, this is not easily undertaken due to the use of the adaptation techniques used herein. Future work will address this issue more rigorously.

4. Integrated GPS/INS

Here, the fused estimated yaw obtained previously is treated as a single imaginary yaw sensor and used by other INS sensors to transform data from body co-ordinate to Earth co-ordinate frame where integration with GPS data is performed using a combination of fuzzy logic and EKF techniques and can be referred to as FEKF.

A continuous time model of the vehicle motion appropriate to this problem is taken to be

$$\dot{X}(t) = F(X(t)) + W(t), \quad (12)$$

$$Z(t) = H(X(t)) + V(t). \quad (13)$$

Denoted by $X(t) = [\lambda(t) \varphi(t) \psi(t) \vartheta(t) \zeta(t) v(t)]^T$ is the model states. $\lambda(t)$ and $\varphi(t)$ are the longitude and latitude of the AUV position in Earth co-ordinate frame which are obtained from a GPS receiver, $\psi(t)$ is the yaw angle obtained from the imaginary yaw sensor, $\vartheta(t)$ is yaw rate, $\zeta(t)$ and $v(t)$ are the surge and sway velocity, respectively.

In this system model, F and H are both continuous functions, continuously differentiable in $X(t)$. The $W(t)$ and $V(t)$ are both zero-mean white noise for the system and measurement models, respectively.

The model states are related through the following kinematically based set of functions ($F(X(t))$ in Eq. (12)):

$$\dot{\zeta}(t) = 0, \quad (14)$$

$$\dot{v}(t) = 0, \quad (15)$$

$$\dot{\psi}(t) = \vartheta(t), \quad (16)$$

$$\dot{\vartheta}(t) = 0, \quad (17)$$

$$\dot{\lambda}(t) = \zeta(t) \cos \psi(t) - v(t) \sin \psi(t), \quad (18)$$

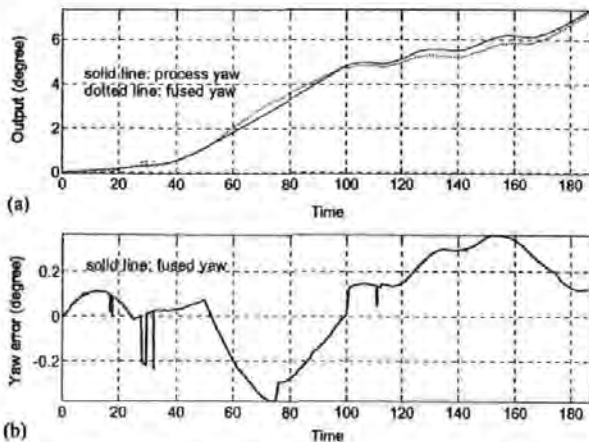


Fig. 11. (a) Process and estimated fused yaw output, and (b) fused yaw error.

$$\dot{\phi}(t) = \zeta(t) \sin \psi(t) + v(t) \cos \psi(t). \quad (19)$$

The output measurements are related through the states by the following output matrix:

$$H(X(t)) = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (20)$$

when GPS signal is available, and when it is not,

$$H(X(t)) = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \quad (21)$$

To obtain an EKF with an effective state prediction equation in a simple form, the continuous time model of (14)–(21) has been linearised about the current state estimates, producing

$$\Phi(t) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\zeta(t) \sin \psi(t) - v(t) \cos \psi(t) & 0 & \cos \psi(t) & -\sin \psi(t) \\ 0 & 0 & \zeta(t) \cos \psi(t) - v(t) \sin \psi(t) & 0 & \sin \psi(t) & \cos \psi(t) \end{bmatrix} \quad (22)$$

and Γ is a matrix identical as in either (20) or (21). Subsequent discretisation with period $T = 0.125$ s of the linearised model results in an EKF algorithm similar to the SKF algorithms in Appendix A (where Φ and Γ are equivalent to A and H), only this time the Φ matrix is updated at every iteration. The initial conditions are $P_0 = 0.01 I_6$ and Q is made constant as $\text{diag}(10, 10, 1, 0.1, 0.1, 0.1)$. The actual value of R is assumed unknown but its initial value is selected as $\text{diag}(1000, 1000, 5, 1, 2, 2)$.

The FSKF algorithm from Section 2 is then implemented, only this time the adaptation of the (i, i) th element of R_k is made in accordance with the (i, i) th element of ΔR_k . Here a single-input–single-output FIS as shown in Fig. 1, is used sequentially to generate the correction factors for the elements in the main diagonal of R_k as the following:

$$R_k(i, i) = R_{k-1}(i, i) + \Delta R_k. \quad (23)$$

Fig. 12 shows the *Hammerhead* AUV trajectory obtained using GPS, dead reckoning using INS sensors (through double integration of the accelerometer data with respect to time) and integrated GPS/INS. As the initial value of R for both $\lambda(t)$ and $\phi(t)$ is 1000, the standard EKF algorithm puts less weight on the position obtained by GPS and more on the prediction

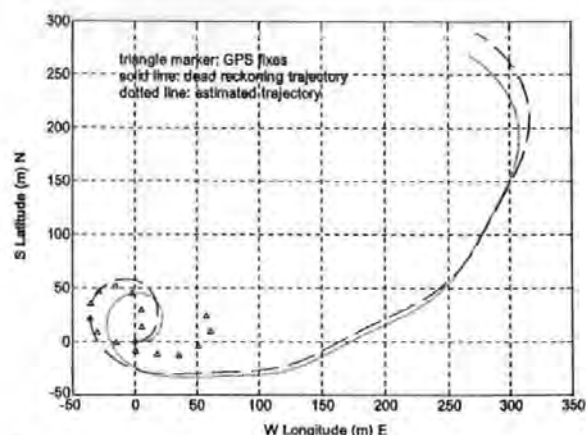


Fig. 12. AUV trajectory obtained using GPS, INS sensors (dead reckoning method) and GPS/INS using EKF without adaptation.

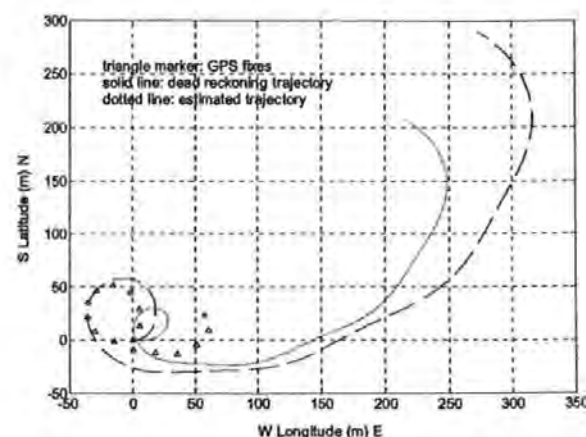


Fig. 13. AUV trajectory obtained using GPS, INS sensors (dead reckoning method) and GPS/INS using EKF with adaptation.

of position obtained from dead reckoning method (using INS sensor data). Fig. 13 shows that the R matrix has been adjusted accordingly and more weight is given to the GPS data, and therefore the estimated trajectory in the integrated INS/GPS is "pulled" a little bit further to the GPS trajectory. However, big discrepancies can still be appreciated between the integrated INS/GPS estimate with respect to the GPS fixes. There are several explanations to this erratic behaviour. The first possibility is that it is caused by the poor level of accuracy of the low-cost GPS being used in this particular application. It is important to note that the proposed algorithm has detected a persistent high actual covariance (\hat{C}_{Inm_k}) for both the $\lambda(t)$ and $\phi(t)$ throughout the trajectory. This results in insufficient weight being given to the GPS fixes in the FEKF and more on the position obtained by the dead reckoning. The second possibility is that the GPS receiver did not lock into a sufficient number of satellites

with a sufficiently small value of position dilution of precision that can provide the required level of accuracy. The use of a differential GPS receiver or a GPS receiver with a wide area augmentation system or a European Geostationary Navigation Overlay Service capability can be considered as a way forward to alleviate this problem.

5. Summary and conclusions

The problem with incomplete a priori knowledge of Q and R is considered. Within this paper, an adaptive Kalman filter approach, based on the filter innovation sequence coupled with fuzzy logic is discussed as an alternative for fusing INS sensor data and integrating INS/GPS position information. Implementation of this approach to the *Hammerhead* heading model, whose responses are measured with electronic compass, IMU and two additional sensors with different noise characteristics, has shown a promising result in improving the estimation of an individual SKF and EKF and enhancing the overall accuracy of the integrated INS/GPS.

Appendix A. Simple Kalman filter algorithms

Given a discrete-time controlled process described by the linear stochastic difference equations:

$$x_{k+1} = A_k x_k + B_k u_k + w_k, \quad (\text{A.1})$$

$$z_k = H_k x_k + v_k, \quad (\text{A.2})$$

where x_k is an $n \times 1$ system state vector, A_k is an $n \times n$ transition matrix, u_k is an $l \times 1$ vector of the input forcing function, B_k is an $n \times l$ matrix, w_k is an $n \times 1$ process noise vector, z_k is an $m \times 1$ measurement vector, H_k is an $m \times n$ measurement matrix and v_k is an $m \times 1$ measurement noise vector. Both the w_k and v_k are assumed to be uncorrelated zero-mean Gaussian white noise sequences with covariance given by

$$E[w_k w_i^T] = \begin{cases} Q_k, & i = k, \\ 0, & i \neq k, \end{cases} \quad (\text{A.3})$$

$$E[v_k v_i^T] = \begin{cases} R_k, & i = k, \\ 0, & i \neq k, \end{cases} \quad (\text{A.4})$$

$$E[w_k v_i^T] = 0, \text{ for all } k \text{ and } i. \quad (\text{A.5})$$

The SKF algorithm can be organised into time update and measurement update equations

Time update equations:

$$\hat{x}_{k+1}^- = A_k \hat{x}_k + B_k u_k, \quad (\text{A.6})$$

$$P_{k+1}^- = A_k P_k A_k^T + Q_k. \quad (\text{A.7})$$

Measurement update equations:

$$K_k = P_k^- H_k^T [H_k P_k^- H_k^T + R_k]^{-1}, \quad (\text{A.8})$$

$$\hat{x}_k = \hat{x}_k^- + K_k [z_k - H_k \hat{x}_k^-], \quad (\text{A.9})$$

$$P_k = [I - K_k H_k] P_k^-. \quad (\text{A.10})$$

The measurement update equations incorporate a new observation into the a priori estimate from the time update equations to obtain an improved a posteriori estimate. In the time and measurement update equations, \hat{x}_k is an estimate of the system state vector x_k , K_k is the Kalman gain and P_k is the covariance matrix of the state estimation error.

References

- Asakawa, K., Kojima, J., Kato, Y., Matsumoto, S., & Kato, N. (2000). Autonomous underwater vehicle aqua explorer 2 for inspection of underwater cables. *Proceedings of the 2000 international symposium on underwater technology*, Tokyo, Japan, pp. 242-247.
- Chao, W. S., Bishop, R. H., & Gough, J. A. (1997). Mixture-of-experts framework for adaptive Kalman filtering. *IEEE Transactions on Systems, Man and Cybernetics—Part B: Cybernetics*, 27(3), 452-464.
- Escamilla-Ambrosio, P. J., & Mort, N. (2001). A hybrid Kalman filter-fuzzy logic multisensor data fusion architecture with fault tolerant characteristics. *Proceedings of the 2001 international conference on artificial intelligence*, Las Vegas, NV, USA, pp. 361-367.
- Fitzgerald, R. J. (1971). Divergence of the Kalman filter. *IEEE Transactions on Automatic Control*, AC-16(6), 736-747.
- Gade, K., & Jølving, B. (1999). An aided navigation post processing filter for detailed seabed mapping UUVs. *Modeling, Identification and Control*, 20(3), 165-176.
- Grenon, G., An, P. E., Smith, S. M., & Healey, A. J. (2001). Enhancement of the inertial navigation system for the morpheous autonomous underwater vehicles. *IEEE Journal of Oceanic Engineering*, 26(4), 548-560.
- Hanlon, P. D., & Maybeck, P. S. (2000). Multiple-model adaptive estimation using a residual correlation Kalman filter bank. *IEEE Transactions on Aerospace and Electronic Systems*, 36(2), 393-406.
- Jetto, L., Longhi, S., & Vitali, D. (1999). Localization of a wheeled mobile robot by sensor data fusion based on a fuzzy logic adapted Kalman filter. *Control Engineering Practice*, 7, 763-771.
- Kailath, T. (1968a). An innovations approach to least-squares estimation, Part I: Linear filtering in additive noise. *IEEE Transactions on Automatic Control*, AC-13(6), 646-655.
- Kailath, T. (1968b). An innovations approach to least-squares estimation, Part II: Linear smoothing in additive white noise. *IEEE Transactions on Automatic Control*, AC-13(6), 655-660.
- Kailath, T. (1970). The innovations approach to detection and estimation theory. *IEEE Proceedings*, 58(5), 680-695.
- Kinsey, J. C., & Whitcomb, L. L. (2003). Preliminary field experience with the DVLNAV integrated navigation system for manned and unmanned submersibles. *Proceedings of the first IFAC workshop on guidance and control of underwater vehicles*, Newport, South Wales, UK, pp. 83-88.
- Kobayashi, K., Cheok, K. C., Watanabe, K., & Muneka, F. (1998). Accurate differential global positioning system via fuzzy logic Kalman filter sensor fusion technique. *IEEE Transactions on Industrial Electronics*, 45(3), 510-518.

- Loebis, D., Chudley, J., & Sutton, R. (2003a). A fuzzy Kalman filter optimized using a genetic algorithm for accurate navigation of an autonomous underwater vehicle. *Proceedings of the sixth IFAC conference on manoeuvring and control of marine craft*, Girona, Spain, pp. 19–24.
- Loebis, D., Sutton, R., & Chudley, J. (2002). Review of multisensor data fusion techniques and their application to autonomous underwater vehicle navigation. *Journal of Marine Engineering and Technology*, 11, 3–14.
- Loebis, D., Sutton, R., & Chudley, J. (2003b). A fuzzy Kalman filter for accurate navigation of an autonomous underwater vehicle. *Proceedings of the first IFAC workshop on guidance and control of underwater vehicles*, Newport, South Wales, UK, pp. 161–166.
- Magill, D. T. (1965). Optimal adaptive estimation of sampled stochastic processes. *IEEE Transaction on Automatic Control*, AC-10(4), 434–439.
- Mehra, R. K. (1970). On the identification of variances and adaptive Kalman filtering. *IEEE Transactions on Automatic Control*, AC-15(2), 175–184.
- Mehra, R. K. (1971). On-line identification of linear dynamic systems with applications to Kalman filtering. *IEEE Transaction on Automatic Control*, AC-16(1), 12–21.
- Mohamed, A. H., & Schwarz, K. P. (1999). Adaptive Kalman filtering for INS/GPS. *Journal of Geodesy*, 73, 193–203.
- Størkersen, N., Kristensen, J., Indreide, A., Seim, J., & Glancy, T. (1998). Hugin—UUV for seabed surveying. *Sea Technology*, 39(2), 99–104.
- Wright, J., Scott, K., Tien-Hsin, C., Lau, B., Lathrop, J., & McCormick, J. (1996). Multi-sensor data fusion for seafloor mapping and ordnance location. *Proceedings of the 1996 symposium on autonomous underwater vehicle technology*, Monterey, CA, USA, pp. 167–175.
- Yun, X., Bachmann, R. E., McGhee, R. B., Whalen, R. H., Roberts, R. L., Knapp, R. G., Healey, A. J., & Zyda, M. J. (1999). Testing and evaluation of an integrated GPS/INS system for small AUV navigation. *IEEE Journal of Oceanic Engineering*, 24(3), 396–404.

PROCEEDINGS OF THE
**Institution of
Mechanical
Engineers**

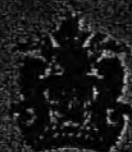


JOURNAL OF
**Engineering for the
Maritime Environment**

PROCEEDINGS PART M



February 2004 Vol 218 No M1
ISSN 1475-0902



IMarEST

Published in association with
RINA and IMarEST

A fuzzy Kalman filter optimized using a multi-objective genetic algorithm for enhanced autonomous underwater vehicle navigation

D Loebis*, R Sutton and J Chudley

Marine and Industrial Dynamic Analysis Research Group, School of Engineering, University of Plymouth, UK

Abstract: In an autonomous underwater vehicle integrated navigation system, short-term temporal accuracy is provided by an inertial navigation subsystem (INS) and long-term accuracy by a global positioning system (GPS). The Kalman filter has been a popular method for integrating the data produced by the two systems to provide optimal estimates of autonomous underwater vehicle position and attitude. In this paper, a sequential use of a linear Kalman filter and extended Kalman filter is proposed. The former is used to fuse the data from a variety of INS sensors whose output is used as an input to the latter where integration with GPS data takes place. The use of a fuzzy-rule-based adaptation scheme to cope with the divergence problem caused by the insufficiently known *a priori* filter statistics is also explored. The choice of fuzzy membership functions for an adaptation scheme is first carried out using a heuristic approach. Multiobjective genetic algorithm techniques are then used to optimize the parameters of the membership functions with respect to a certain performance criteria in order to improve the overall accuracy of the integrated navigation system. Simulation results are presented that show that the proposed algorithms can provide a significant improvement in the overall navigation performance of an autonomous underwater vehicle navigation system.

Keywords: autonomous underwater vehicles, navigation, sensor fusion, Kalman filters, extended Kalman filters, fuzzy logic, multiobjective genetic algorithm

NOTATION

$A(t)$	linearized state transition matrix of the kinematic model
A_k	discretized state transition matrix of the kinematic model
B_k	matrix that relates the control input to the state vector
c_k	weight of the estimated sensor output
$C(t)$	linearized measurement matrix of the kinematic model
C_k	discretized measurement matrix of the kinematic model
\hat{C}_{Inn_k}	actual covariance value of the innovation
delta_k	discrepancy between the actual and theoretical covariance value of innovation sequence
f	scalar function of design parameters
F	vector of the objective function of design parameters

$F(\cdot)$	state transition matrix of the kinematic model
$H(\cdot)$	measurement matrix of the kinematic model
H_k	measurement matrix of the dynamic model
i	number of fuzzy membership functions
Inn_k	innovation
J_0	first sample inside the estimation window
J_{ze}	performance measure of the fused sensor output
J_{zv}	performance measure of the non-fused sensor output
k	discrete-time index
K_k	Kalman gain
l	number of sensors or the FKF output
L_i	output fuzzy membership functions
M	size of the moving estimation window
n	number of samples
O_i	input fuzzy membership functions
p	design parameters of the optimization problem
P_k	state error covariance
Q_k	process noise covariance
$r(t)$	yaw rate of the turn state of the dynamic model

The MS was received on 13 August 2003 and was accepted after revision for publication on 25 February 2004.

* Corresponding author: Marine and Industrial Dynamic Analysis Research Group, School of Engineering, University of Plymouth, Drake Circus, Plymouth PL4 8AA, UK.

R_k	measurement error covariance
S_k	theoretical covariance value of innovation
t	continuous time
$v(t)$	sway velocity state of the dynamic model
$w(t)$	measurement white noise of the dynamic model
$V(t)$	measurement white noise of the kinematic model
$w(t)$	process white noise of the dynamic model
$W(t)$	process white noise of the kinematic model
x_k	states of the dynamic model
$X(t)$	states of the kinematic model
z_k	real-time measurement
z_{a_k}	actual value measurement
$Z(t)$	measurement vector of the kinematic model
$\delta_r(t)$	rudder deflection
$\dot{\theta}(t)$	yaw rate of the turn of the kinematic model
κ	output variable of fuzzy sets
$\lambda(t)$	longitude
$v(t)$	sway velocity state of the kinematic model
$\zeta(t)$	surge velocity state of the kinematic model
$\varphi(t)$	latitude
χ	input variable of fuzzy sets
$\psi(t)$	yaw angle state of the dynamic model
$\Psi(t)$	yaw angle state of the kinematic model
Ω	set of constraints on design parameters of the optimization problem

1 INTRODUCTION

In the past few decades, there have been numerous worldwide research and development activities in order to explore the oceans of the world. As an ocean is an inherently hostile and hazardous environment, the need for an underwater robotic system, especially one with high reliability and fully built-in intelligence, becomes apparent. The autonomous underwater vehicle (AUV) class of vessel meets these requirements.

To achieve truly autonomous behaviour, an AUV must be able to locate itself accurately during an operating scenario using only its onboard sensors. In the past, fusing of inertial navigation system (INS) sensors and the integration with a global positioning system (GPS) through the use of a conventional linear Kalman filter (LKF) and an extended Kalman filter (EKF) has been a popular method for localization of an AUV [1]. However, a significant difficulty in designing a KF (refers to both LKF and EKF) can often be traced to incomplete *a priori* knowledge of the process covariance matrix (Q) and measurement noise covariance matrix (R). In most practical applications, these matrices are initially estimated or even unknown. The problem here is that the optimality of the estimation algorithm in the KF setting is closely connected to the quality of *a priori* information about the process and measurement noise [2, 3]. It has been

shown that insufficiently known *a priori* filter statistics can reduce the precision of the estimated filter states or introduces biases to their estimates. In addition, incorrect *a priori* information can lead to practical divergence of the filter [4]. From the aforementioned it may be argued that the conventional KF with fixed R and/or Q should be replaced by an adaptive estimation formulation, as discussed in the next section.

2 THE ADAPTIVE KALMAN FILTER ALGORITHM

In the past few years, only a few publications in the area of adaptive Kalman filtering can be found in the literature. The two major approaches that have been proposed for adaptive Kalman filtering are multiple model adaptive estimation (MMAE) and innovation adaptive estimation (IAE). Although the implementation of these approaches are quite different, they both share the same concept of utilizing new statistical information obtained from the innovation (or residual) sequence. In both cases, the innovation inn_k at sample time k is the difference between the real-time measurement z_k received by the filter and its estimated (predicted) value \hat{z}_k . The predicted measurement is the projection of the filter predicted states \hat{x}_k^- on to the measurement space through the measurement design matrix H_k . Innovation represents additional information available to the filter as a result of the new measurement z_k . The occurrence of data with statistics different from the *a priori* information will first show up in the innovation vector. For this reason the innovation sequence represents the information content in the new observation and is considered the most relevant source of information to the filter adaptation. Interested readers can refer to references [5] to [7] for a more detailed discussion of the innovation sequence and its use in linear filter theory.

In the MMAE approach, a bank of Kalman filters runs in parallel [8, 9] or with a gating algorithm [10] under a different model for the statistical filter information matrices, i.e. Q and R . In the IAE approach [2, 3], the Q and R themselves are adapted as measurements evolve with time. In this paper, the IAE approach [2, 3] coupled with fuzzy logic techniques is used to adjust the R matrix of the KF.

The fuzzy logic membership functions for the IAE approach are initially established by a combination of knowledge, experience and observation and may thus not be optimal. Additionally, fine-tuning of its performance is still a matter of trial and error. Many studies have shown that genetic algorithms (GAs) have the ability to find fuzzy membership functions closer to optimal solutions and may be made to implement self-tuning and adaptive schemes [11]. However, this paper is the first known use of the multiobjective genetic algorithm (MOGA) for the optimization of the membership

function of a fuzzy system in the noise adaptation of a KF and is thus considered as the major contribution of this particular study in relation to AUV technology.

2.1 Fuzzy Kalman filter

In this subsection, an on-line innovation-based adaptive scheme of the KF to adjust the R matrix employing the principles of fuzzy logic is presented. The fuzzy logic is chosen mainly because of its simplicity. This motivates interest in the topic, as testified by related articles that have been appearing in the literature [12–14].

The fuzzy logic Kalman filter (FKF) proposed herein is based on the IAE approach using a technique known as covariance matching [2]. The basic idea behind the technique is to make the actual value of the covariance of the innovation sequences match its theoretical value.

The actual covariance is defined as an approximation of the Inn_k sample covariance through averaging inside a moving estimation window of size M [15], which takes the following form:

$$\hat{C}_{Inn_k} = \frac{1}{M} \sum_{j=j_0}^k Inn_k Inn_k^T \quad (1)$$

where $j_0 = k - M + 1$ is the first sample inside the estimation window. An empirical experiment is conducted to choose the window size M . From experimentation it was found that a good size for the moving window in equation (1) for the Inn_k used in this paper is 15. The value of M is dependent on the dynamic of the Inn_k and therefore can be different for different types of applications.

The theoretical covariance of the innovation sequence is defined as

$$S_k = H_k P_k^- H_k^T + R_k \quad (2)$$

The logic of the adaptation algorithm using the covariance matching technique can be qualitatively described as follows. If the actual covariance value \hat{C}_{Inn_k} is observed, whose value is within the range predicted by theory S_k and the difference is very near to zero, this indicates that both covariances match almost perfectly and only a small

change is needed to be made on the value of R . If the actual covariance is greater than its theoretical value, the value of R should be decreased. On the contrary, if \hat{C}_{Inn_k} is less than S_k , the value of R should be increased. This adjustment mechanism lends itself very well to being dealt with using a fuzzy logic approach based on rules of the kind:

$$\text{IF } \langle \text{antecedent} \rangle \text{ THEN } \langle \text{consequent} \rangle \quad (3)$$

where antecedent and consequent are of the form $\chi \in O_i$, $\kappa \in L_i$, $i = 1, 2, \dots$, respectively, where χ and κ are the input and output variables respectively and O_i and L_i are the fuzzy sets.

To implement the above covariance matching technique using the fuzzy logic approach, a new variable called δ_{Inn_k} is defined to detect the discrepancy between \hat{C}_{Inn_k} and S_k . The following three fuzzy rules of the kind (3) are used:

$$\text{IF } \langle \delta_{Inn_k} \cong 0 \rangle \text{ THEN } \langle R_k \text{ is unchanged} \rangle \quad (4)$$

$$\text{IF } \langle \delta_{Inn_k} > 0 \rangle \text{ THEN } \langle R_k \text{ is decreased} \rangle \quad (5)$$

$$\text{IF } \langle \delta_{Inn_k} < 0 \rangle \text{ THEN } \langle R_k \text{ is increased} \rangle \quad (6)$$

Thus R is adjusted according to

$$R_k = R_{k-1} + \Delta R_k \quad (7)$$

where ΔR_k is added or subtracted from R at each instant of time. Here δ_{Inn_k} is the input to the fuzzy inference system (FIS) and ΔR_k is the output.

On the basis of the above adaptation hypothesis, the FIS can be implemented using three fuzzy sets for δ_{Inn_k} : $N = \text{Negative}$, $Z = \text{Zero}$ and $P = \text{Positive}$. For ΔR_k the fuzzy sets are specified as $I = \text{Increase}$, $M = \text{Maintain}$ and $D = \text{Decrease}$. The membership functions of these fuzzy sets which are first designed using a heuristic approach are shown in Fig. 1.

2.2 Fuzzy logic observer

To monitor the performance of an FKF, another FIS called the fuzzy logic observer (FLO) [16] is used. The FLO assigns a weight or degree of confidence denoted

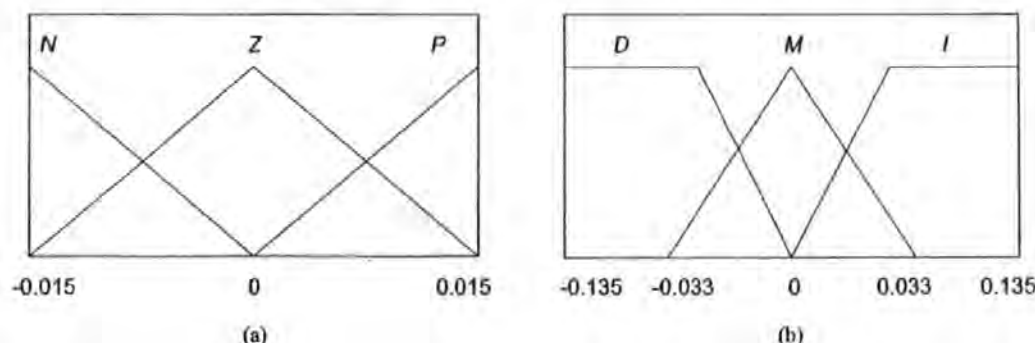


Fig. 1 Membership function of (a) δ_{Inn_k} and (b) ΔR_k

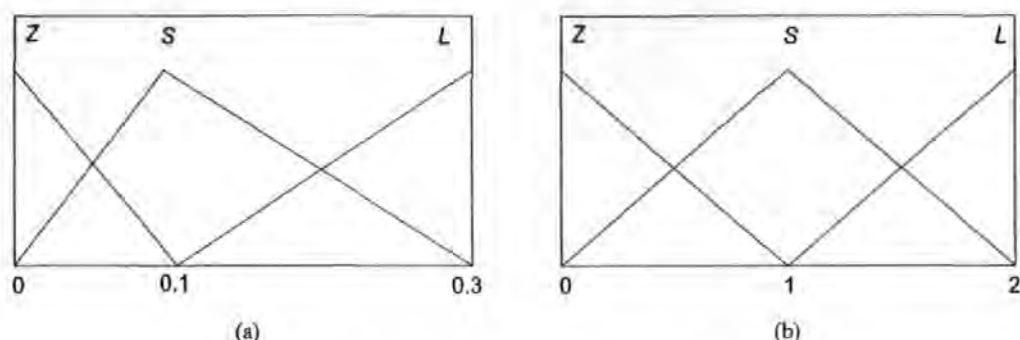


Fig. 2 Membership function of (a) $|\delta_k|$ and (b) R_k

as c_k , a number on the interval $[0, 1]$, to the FKF state estimate. The FLO is implemented using two inputs: the values of $|\delta_k|$ and R_k . The membership functions of these variables are shown in Fig. 2.

The fuzzy labels for the membership functions: $Z = \text{Zero}$, $S = \text{Small}$ and $L = \text{Large}$. Three fuzzy singletons are defined for the output c_k and are labelled as $G = \text{Good}$, $AV = \text{Average}$ and $P = \text{Poor}$, with values 1, 0.5 and 0 respectively. The basic heuristic hypothesis for the FLO is as follows: if the value of $|\delta_k|$ is near to zero and the value of R_k is near to zero, then the FKF works almost perfectly and the state estimate of the FKF is assigned a weight near 1. On the contrary, if one or both of these values increases far from zero, it means that the FKF performance is degrading and the FLO assigns a weight near 0. Table 1 gives the complete fuzzy rule base of each FLO.

2.3 Fuzzy membership functions optimization

GAs as function optimizers are global optimization techniques based on natural selection [17]. A simple, or single, GA is restricted to tackling optimization problems of the form

$$\min f(p), \quad p \in \Omega \quad (8)$$

where $p = [p_1, p_2, \dots, p_q]$ represents the design parameters of the problem and Ω represents a set of constraints on those parameters. The objective function, f , to be minimized is a scalar function of the design parameters. Most practical engineering problems, however,

require more than one objective function to be optimized simultaneously. The general form of this type of problem is

$$\min F(p), \quad p \in \Omega, \quad (9)$$

where p and Ω are defined in the same way as before and $F = [f_1, f_2, \dots, f_n]$ is a vector of objective functions to be minimized. In this paper, functions of this type are referred to as multiobjective functions.

The MOGA is presented here as a tool to optimize simultaneously the FKF membership functions with respect to, possibly competing, multiobjective functions. The result of such an optimization is a number of points on the surface of maximum attainment in the dimension of the problem, known as a Pareto-optimal surface of non-dominated solutions [18]. These are defined such that a non-dominated solution is one for which an improvement in one objective will lead to a degradation in one or more objectives. Here, a population-based optimizer can generate a number of points on this surface, giving the designer the ability to trade objectives against each other.

Figure 3 shows a schematic of the MOGA. The algorithm starts by evaluating the objectives proposed by an initial random population. The user can interact with the optimization process by altering the goal and priority information as the optimization progresses, which enable the user to effectively steer the optimization towards a region of the objective space of particular interest. Pareto-optimal ranking assigns all individuals in a current population a rank equal to the degree of non-dominance in the corresponding trade-off region. This kind of ranking is non-unique; e.g. non-dominated individuals are ranked 0, whereas the solution ranked 5 is dominated by five other solutions in a multiobjective sense. Due to the stochastic nature of a MOGA, individuals with the same fitness may produce a different number of offspring. The resulting accumulation of the imbalances in reproduction may lead the search into a random area of the trade-off surface. This phenomenon is known as genetic drift and can drastically reduce the quality and efficiency of the search. Here, the crossover and mutation

Table 1 Fuzzy rule base FLO

$ \delta_k $	R_k		
	Z	S	L
Z	G	G	AV
S	G	AV	P
L	AV	P	P

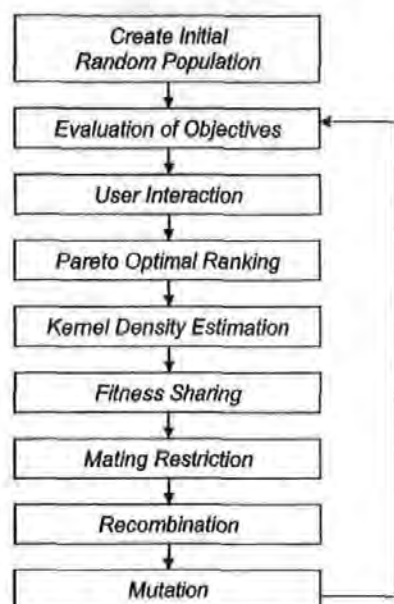


Fig. 3 Schematic of the MOGA

can be less likely to produce different individuals and may result in restriction of the coverage search space. Fitness sharing helps reduce genetic drift by penalizing the fitness of individuals in popular neighbourhoods in favour of more remote individuals of similar fitness [19]. The essence of fitness sharing operator is the kernel density estimator, which statistically estimates the population density at each point. Finally, mating restriction [20] limits the recombination of the unacceptably large number of unfit individuals, called *lethals*. It is achieved

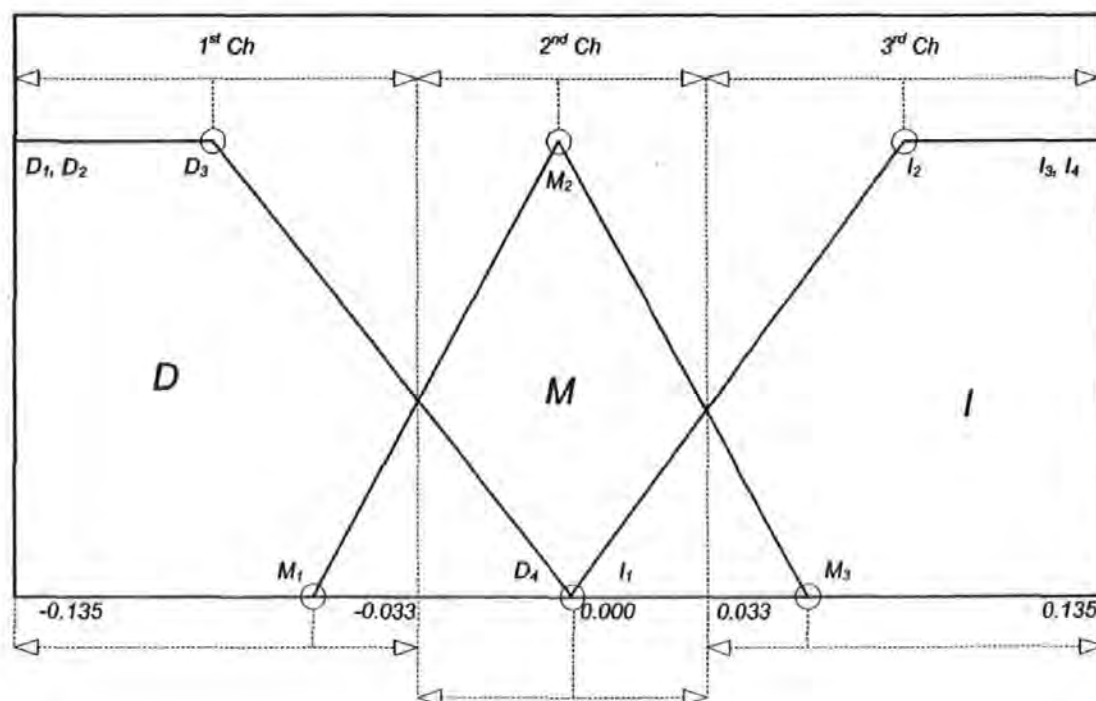
by restricting reproduction to individuals that are within a given distance of each other.

To translate the FKF membership functions to a representation useful as genetic material, they are parameterized with real-valued variables. Each of these variables constitutes a gene of the chromosomes for the MOGA. Boundaries of chromosomes are required for the creation of chromosomes in the right limits so that the MOGA is not misled to some other area of search space. The technique adapted in this paper is to define the boundaries of the output membership functions according to the furthest points and the crossover points of two adjacent membership functions. In other words, the boundaries of the FKF consist of three real-valued chromosomes (*Chs*), as shown in Fig. 4.

The trapezoidal membership functions' two furthest points, -0.135 (D_1), -0.135 (D_2) and 0.135 (I_3), 0.135 (I_4) of the FKF, remain the same in the GA's description to allow a similar representation as the fuzzy system's definition. As can be seen from Fig. 4, D_3 and M_1 can change value in the first *Ch* boundary, D_4 , M_2 and I_1 in the second *Ch* boundary and, finally, M_3 and I_2 in the third *Ch* boundary. Table 2 shows the encoding used for optimization of the membership functions.

Table 2 FKF boundaries

Limit	Parameter		
	D_3, M_1	D_4, M_2, I_1	M_3, I_2
Upper limit	-0.135	-0.033	0.033
Lower limit	-0.033	0.033	0.135

Fig. 4 Membership function and boundaries of R_k

3 FUSION OF INS SENSOR DATA

In this section, the FKF technique is applied to maintain the optimality of an AUV heading estimation process. The FLO will then be used subsequently to fuse the estimated heading values. To this end, an AUV linearized heading dynamic model derived by using sway and yaw equations of motion along with Euler angle mapping [21] is employed. It is assumed in this model that the forward velocity of the vehicle is constant at 1.3 m/s and the vehicle is not at an angle of roll and pitch. The components of the equations were obtained using standard fluid techniques such as Kirchhoff's equations [22], which are usually simplified by neglecting second-order terms, either explicitly or implicitly, from Taylor series expansions. The state-space form (please also refer to the Appendix) of the dynamic model is

$$\begin{bmatrix} \dot{v}(t) \\ \dot{r}(t) \\ \dot{\psi}(t) \end{bmatrix} = \begin{bmatrix} -2.09 & 0.376 & 0 \\ 7.96 & -8.69 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v(t) \\ r(t) \\ \psi(t) \end{bmatrix} + \begin{bmatrix} 1.07 \\ -14.1 \\ 0 \end{bmatrix} \delta_r(t) + w(t) \quad (10)$$

where $v(t)$, $r(t)$ and $\psi(t)$ represent the sway velocity, yaw rate of turn and yaw angle. The H (see the Appendix) in this case is $[0 \ 0 \ 1]^T$ (only the third state is observed by its corresponding sensor). The $w(t)$ and $v(t)$ are both zero mean white noise for the system and measurement models respectively and $\delta_r(t)$ is the rudder deflection. A sample time of 0.125 s is used to discretize the linearized model. The initial conditions are $[v_0 \ r_0 \ \psi_0]^T = [0 \text{ m/s} \ 0 \text{ rad/s} \ 0 \text{ rad}]^T$, $P_0 = \text{diag}[0.01 \text{ m/s}^2 \ 0.01 \text{ rad/s}^2 \ 0.01 \text{ rad}^2]$ and Q_k is made constant as $\text{diag}[0.01 \text{ m/s}^2 \ 0.01 \text{ rad/s}^2 \ 0.01 \text{ rad}^2]$. The values of P_0 and Q_k are determined heuristically. In real-time applications, the Q_k values are dependent on temporal and spatial variations in the environment such as sea conditions, ocean current, local magnetic variations, etc., and therefore appropriate adjustments to the initial values of Q also need to be undertaken. This topic will be addressed in forthcoming papers. In equation (10) a sinusoidal input was applied to the rudder. Four yaw sensors with different noise characteristics are considered to measure the response of the vehicle.

The actual value of R for each yaw sensor is assumed unknown but its initial value is selected as 0.01 rad^2 . The FKF algorithm optimized using MOGA with parameters shown in Table 3 was then implemented with the trade-off graph shown in Fig. 5 and simulation results shown in the next section.

Table 3 MOGA parameters

Parameters	Values
Number of objective functions	5
Number of generations	200
Number of individuals per generation	25
Generation gap in selection operation	0.95
Rate in recombination operation	0.8
Rate in mutation operation	0.09

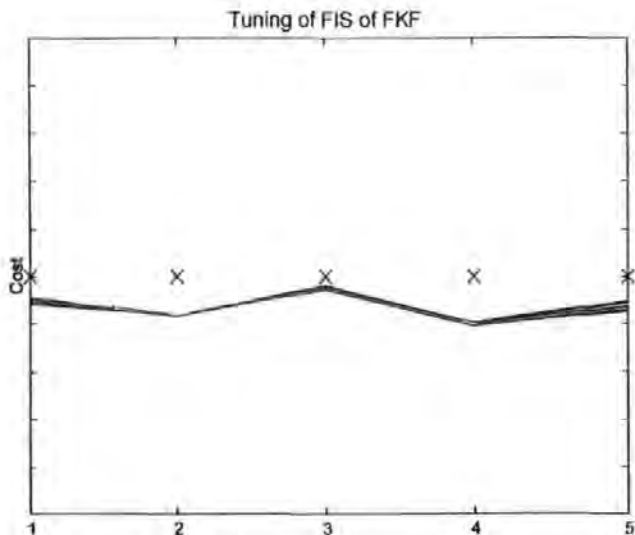


Fig. 5 Trade-off graph for a FKF search

3.1 Simulation results

Figures 6 and 7 are the simulation results showing the response of the AUV observed by sensors with constant Gaussian noise, while Figs 8 and 9 are the results observed by sensors with uniform noise increasing and decreasing with time respectively. These types of noise are included in the simulation to demonstrate the effectiveness of the proposed adaptation mechanism in general. A possible real-time scenario that can result in the noise with the characteristic shown in Fig. 8 is that of the third yaw sensor located in close proximity to an electronic hardware such as a propeller d.c. motor whose internal temperature increases with time and affects the sensor ambient temperature. A similar scenario can also occur when the fourth yaw sensor is located in close proximity to another electronic hardware such as a hydro-plane stepper motor whose initial internal temperature is high and settles down after some time. Figure 10 shows the values of R after the FKF has been run. Figures 6 and 9 also show several peaks in the simulations of sensors 1 and 4. These are to indicate faults in the sensor. There are two types of fault defined in this simulation work, transient and persistent faults [16]. Transient fault

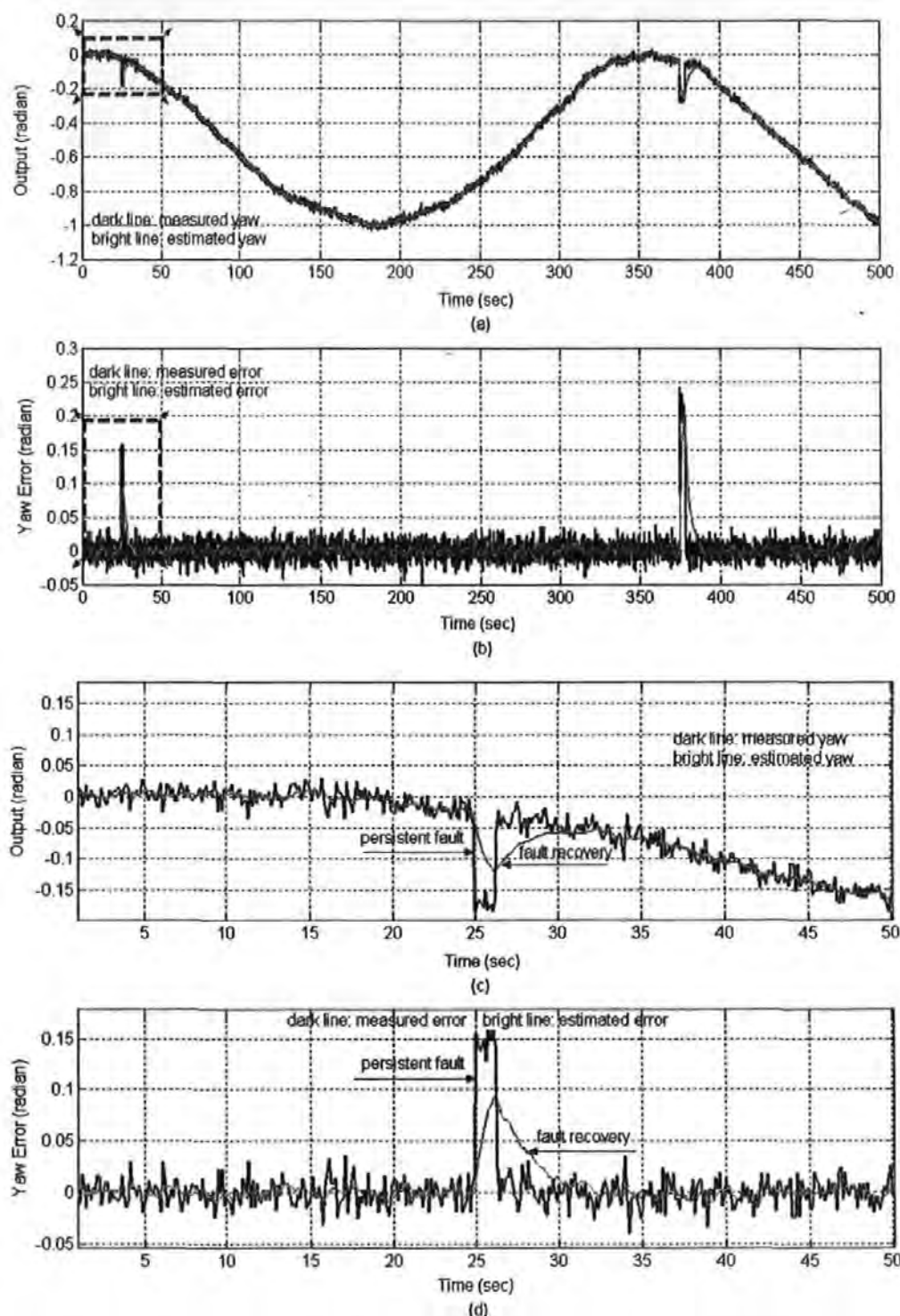


Fig. 6 (a) Measured and estimated yaw output; (b) measured and estimated yaw error of sensor 1; (c) measured and recovered yaw output; (d) measured and recovered yaw error of sensor 1

happens when the sensor output increases abruptly for only a sample period of time. Persistent faults occur when the transient faults persist for an instant of time. Consequently, the peaks in sensor 1 and 4 simulations show the persistent and transient faults respectively. Figures 6c, d and 9c, d provide a closer look on the

indicated areas in Figs 6a, b and 9a, b respectively. It is clear in both cases that the algorithms have detected faults in the system and appropriate actions have been undertaken to recover the signals. Direct observation of Figs 6 to 9 shows how the proposed method has significantly reduced the level of error in the system.

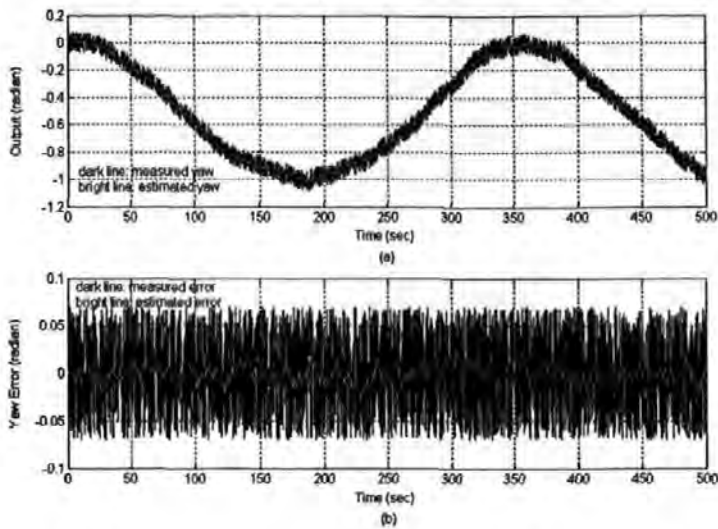


Fig. 7 (a) Measured and estimated yaw output; (b) measured and estimated yaw error of sensor 2

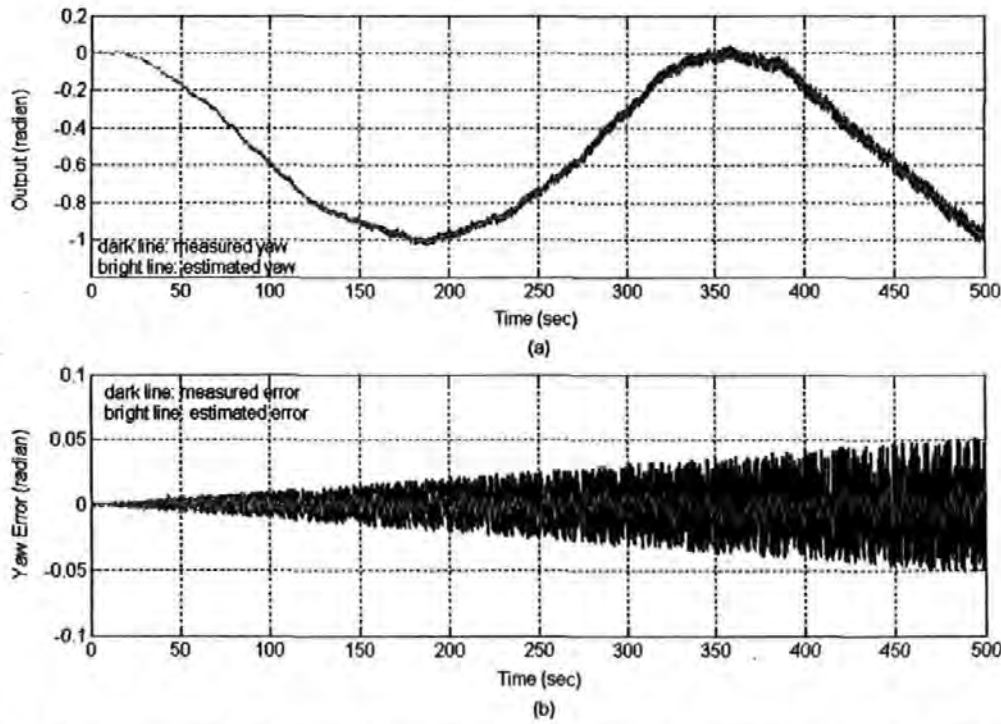


Fig. 8 (a) Measured and estimated yaw output; (b) measured and estimated yaw error of sensor 3

To fuse the estimated yaw, a centre of gravity method is used (please note that as $H=[0 \ 0 \ 1]^T$ the vector z now takes the form scalar z)

$$z_k = \frac{\sum_{i=1}^n \hat{z}_{k_i} c_{k_i}}{\sum_{i=1}^n c_{k_i}} \tag{11}$$

where \hat{z}_{k_i} is the output of the i th FKF ($i=1, 2, 3, 4$) and c_{k_i} is the respective weight at instant time k .

Finally, the following performance measures are adopted for comparison purposes:

$$J_{zy} = \sqrt{\frac{1}{n} \sum_{k=1}^n (za_k - z_k)^2} \tag{12}$$

$$J_{ze} = \sqrt{\frac{1}{n} \sum_{k=1}^n (za_k - \hat{z}_k)^2} \tag{13}$$

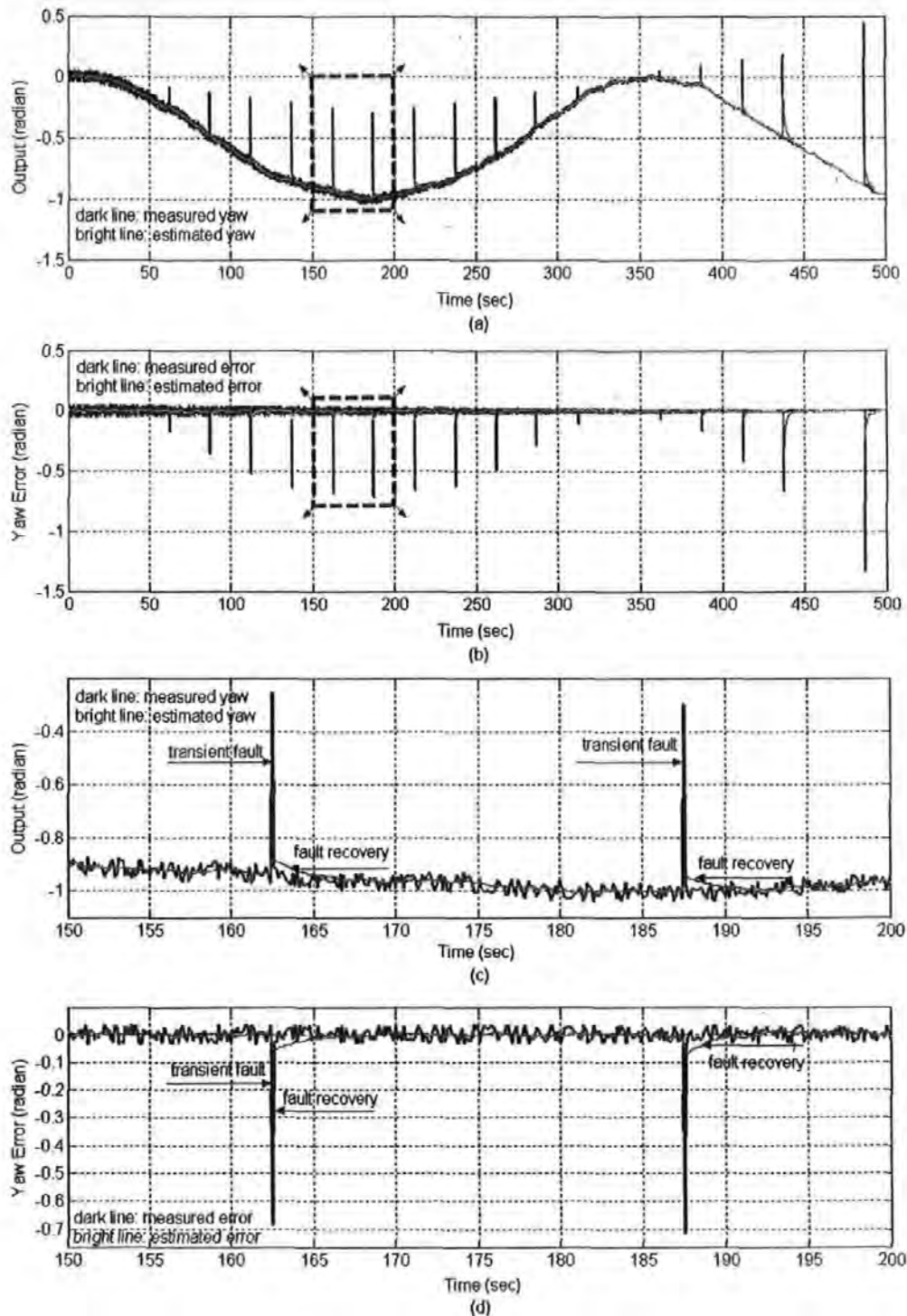


Fig. 9 (a) Measured and estimated yaw output; (b) measured and estimated yaw error of sensor 4; (c) measured and recovered yaw output; (d) measured and recovered yaw error of sensor 4

where z_{a_k} is the actual value of the yaw, z_k is the measured yaw, \hat{z}_k is the estimated yaw at an instant of time k and $n = \text{number of samples}$.

Table 4 shows the comparison of performance of each individual measured sensor output with the one obtained using standard FKF and FKF optimized using MOGA (FKF-MOGA) respectively. A further comparison is

also made between the performance of the fused sensor outputs obtained using both FKF schemes. A first look at the table shows that J_{ze} of each individual sensor always outperforms J_{zv} . Most importantly is that every single J_{ze} of the fused sensor output from both FKF schemes outperforms its individual sensor counterpart. It is also clear that the MOGA optimization techniques

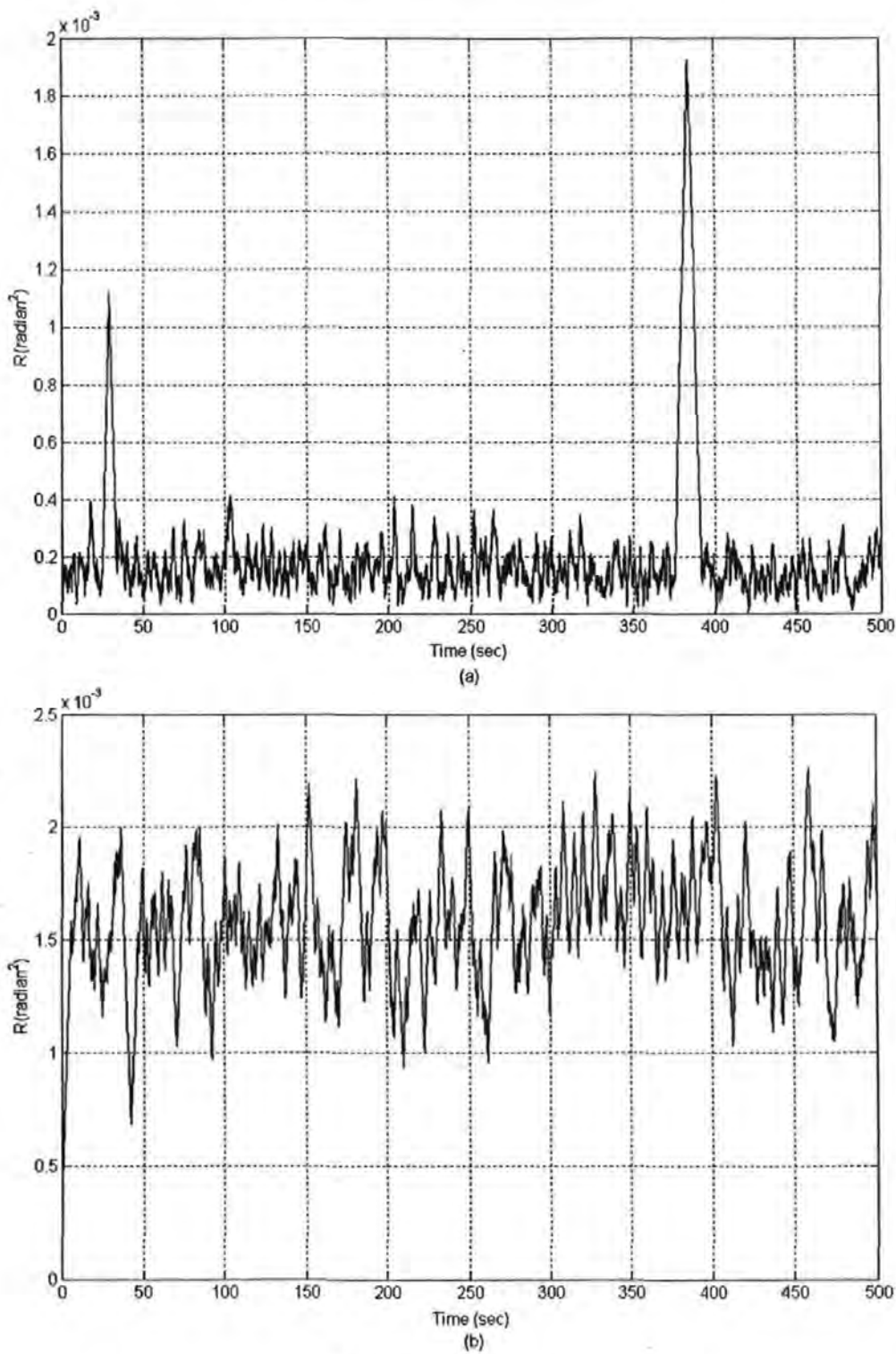


Fig. 10 (continued over)

have significantly increased the performance of sensor 1, as indicated by the corresponding J_{ze} values. The FKF-MOGA J_{ze} values of sensor 4 also show a similar trend of improvement, although not as good as the ones produced by sensor 1. The J_{ze} values of sensors 2 and 3 in the FKF-MOGA case are slightly inferior compared to

those in the standard FKF case. A logical explanation to this is that the MOGA mutation process might not have produced a sufficiently 'mutated' individual in the specified number of generations. Given the complexity and the computational burden of the proposed algorithms, it is decided that running the MOGA for many more

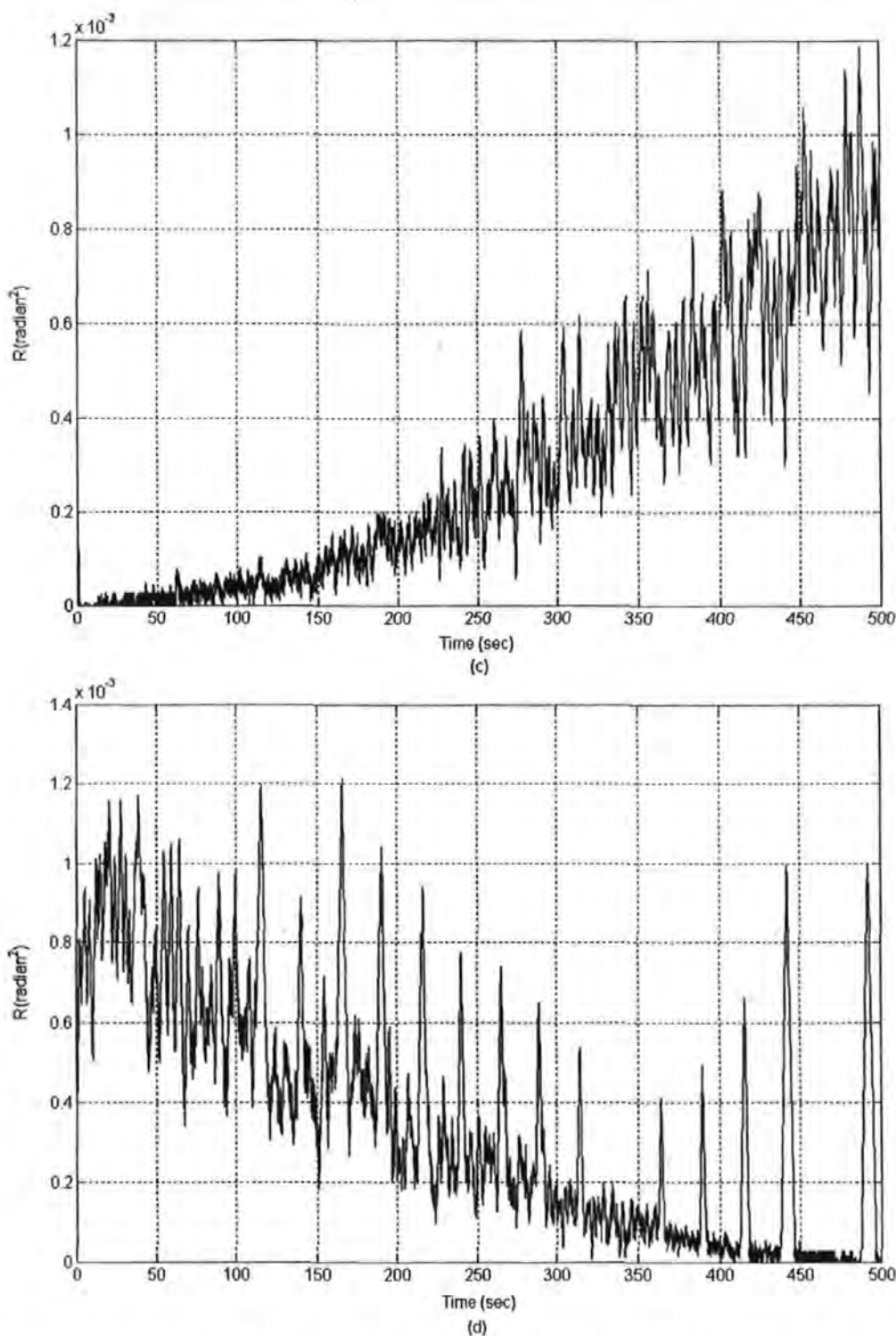


Fig. 10 (a) The R_k after FKF adaptation for sensor 1; (b) the R_k after FKF adaptation for sensor 2; (c) the R_k after FKF adaptation for sensor 3; (d) the R_k after FKF adaptation for sensor 4

numbers of generations will be the topic of forthcoming papers. However, it should be noted that any improvement in J_{ze} values of either sensor 2 or 3 can degrade the J_{ze} values of other sensors. This stems from the non-dominated nature of MOGA solutions for which an improvement in one objective will lead to a degradation in one or more other objectives, as discussed in section 2.3.

4 GPS/INS NAVIGATION

Here, the fused estimated yaw obtained previously is treated as a single imaginary yaw sensor and used by other INS sensors to transform data from a body coordinate frame to a geographical (north-east-down, or NED) coordinate frame where integration with

Table 4 Comparison of performance

Sensor	Performance (rad)				
	J_{2v}	Non-MOGA	J_{2e}		
			MOGA		
			First	Second	Third
Sensor 1	0.026 687 829 592 25	0.019 097 392 675 32	0.004 686 209 041 60	0.004 658 283 773 57	0.004 700 171 675 62
Sensor 2	0.039 982 002 504 69	0.008 758 062 186 51	0.008 773 770 149 78	0.008 765 043 503 52	0.008 759 807 515 76
Sensor 3	0.017 336 355 460 06	0.004 944 517 770 90	0.004 979 424 355 94	0.005 010 840 282 48	0.005 003 858 965 47
Sensor 4	0.020 936 969 706 92	0.005 070 181 477 04	0.004 879 940 588 58	0.004 872 959 271 57	0.004 869 468 613 06
Fused		0.003 499 385 150 25	0.003 722 787 294 50	0.003 729 768 611 51	0.003 728 023 282 26

converted GPS data is performed using a combination of FKF and EKF techniques, which can be referred to as a fuzzy extended Kalman filter (FEKF). Figure 11 shows this relationship and serves as an overall representation of the algorithms that have been discussed so far.

A continuous-time model of the vehicle motion appropriate to this problem is taken to be

$\dot{X}(t) = F(X(t)) + W(t)$ (14)

$Z(t) = H(X(t)) + V(t)$ (15)

where $X(t) = [\lambda(t) \varphi(t) \Psi(t) \vartheta(t) \zeta(t) v(t)]^T$ show the model states, $\lambda(t)$ and $\varphi(t)$ are the longitude and latitude of the AUV position converted from deg-min-s in the Earth coordinate frame into metres in the NED coordinate frame, $\Psi(t)$ is the yaw angle obtained from the imaginary yaw sensor, $\vartheta(t)$ is the yaw rate, and $\zeta(t)$ and $v(t)$ are the surge and sway velocities respectively.

In this system model, $F(\cdot)$ and $H(\cdot)$ are both continuous function, continuously differentiable in $X(t)$.

The $W(t)$ and $V(t)$ are both zero mean white noise for the system and measurement models respectively.

The model states are related through the following kinematically based set of functions $[F(X(t))]$ in equation (14):

$\dot{\zeta}(t) = 0$ (16)

$\dot{v}(t) = 0$ (17)

$\dot{\Psi}(t) = \vartheta(t)$ (18)

$\dot{\vartheta}(t) = 0$ (19)

$\dot{\lambda}(t) = \zeta(t) \cos \Psi(t) - v(t) \sin \Psi(t)$ (20)

$\dot{\varphi}(t) = \zeta(t) \sin \Psi(t) + v(t) \cos \Psi(t)$ (21)

The output measurements are related through the states by the identity matrix $H(X(t))$. To obtain an EKF with an effective state prediction equation in a simple form, the continuous time model of equations (16) to (21) have been linearized about the current state

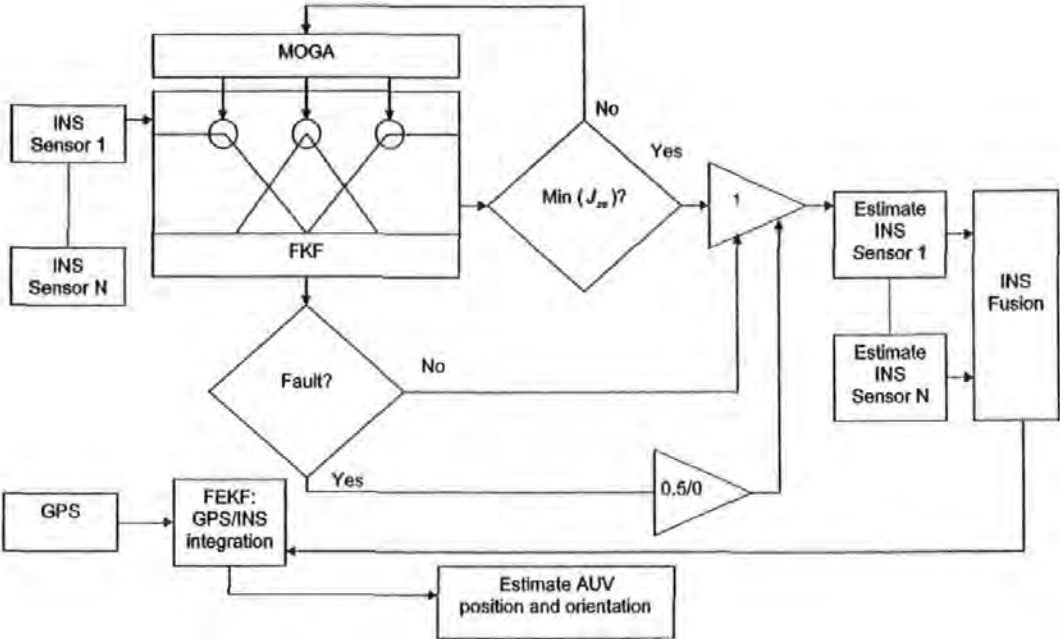


Fig. 11 Block diagram of the INS/GPS using the F(E)KF

estimates, producing

$$A(t) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\zeta(t) \sin \Psi(t) - v(t) \cos \Psi(t) & 0 & \cos \Psi(t) & -\sin \Psi(t) \\ 0 & 0 & \zeta(t) \cos \Psi(t) - v(t) \sin \Psi(t) & 0 & \sin \Psi(t) & \cos \Psi(t) \end{bmatrix} \quad (22)$$

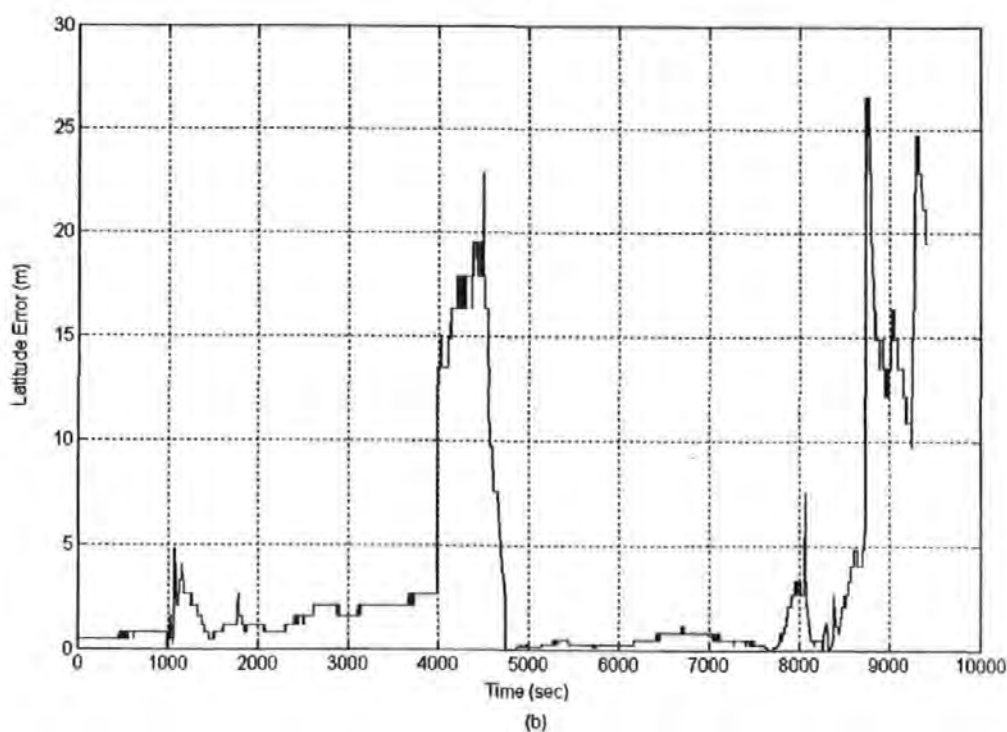
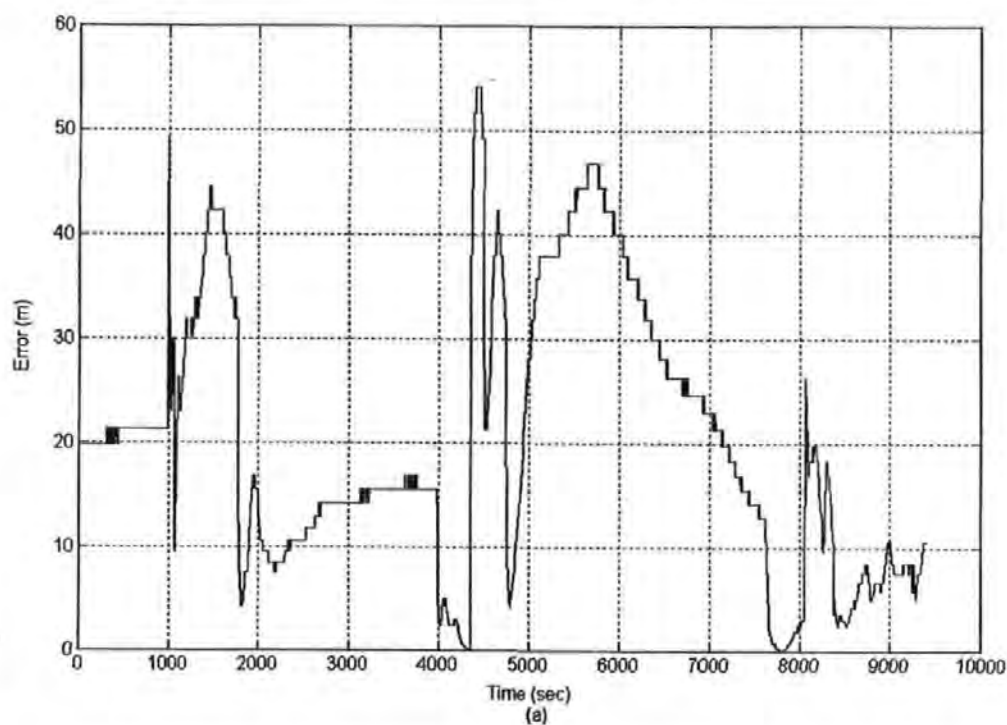


Fig. 12 (a) Longitude error; (b) Latitude error

and C is an identity matrix. Subsequent discretization with period $T = 0.125$ s of the linearized model results in an EKF algorithm similar to the LKF algorithms in the Appendix, only this time the A matrix is updated at every iteration. The initial conditions are $P_0 = \text{diag}[0.01 \text{ m}^2 \ 0.01 \text{ m}^2 \ 0.01(\text{rad}^2) \ 0.01(\text{rad/s}^2) \ 0.01 \text{ m/s}^2 \ 0.01 \text{ m/s}^2]$ and Q is made constant as $\text{diag}[10 \text{ m}^2 \ 10 \text{ m}^2 \ 0.000001 \text{ rad}^2 \ 0.01 \text{ rad/s}^2 \ 0.1 \text{ m/s}^2 \ 0.1 \text{ m/s}^2]$. As in the case of fusion of

INS sensor data discussed previously, the values of P_0 and Q here are also determined heuristically. The initial value of R is selected as $\text{diag}[20.18 \text{ m}^2 \ 3.30 \text{ m}^2 \ 0 \text{ rad}^2 \ 0 \text{ rad/s}^2 \ 0.000009 \text{ m/s}^2 \ 0.000016 \text{ m/s}^2]$. The values of $R(1, 1)$ and $R(2, 2)$ are determined by error analysis of the output of an actual GARMIN GPS 15LW receiver over a period of several hours at the University of Plymouth testing site with latitude $50^\circ 22' 33.0552''$ North

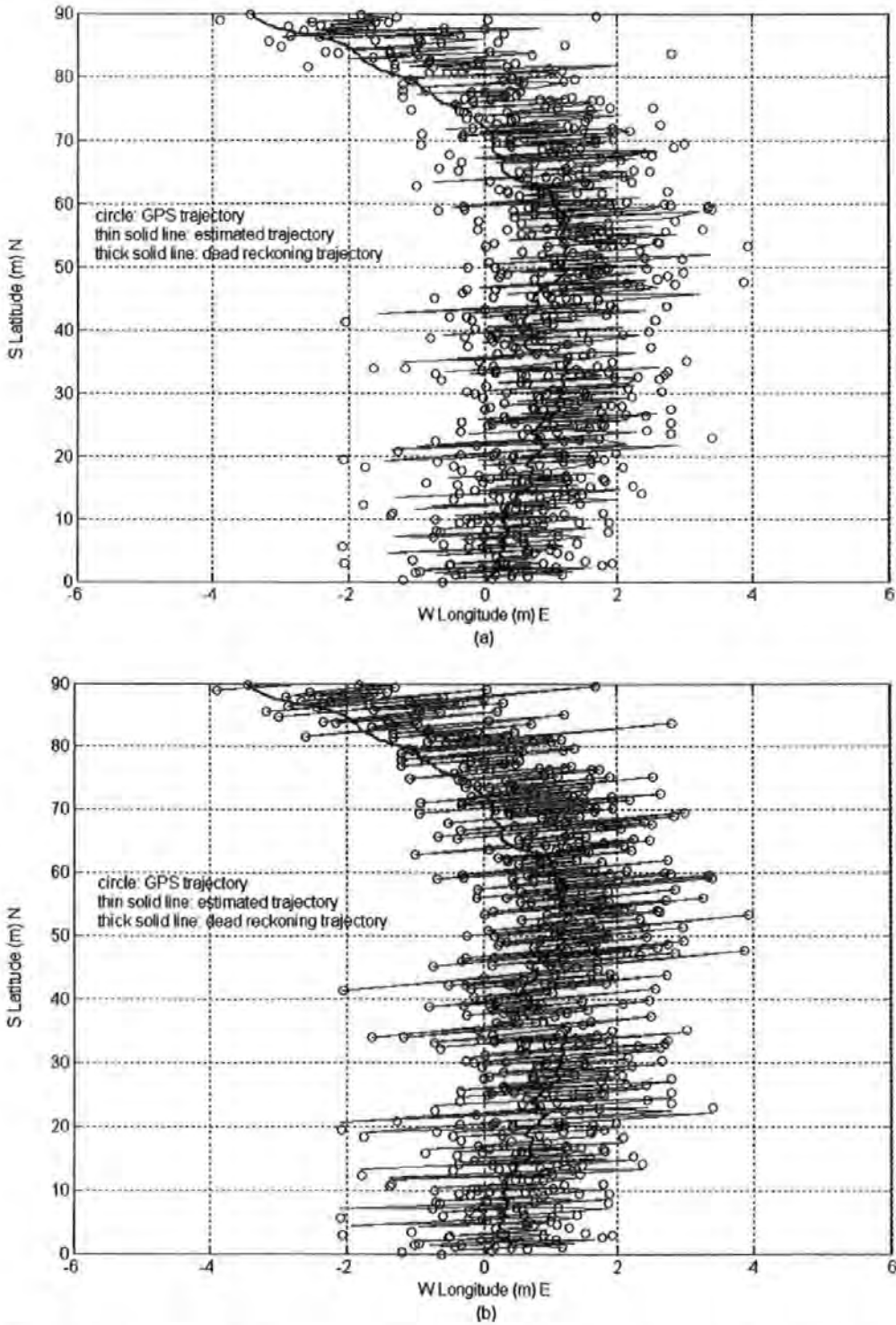


Fig. 13 (a) Initial AUV trajectory using the standard EKF; (b) initial AUV trajectory using the FEKF

and longitude $004^{\circ} 08' 21.1438''$ West. To generate the error time series shown in Fig. 12, the deg-min-s of difference between the output of the receiver and the actual known position was converted into metres, using methods available in the literature [23]. $R(5, 5)$ and $R(6, 6)$ are chosen to represent the R of an RDI Navigator Doppler Velocity Log at 1200 kHz [24]. As the output from the imaginary yaw sensor assumed to

be noise free, the initial values of $R(3, 3)$ and $R(4, 4)$ are selected as 0 rad^2 and 0 rad/s^2 respectively.

The FKF algorithm from section 2.1 is then implemented, only this time the adaptation of the (i, i) th element of R_k is made in accordance with the (i, i) th element of Δ_k . Here a single-input-single-output (SISO) FIS, as shown in Fig. 1, is used sequentially to generate the correction factors for the elements in the main diagonal

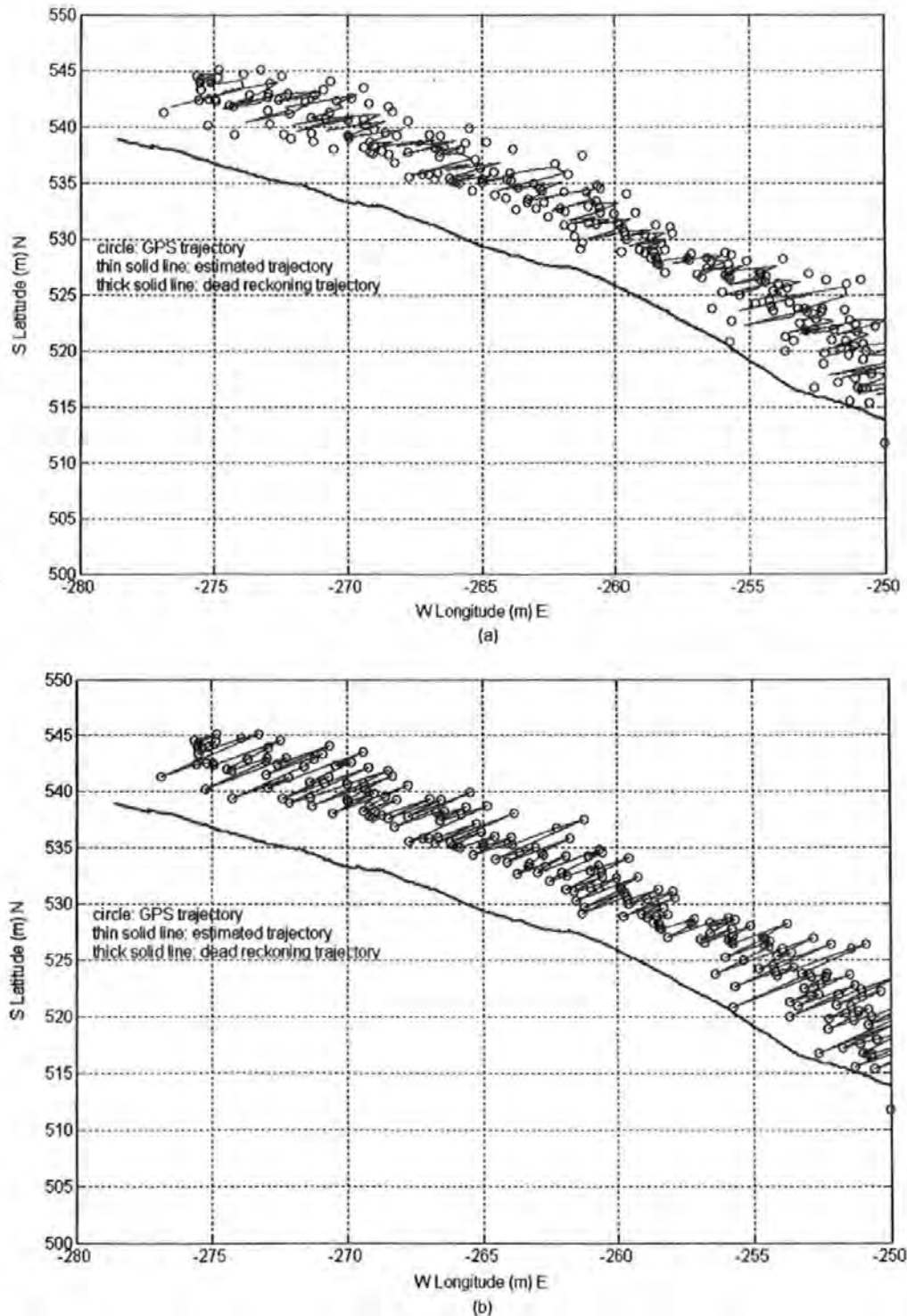


Fig. 14 (a) Final AUV trajectory using the standard EKF; (b) final AUV trajectory using the FEKF

of R_k as the following:

$$R_k(i, i) = R_{k-1}(i, i) + \Delta R_k \quad (23)$$

Figures 13 and 14 are the simulation results showing the AUV trajectory at the start and the end of its mission. The longitude and latitude of the vehicle during the course of the mission are simulated while being observed by the on-board GPS receiver with constant Gaussian noise with R values lower than the assumed initial values. On the contrary, the vehicle's surge and sway velocities are simulated being observed by a DVL with constant Gaussian noise with R values much higher than the assumed initial values. This logically will cause less weight being put on the position obtained by the GPS and more on the prediction of position obtained from the dead reckoning method (using DVL data) at the start of the simulation and conversely towards the end. Figure 13a shows how the EKF makes relatively slower adjustments to the R values compared to the adjustments made by the FEKF shown in Fig. 13b. It is clear here that only after a few samples of time, the FEKF has learned the true nature of the sensor noise and put more weight on the position obtained by the GPS receiver than on the prediction of position obtained by dead reckoning accordingly. Figure 14a shows how the EKF has still some 'confidence' on the dead reckoned position at the end of the simulated trajectory, while the FEKF shown in Fig. 14b has put 100 per cent confidence on the GPS position. The EKF results in an estimated trajectory with a root mean square error (r.m.s.e.) of 0.6157 m for longitude and 0.2626 m for latitude. With the proposed FEKF adaptation, the r.m.s.e. for longitude is 0.1098 m and 0.0158 m for latitude.

5 SUMMARY AND CONCLUSIONS

The problem with incomplete *a priori* knowledge of Q (process covariance matrix) and R (measurement covariance matrix) is considered. Within this paper, an adaptive Kalman filter approach based on the filter innovation sequence coupled with fuzzy logic optimized using a MOGA is discussed as an alternative for fusing INS sensor data and integrating INS/GPS position information. Implementation of this approach to a linearized heading model of an AUV, whose responses are measured with sensors with different noise characteristics, has shown a significant result in improving the estimation of an individual KF.

REFERENCES

- 1 Loebis, D., Sutton, R. and Chudley, J. Review of multi-sensor data fusion techniques and their application to autonomous underwater vehicle navigation. *J. Mar. Engng Technol.*, 2002, **A1**, 3–14.

- 2 Mehra, R. K. On the identification of variances and adaptive Kalman filtering. *IEEE Trans. Autom. Control*, 1970, **AC-15**(2), 175–184.
- 3 Mehra, R. K. On-line identification of linear dynamic systems with applications to Kalman filtering. *IEEE Trans. Autom. Control*, 1971, **AC-16**(1), 12–21.
- 4 Fitzgerald, R. J. Divergence of the Kalman filter. *IEEE Trans. Autom. Control*, 1971, **AC-16**(6), 736–747.
- 5 Kailath, T. An innovations approach to least-squares estimation, Part I: linear filtering in additive noise. *IEEE Trans. Autom. Control*, 1968, **AC-13**(6), 646–655.
- 6 Kailath, T. An innovations approach to least-squares estimation, Part II: linear smoothing in additive white noise. *IEEE Trans. Autom. Control*, 1968, **AC-13**(6), 655–660.
- 7 Kailath, T. The innovations approach to detection and estimation theory. *IEEE Proc.*, 1970, **58**(5), 680–695.
- 8 Magill, D. T. Optimal adaptive estimation of sampled stochastic processes. *IEEE Trans. Autom. Control*, 1965, **AC-10**(4), 434–439.
- 9 Hanlon, P. D. and Maybeck, P. S. Multiple-model adaptive estimation using a residual correlation Kalman filter bank. *IEEE Trans. Aerospace and Electronic Systems*, 2000, **36**(2), 393–406.
- 10 Chaer, W. S., Bishop, R. H. and Gough, J. A Mixture-of-experts framework for adaptive Kalman filtering. *IEEE Trans. Systems, Man and Cybernetics—Part B: Cybernetics*, 1997, **27**(3), 452–464.
- 11 Cordón, O., Herrera, F. and Lozano, M. On the combination of fuzzy logic and evolutionary computation: a short review and bibliography. In *Fuzzy Evolutionary Computation* (Ed. W. Pedrycz), 1998, pp. 57–77 (Kluwer Academic, Dordrecht, The Netherlands).
- 12 Kobayashi, K., Cheok, K. C., Watanabe, K. and Muneke, F. Accurate differential global positioning system via fuzzy logic Kalman filter sensor fusion technique. *IEEE Trans. Ind. Electronics*, 1998, **45**(3), 510–518.
- 13 Jetto, L., Longhi, S. and Vitali, D. Localization of a wheeled mobile robot by sensor data fusion based on a fuzzy logic adapted Kalman filter. *Control Eng. Practice*, 1999, **7**, 763–771.
- 14 Loebis, D., Sutton, R. and Chudley, J. A fuzzy Kalman filter for accurate navigation of an autonomous underwater vehicle. In *Proceedings of the 1st IFAC Workshop on Guidance and Control of Underwater Vehicles*, Newport, South Wales, 2003, pp. 161–166.
- 15 Mohamed, A. H. and Schwarz, K. P. Adaptive Kalman filtering for INS/GPS. *J. Geodesy*, 1999, **73**, 193–203.
- 16 Escamilla-Ambrosio, P. J. and Mort, N. A. Hybrid Kalman filter-fuzzy logic multisensor data fusion architecture with fault tolerant characteristics. In *Proceedings of the 2001 International Conference on Artificial Intelligence*, Las Vegas, Nevada, 2001, pp. 361–367.
- 17 Goldberg, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*, 1989 (Addison Wesley, Reading, Massachusetts).
- 18 Fonseca, C. M. and Fleming, P. J. Multiobjective optimisation and multiple constraint handling with evolutionary algorithms. I. A unified formulation. *IEEE Trans. Systems, Man and Cybernetics*, 1998, **A**(28), 26–37.

- 19 Goldberg, D. E. and Richardson, J. Genetic algorithms with sharing for multimodal function optimization. In Proceedings of the 2nd International Conference on *Genetic Algorithms* (Ed. J. J. Grefenstette), 1987, Cambridge, MA, USA, pp. 41–49.
- 20 Deb, K. and Goldberg, D. E. An investigation of niche and species formation in genetic optimization. In Proceedings of the 3rd International Conference on *Genetic Algorithms* (Ed. J. D. Schaffer), 1989, Arlington, VA, USA, pp. 42–50.
- 21 Lea, R. K. Control of a tethered underwater flight vehicle. PhD thesis, University of Southampton, 1998.
- 22 Fossen, T. I. *Guidance and Control of Ocean Vehicles*, 1995 (John Wiley, New York).
- 23 Kennedy, M. *The Global Positioning System and GIS*, 2002 (Taylor and Francis, London).
- 24 Grenon, G., An, P. E., Smith, S. M. and Healey, A. J. Enhancement of the Inertial Navigation System for the Morpheus autonomous underwater vehicle. *IEEE J. Oceanic Engng*, 2001, 26(4), 548–560.

APPENDIX

Kalman filter algorithm

Given a discrete-time controlled process described by the linear stochastic difference equations

$$\mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_k \quad (24)$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \quad (25)$$

where \mathbf{x}_k is an $n \times 1$ system state vector, \mathbf{A}_k is an $n \times n$ transition matrix, \mathbf{u}_k is an $l \times 1$ vector of the input forcing function, \mathbf{B}_k is an $n \times l$ matrix, \mathbf{w}_k is an $n \times 1$ process noise vector, \mathbf{z}_k is an $m \times 1$ measurement vector, \mathbf{H}_k

is an $m \times n$ measurement matrix and \mathbf{v}_k is an $m \times 1$ measurement noise vector. Both \mathbf{w}_k and \mathbf{v}_k are assumed to be uncorrelated zero-mean Gaussian white noise sequences with covariance given by

$$E[\mathbf{w}_k \mathbf{w}_i^T] = \begin{cases} \mathbf{Q}_k, & i = k \\ 0, & i \neq k \end{cases} \quad (26)$$

$$E[\mathbf{v}_k \mathbf{v}_i^T] = \begin{cases} \mathbf{R}_k, & i = k \\ 0, & i \neq k \end{cases} \quad (27)$$

$$E[\mathbf{w}_k \mathbf{v}_i^T] = 0, \quad \text{for all } k \text{ and } i \quad (28)$$

The KF algorithm can be organized into a time update and measurement update equations:

Time update equations:

$$\hat{\mathbf{x}}_{k+1}^- = \mathbf{A}_k \hat{\mathbf{x}}_k + \mathbf{B}_k \mathbf{u}_k \quad (29)$$

$$\mathbf{P}_{k+1}^- = \mathbf{A}_k \mathbf{P}_k \mathbf{A}_k^T + \mathbf{Q}_k \quad (30)$$

Measurement update equations:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T [\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k]^{-1} \quad (31)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k [\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-] \quad (32)$$

$$\mathbf{P}_k = [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k] \mathbf{P}_k^- \quad (33)$$

The measurement update equations incorporate a new observation into the *a priori* estimate from the time update equations to obtain an improved *a posteriori* estimate. In the time and measurement update equations, $\hat{\mathbf{x}}_k$ is an estimate of the system state vector \mathbf{x}_k , \mathbf{K}_k is the Kalman gain and \mathbf{P}_k is the covariance matrix of the state estimation error.



ANCON
CIVITAS
DORICA
CVM PORTU
TRAIANI

Ancona, Italy
July 7-9, 2004

IFAC Conference on

**CONTROL APPLICATIONS IN
MARINE SYSTEMS**

Università Politecnica delle Marche

Dipartimento di Ingegneria Informatica, Gestionale e dell'Automazione

A SOFT COMPUTING METHOD FOR AN AUV NAVIGATION SYSTEM WITH PSEUDO-REAL-TIME APPLICABILITY

D. Loebis, R. Sutton, J. Chudley

*Marine and Industrial Dynamic Analysis Research Group,
Reynolds Building, School of Engineering, The University of Plymouth,
Drake Circus, Plymouth, PL4 8AA*

Abstract: This paper describes the implementation of a soft computing method based on fuzzy logic and multiobjective genetic algorithm techniques to adapt the parameters of an error-state complementary Kalman filter (ESCKF) to enhance the accuracy of an autonomous underwater vehicle (AUV) navigation system. In the ESCKF, inertially-derived quantities from an inertial navigation system (INS) sensor are combined with direct measurements of the same quantities by use of the global positioning system (GPS) and other aiding sensors. The backlash of the integration processes however, is that errors can grow rapidly and the values obtained therein can drift off the true value significantly. By contrast, the directly-measured data contain high frequency noise with bounded error. This instinctively suggests integrating the two sets of quantities, which is exactly what the ESCKF does. To maintain the stability and performance of the ESCKF, which is likely to deteriorate when the assumed error and noise characteristics do not reflect the true ones, a fuzzy logic based scheme is used to make these values adaptive. The choice of fuzzy membership functions for this scheme is first carried out using a heuristic approach and further refined using a multiobjective genetic algorithm method. Copyright© 2004 IFAC

Keywords: Autonomous underwater vehicles, navigation, Kalman filters, fuzzy logic, genetic algorithm, multiobjective optimization

1. INTRODUCTION

The development of AUVs for scientific, military and commercial purposes in applications such as ocean surveying, unexploded ordnance hunting, and cable tracking and inspection requires the corresponding development of navigation systems. Such systems are necessary to provide knowledge of vehicle position and attitude. The need for accuracy in such systems is paramount: erroneous position and attitude data can lead to a meaning-

less interpretation of the collected data or even to a catastrophic failure of an AUV. A growing number of research groups around the world are developing integrated navigation systems utilising INS and GPS. However, few of these works make explicit the essential need for fusion of several INS sensors that enable the users to maintain the accuracy or even to prevent a complete failure of this part of the navigation system, before being integrated with the GPS. Several estimation methods have been used in the past for multisens-

sensor data fusion and integration purposes. To this end, the Kalman filter (KF) and its variants have been popular methods in the past and interest in developing the algorithms has continued to the present day.

However, a significant difficulty in designing a KF can often be traced to incomplete *a priori* knowledge of the process noise covariance matrix (Q) and measurement noise covariance matrix (R). In most practical applications, these matrices are initially estimated or even unknown. The problem here is that the optimality of the estimation algorithm in the KF setting is closely connected to the quality of *a priori* information about the Q and R (Mehra, 1970). It has been shown that insufficiently known *a priori* filter statistics can reduce the precision of the estimated filter states or introduce biases to their estimates. In addition, incorrect *a priori* information can lead to practical divergence of the filter. From the aforementioned it may be argued that the conventional KF with fixed Q and/or R should be replaced by an adaptive estimation formulation. In this paper, a novel fuzzy error-state complementary Kalman filter (FESCKF) is proposed. With this method, a KF with an error-state model obtained using first order Markov processes and error data analysis is used in parallel with fuzzy logic techniques to adjust R . A further improvement can be achieved using multiobjective genetic algorithm (MOGA) techniques, whereby the fuzzy membership functions are adjusted to produce the most optimum result.

The structure of the paper is as follows: section 2 introduces the concept of the ESCKF and the derivation of the process and measurement model and the associated noise covariance matrices. Section 3 discusses the proposed KF adaptation mechanism followed by fuzzy membership function optimization (FESCKF). Section 4 and 5 provide simulation results and concluding remarks respectively.

2. ESCKF MODELLING

2.1 The Concept of ESCKF

Brown and Hwang (1997) discuss the advantages of the ESCKF method over the total state Kalman filter. The most important advantage is that any nonlinear relationship between the process dynamics in the inertial system and the measurement relationships can be removed in a differencing operation, and the filter becomes linear. This linearity condition is required by the Kalman filter. This condition can also lead to faster codes

execution as linearisation operations are relatively slow to execute.

KF algorithms are widely available in the literature. The interested reader can refer to Brown and Hwang (1997). Works on ESCKF however, are very limited especially in the field of AUV navigation systems. An example of this work can be found in Gustaffson *et al.* (2001). Like in the KF, the ESCKF algorithm can be divided into two major parts: the measurement update and the time update. In ESCKF, the measurement update is obtained by subtracting the direct measurements from the computed version of the same quantities. By doing this, the true values cancel each other out and what remains is the difference between the measurement errors and drift errors. In the time update, the estimates are obtained by subtracting the estimated drift errors from the forward filter pass from the computed version of the same quantities.

In this paper, measurement errors from an accelerometer and a gyroscope (assembled in an inertial measurement unit (IMU)), a TCM2 electronic compass and a GPS receiver unit are estimated and modelled using first order Markov processes which are defined in the following manner:

$$\dot{x} = -\frac{1}{\tau} \cdot x + \gamma \quad (1)$$

In (1), x is the error process to be modelled, τ is the time constant of the assumed Markov process and γ is white noise. For modelling purposes, all sensor data have been collected in static conditions for a period of approximately 2.5 hours. For the same purpose, three different frames of reference are defined. The body-fixed (b) frame of reference is aligned to the axes of the AUV, where forward-starboard-down correspond to $x-y-z$. These need to be transformed to the geographical (g) frame of reference, where $x-y-z$ correspond to North-East-Down. For these particular application, the measurements in question are 3D accelerations, as well as angular rates measured by the IMU. Earth-centred Earth-fixed (ECEF) frame is where the GPS latitude and longitude are defined. The following subsections give the derivation of the process matrix (F) and the corresponding noise covariance matrix (Q)

2.2 Process and Noise Covariance Matrix

The elements of the state of the ESCKF are defined as follows:

$$x = [x_d^g \ y_d^g \ \psi_d^g \ x_e^g \ y_e^g \ \psi_e^g \ r_e^b \ u_e^h \ v_e^h] \quad (2)$$

In (2) the subscripts d and e denote drift and sensor errors respectively. Superscripts g , b and

h denote geographical frame, body-fixed-frame and horizontal frame respectively. Drift errors in position, x_d^g and y_d^g stem from the error in the integrated acceleration (u_e^h and v_e^h), and compass error ψ_e^g . ψ_d^g is heading drift error which comes from the error in the integrated yaw rate. Measurement errors in the position blend are respectively represented by the states x_e^g and y_e^g . Finally, state r_e^b represents gyroscope's yaw error.

The differential equation describing the relationship between dead reckoning position and the horizontal velocity from the integrated acceleration is given as:

$$\begin{bmatrix} \dot{x}_c^g \\ \dot{y}_c^g \end{bmatrix} = \begin{bmatrix} \cos \psi_m^g & -\sin \psi_m^g \\ \sin \psi_m^g & \cos \psi_m^g \end{bmatrix} \cdot \begin{bmatrix} u_m^h \\ v_m^h \end{bmatrix} \quad (3)$$

In (3) the subscript c denotes *computed* and m denotes *measured*. Expanding this into true values and errors yields:

$$\begin{bmatrix} \dot{x}^g + \dot{x}_d^g \\ \dot{y}^g + \dot{y}_d^g \end{bmatrix} = \begin{bmatrix} \cos(\psi^g + \psi_e^g) & -\sin(\psi^g + \psi_e^g) \\ \sin(\psi^g + \psi_e^g) & \cos(\psi^g + \psi_e^g) \end{bmatrix} \cdot \begin{bmatrix} u_m^h \\ v_m^h \end{bmatrix} \quad (4)$$

Expansion of (4) by applying trigonometric formulas and by assuming that the measurement error is sufficiently small whereby the relations $\cos \psi_e^g \cong 1$ and $\sin \psi_e^g \cong \psi_e^g$ holds, yields:

$$\begin{aligned} \begin{bmatrix} \dot{x}^g + \dot{x}_d^g \\ \dot{y}^g + \dot{y}_d^g \end{bmatrix} &= \begin{bmatrix} \cos \psi^g - \sin \psi^g \cdot \psi_e^g \\ \sin \psi^g + \cos \psi^g \cdot \psi_e^g \end{bmatrix} \cdot \begin{bmatrix} u^h \\ v^h \end{bmatrix} + \\ &\begin{bmatrix} -\sin \psi^g - \cos \psi^g \cdot \psi_e^g \\ \cos \psi^g - \sin \psi^g \cdot \psi_e^g \end{bmatrix} \cdot \begin{bmatrix} u_e^h \\ v_e^h \end{bmatrix} + \\ &\begin{bmatrix} \cos(\psi^g + \psi_e^g) - \sin(\psi^g + \psi_e^g) \\ \sin(\psi^g + \psi_e^g) \cos(\psi^g + \psi_e^g) \end{bmatrix} \cdot \begin{bmatrix} u_e^h \\ v_e^h \end{bmatrix} \end{aligned} \quad (5)$$

By substituting true values into (3) and subtracting the result from (5), gives

$$\begin{bmatrix} \dot{x}_d^g \\ \dot{y}_d^g \end{bmatrix} = \begin{bmatrix} -\sin \psi^g \cdot \psi_e^g - \cos \psi^g \cdot \psi_e^g \\ \cos \psi^g \cdot \psi_e^g - \sin \psi^g \cdot \psi_e^g \end{bmatrix} \cdot \begin{bmatrix} u^h \\ v^h \end{bmatrix} + \begin{bmatrix} \cos(\psi^g + \psi_e^g) - \sin(\psi^g + \psi_e^g) \\ \sin(\psi^g + \psi_e^g) \cos(\psi^g + \psi_e^g) \end{bmatrix} \cdot \begin{bmatrix} u_e^h \\ v_e^h \end{bmatrix} \quad (6)$$

To explicitly relate drift errors in position to the accelerometer error, trivial alteration is applied to the first term of right hand side of (6). Further, by assuming that the current estimate of heading and velocities from the Kalman filter ($\hat{\psi}^g, \hat{u}$ and \hat{v}) are sufficiently close to the true heading (ψ^g) and velocities and also by keeping in mind that $\psi^g + \psi_e^g = \psi_m^g$, (6) becomes:

$$\begin{bmatrix} \dot{x}_d^g \\ \dot{y}_d^g \end{bmatrix} = \begin{bmatrix} -\sin \hat{\psi}^g \cdot \hat{u}^h - \cos \hat{\psi}^g \cdot \hat{v}^h \\ \cos \hat{\psi}^g \cdot \hat{u}^h - \sin \hat{\psi}^g \cdot \hat{v}^h \end{bmatrix} \cdot \begin{bmatrix} \psi_e^g \\ \psi_e^g \end{bmatrix} +$$

$$\begin{bmatrix} \cos \psi_m^g & -\sin \psi_m^g \\ \sin \psi_m^g & \cos \psi_m^g \end{bmatrix} \cdot \begin{bmatrix} u_e^h \\ v_e^h \end{bmatrix} \quad (7)$$

As in the case with position drift, the heading drift error evolution (ψ_d^g) will be directly dependent on the yaw rate sensor error (r_e^b), because it is rotated into horizontal frame before it is integrated up to yield an alternative heading. The differential equation describing the relationship between body-fixed angular rates and horizontal heading can be written as:

$$\dot{\psi}_c^h = \frac{\sin \phi_m^h \cdot q_m^b + \cos \phi_m^h \cdot r_m^b}{\cos \theta_m^h} \quad (8)$$

In (8) ϕ , θ , q and r are the roll, pitch, pitch rate and yaw rate respectively. Expanding computed heading and measured yaw rate into true values plus drift and sensor errors, and using the assumptions that $q^b \cong q_m^b$, $\psi^h \cong \psi_m^h$ and $\theta \cong \theta_m^h$, true values cancel each other out. Observing that the computed heading is initialised by a TCM2 reading, the following expression gives the sought relation between drift error and yaw rate sensor error:

$$\dot{\psi}_d^g = \frac{\cos \phi_m^h}{\cos \theta_m^h} \cdot r_e^b \quad (9)$$

The rest of the diagonal elements in the process matrix describe the sensor error processes, which are modelled using a first order Markov processes. Based on the derivations and assumptions in (3) through (9), the process matrix can be written as in (10) given in the next page.

The variance of the process noise for a Markov error model can be described as in Brown and Hwang (1997),

$$\text{variance}[w_k] = (1 - e^{-\frac{\Delta t}{\tau}}) \cdot \text{variance}[x_k] \quad (11)$$

In (11), $e^{-\frac{\Delta t}{\tau}}$ is the state transition parameter for the Markov error model. Δt is the discrete time interval and τ is the time constant. By taking the approximation: $e^{-\beta \Delta t} \approx 1 - \beta \cdot \Delta t$, where $\beta = \tau^{-1}$, the following is true:

$$\text{variance}[w_k] = (2\beta \Delta t - (\beta \Delta t)^2) \cdot \text{variance}[x_k] \quad (12)$$

According to the process model, the heading drift error state represents the integrated yaw error state, in effect an integrated Markov process. The process noise covariance matrix for these two states can be defined as in Brown and Hwang (1997),

$$Q = \begin{bmatrix} E[\psi_d^g \psi_d^g] & E[\psi_d^g r_e^b] \\ E[\psi_d^g r_e^b] & E[r_e^b r_e^b] \end{bmatrix} \quad (13)$$

where

$$F = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -\sin \hat{\psi}^g \cdot \hat{u}^h - \cos \hat{\psi}^g \cdot \hat{v}^h & 0 & \cos \psi_m^g & -\sin \psi_m^g \\ 0 & 0 & 0 & 0 & 0 & \cos \hat{\psi}^g \cdot \hat{u}^h - \sin \hat{\psi}^g \cdot \hat{v}^h & 0 & \sin \psi_m^g & \cos \psi_m^g \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{\cos \phi_m^h}{\cos \theta_m^h} & 0 & 0 \\ 0 & 0 & 0 & \frac{-1}{\tau_x} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{-1}{\tau_y} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{-1}{\tau_\psi} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{-1}{\tau_\psi} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-1}{\tau_u} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-1}{\tau_v} \end{bmatrix} \quad (10)$$

$$Q_{11} = \frac{2\sigma^2}{\beta} \left[\Delta t - \frac{2}{\beta}(1 - \phi) + \frac{1}{2\beta}(1 - \phi^2) \right] \quad (14)$$

$$Q_{12} = Q_{21} = 2\sigma^2 \left[\frac{1}{\beta}(1 - \phi) + \frac{1}{2\beta}(1 - \phi^2) \right] \quad (15)$$

$$Q_{22} = \sigma^2(1 - \phi^2) \quad (16)$$

where β is the inverse of the Markov time constant, σ^2 is the process noise variance of the yaw rate Markov error and $\phi = e^{-\frac{\Delta t}{\tau}}$ and defined as before.

For the first two states in the process model, x_d^g and y_d^g , the analysis is more complicated and for this reason, the noise covariance matrix of the two states are obtained from an empirical result, and provision for the adjustment method has been made and will be reported in the future.

2.3 Measurement and Noise Covariance Matrix

The measurement matrix H relates the available measurement updates to the element in the state vector and takes the following form:

$$H = \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \quad (17)$$

The measurement noise covariance matrix R_k is determined empirically and given as:

$$R_k = \begin{bmatrix} \sigma_{X-Position}^2 & 0 & 0 \\ 0 & \sigma_{Y-Position}^2 & 0 \\ 0 & 0 & \sigma_{Heading}^2 \end{bmatrix} \quad (18)$$

where $\sigma_{X-Position}^2$, $\sigma_{Y-Position}^2$, $\sigma_{Heading}^2$ are the variance in X, Y direction and heading respectively. These values will be adapted using the algorithm discussed in the next section.

3. THE ADAPTIVE TUNING OF KALMAN FILTER ALGORITHM

Over the past few years, only a few publications in the area of adaptive Kalman filtering can be found in the literature. One of the most popular method is innovation adaptive estimation (IAE). The innovation Inn_k at sample time k is the difference between the real measurement z_k received by the filter and its estimated (predicted) value \hat{z}_k . The predicted measurement is the projection of the filter predicted states x_k^- onto the measurement space through the measurement matrix H_k . Innovation represents additional information available to the filter as a result of the new measurement z_k . The occurrence of data with statistics different from the *a priori* information will first show up in the innovation vector. For this reason the innovation sequence represents the information content in the new observation and is considered the most relevant source of information to the filter adaptation.

Herein, the IAE approach coupled with fuzzy logic techniques with membership functions designed using heuristic methods and further refined using MOGA is used to adjust the R matrix of the ESCKF. Initial work on this approach can be found in Loebis *et al.* (2003).

3.1 Fuzzy error state complementary Kalman filter

In this sub-section, an on-line innovation-based adaptive scheme of the ESCKF to adjust the R matrix employing the principles of fuzzy logic is presented. The fuzzy logic is chosen mainly because of its simplicity. This motivates the interest in the topic, as testified by related papers which have been appearing in the literature (Loebis *et al.*, 2003; Escamilla-Ambrosio and Mort, 2001).

The FESCKF proposed herein is based on the IAE approach using a technique known as covariance-matching (Mehra, 1970). The basic idea behind the technique is to make the actual value of the covariance of the innovation sequences match its theoretical value.

The actual covariance is defined as an approximation of the Inn_k sample covariance through averaging inside a moving estimation window of size M (Mohamed and Schwarz, 1999) which takes the following form:

$$\hat{C}_{\text{Inn}_k} = \frac{1}{M} \sum_{j=j_0}^k \text{Inn}_j \cdot \text{Inn}_j^T \quad (19)$$

where $j_0 = k - M + 1$ is the first sample inside the estimation window. An empirical experiment is conducted to choose the window size M . From experimentation it was found that a good size for the moving window in (19) is 15. The theoretical covariance of the innovation sequence is defined as

$$S_k = H_k \cdot P_k^- \cdot H_k^T + R_k \quad (20)$$

The logic of the adaptation algorithm using covariance matching technique can be qualitatively described as follows. If the actual covariance value \hat{C}_{Inn_k} is observed, whose value is within the range predicted by theory S_k and the difference is very near to zero, this indicates that both covariances match almost perfectly and only a small change is needed to be made on the value of R . If the actual covariance is greater than its theoretical value, the value of R should be decreased. On the contrary, if \hat{C}_{Inn_k} is less than S_k , the value of R should be increased. This adjustment mechanism lends itself very well to being dealt with using a fuzzy-logic approach based on rules of the kind:

$$\text{IF } \langle \text{antecedent} \rangle \text{ THEN } \langle \text{consequent} \rangle \quad (21)$$

where antecedent and consequent are of the form $\nu \in O_i$, $\kappa \in L_i$, $i = 1, 2, \dots$ respectively. Where ν and κ are the input and output variables, respectively, and O_i and L_i are the fuzzy sets.

To implement the above covariance matching technique using the fuzzy logic approach, a new variable called delta_k , is defined to detect the discrepancy between \hat{C}_{Inn_k} and S_k . The following fuzzy rules of the kind (21) are used:

$$\text{IF } \langle \text{delta}_k \cong 0 \rangle \text{ THEN } \langle R_k \text{ is unchanged} \rangle \quad (22)$$

$$\text{IF } \langle \text{delta}_k > 0 \rangle \text{ THEN } \langle R_k \text{ is decreased} \rangle \quad (23)$$

$$\text{IF } \langle \text{delta}_k < 0 \rangle \text{ THEN } \langle R_k \text{ is increased} \rangle \quad (24)$$

Thus R is adjusted according to,

$$R_k = R_{k-1} + \Delta R_k \quad (25)$$

where ΔR_k is added or subtracted from R at each instant of time. Here delta_k is the input to the fuzzy inference system (FIS) and ΔR_k is the output.

3.2 Fuzzy membership functions optimization

MOGA is used here to optimize the membership functions of the FESCKF. To translate the FESCKF membership functions to a representation useful as genetic material, they are parameterised with real-valued variables. Each of these variables constitutes a gene of the chromosomes for the MOGA. Boundaries of chromosomes are required for the creation of chromosomes in the right limits so that the MOGA is not misled to some other area of search space. The technique adopted in this paper is to define the boundaries of the output membership functions according to the furthest points and the crossover points of two adjacent membership functions. In other words, the boundaries of FESCKF consist of three real-valued chromosomes (*Chs*), as in Figure 1. The

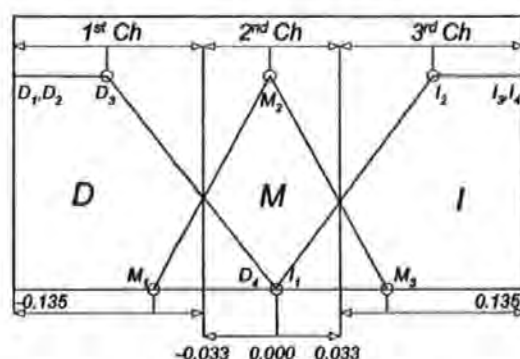


Fig. 1. Membership function and boundaries of R_k

trapezoidal membership functions' two furthest points, -0.135 (D_1), -0.135 (D_2) and 0.135 (I_3), 0.135 (I_4) of FESCKF, remain the same in the GA's description to allow a similar representation as the fuzzy system's definition. As can be seen from Figure 1, D_3 and M_1 can change value in the 1st *Ch* boundary, D_4 , M_2 and I_1 in the 2nd *Ch* boundary, and finally, M_3 and I_2 in 3rd *Ch*.

4. SIMULATION RESULTS

In this section the FESCKF algorithm is applied to a set of simulated sensor data, i.e. latitude and longitude data from a GPS unit, 3D accelerometer and gyroscope data from four IMUs located in different parts of the vehicle, and yaw data from four TCM2s located in close proximity to the IMUs. Herein, these sensors are used to capture

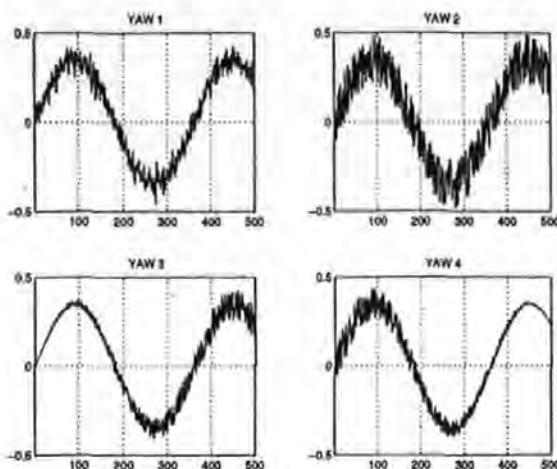


Fig. 2. Yaw output from TCM2s with (a) constant Gaussian noise1, (b) constant Gaussian noise2, (c) uniform noise increasing with time and (d) uniform noise decreasing with time

the position and attitude of the vehicle as a sinusoidal input applied to its rudder. Although there are redundancy on the 3D acceleration and attitude data, due to limited space, it is decided to focus the discussion in this paper on the fusion of the yaw data produced by the TCM2s. The yaw rates produced by the IMUs are integrated once to produce a computed version of the corresponding yaw data. As discussed in Section 2, the yaw measurement update is obtained by subtracting the yaw (TCM2s) direct measurements from the yaw (gyroscopes) computed version. Figure 2 shows the output of the TCM2s. The initial R was assumed to be $\text{diag}[500m^2 \ 500m^2 \ 0.1rad^2]$, $\hat{x}_0 = 0$, $P_0 = 0.01I_9$. The value of R was first adapted using the FESCKF with membership functions designed heuristically and further refined using MOGA with the parameters shown in Table 1. For comparison purposes, the following performance measures were adopted:

$$J_{zv} = \sqrt{\frac{1}{n} \sum_{k=1}^n (za_k - z_k)^2} \quad (26)$$

$$J_{ze} = \sqrt{\frac{1}{n} \sum_{k=1}^n (za_k - \hat{z}_k)^2} \quad (27)$$

where za_k is the actual value of the yaw, z_k is the measured yaw, \hat{z}_k is the estimated yaw at an instant of time k and n = number of samples. The performance comparison is presented in Table 2. It is clear that the J_{ze} -s of each sensor in both non-MOGA and MOGA always outperform the corresponding J_{zv} -s. It is also clear that the J_{ze} -s of MOGA case always produce a better result than the non-MOGA case. Most importantly, the J_{ze} -s of the fused sensor are better compared to the J_{ze} -s of the individual sensor.

Parameters	Values
Number of objective functions	5
Number of generation	200
Number of individual per generation	25
Generation gap in selection operation	0.95
Rate in rate in recombination operation	0.8
rate in mutation operation	0.09

Table 1. MOGA parameters

Sensor	$J_{zv}(\text{rad})$	$J_{ze}(\text{rad})$	
		Non-MOGA	MOGA
Sensor 1	0.0393	0.0379	0.0324
Sensor 2	0.0799	0.0793	0.0668
Sensor 3	0.0341	0.0293	0.0285
Sensor 4	0.0350	0.0339	0.0299
Fused		0.0245	0.0184

Table 2. Performance comparison

5. CONCLUDING REMARKS

A novel method to obtain an accurate AUV navigation system is proposed. The method is based on the ESCKF coupled with fuzzy logic to adjust the value of measurement noise covariance matrix R . MOGA is proposed to further refine the result. The simulation results presented in this paper have shown the ability of the proposed algorithm to produce a significant improvement over the conventional method.

REFERENCES

- Brown, R. G. and P. Y. C. Hwang (1997). *Introduction to Random Signals and Applied Kalman Filtering*. 3rd Ed. John Wiley and Sons.
- Escamilla-Ambrosio, P. J. and N. Mort (2001). A Hybrid Kalman Filter-Fuzzy Logic Multisensor Data Fusion Architecture with Fault Tolerant Characteristics. In: *Proc. of the 2001 International Conference on Artificial Intelligence*. Las Vegas, NV, USA. pp. 361-367.
- Gustaffson, E., E. An and S. Smith (2001). A Postprocessing Kalman Smoother for Underwater Vehicle Navigation. In: *Proc. 12th International Symposium on Unmanned Un-thetered Submersible Technology*. New Hampshire, NH, USA. pp. 1-7.
- Loebis, D., R. Sutton and J. Chudley (2003). A Fuzzy Kalman Filter for Accurate Navigation of an Autonomous Underwater Vehicle. In: *Proc. 1st IFAC Workshop on Guidance and Control of Underwater Vehicles*. Newport, South Wales, UK. pp. 161-166.
- Mehra, R. K. (1970). On the Identification of Variances and Adaptive Kalman Filtering. *IEEE Transactions on Automatic Control* AC-16(1), 12-21.
- Mohamed, A. H. and K. P. Schwarz (1999). Adaptive Kalman Filtering for INS/GPS. *Journal of Geodesy* 73, 193-203.

THE APPLICATION OF SOFT COMPUTING TECHNIQUES TO AN INTEGRATED NAVIGATION SYSTEM OF AN AUV

D. Loebis*, R. Sutton*, J. Chudley*, W. Naeem*,
F. R. Dalglish**, S. Tetlow**

* Marine and Industrial Dynamic Analysis Research Group,
Reynolds Building, School of Engineering, The University of Plymouth,
Drake Circus, Plymouth, PL4 8AA

** Offshore Technology Centre, Cranfield University,
Bedfordshire, Cranfield, MK43 0AL

Abstract: This paper describes the implementation of a soft computing method based on fuzzy logic and multiobjective genetic algorithm (MOGA) techniques to adapt the parameters of an error-state complementary Kalman filter (ESCKF) to enhance the accuracy of an autonomous underwater vehicle (AUV) integrated navigation system. In the ESCKF, inertially-derived quantities from an inertial navigation system (INS) sensor are combined with direct measurements of the same quantities by use of the global positioning system (GPS) when the vehicle is on the surface and the velocity estimator output from a visual navigation system (VNS) based on laser stripe illumination methodology when the vehicle is in close contact to the bottom of the sea whilst performing an underwater mission. This strategy will alleviate the need for frequent excursions to the surface to obtain a GPS fix to reset the navigation solution produced by the INS that tends to drift after a certain period of time. This technique exploits the complementary error characteristics in such a way so as to produce optimal estimates in terms of minimum variance. To this end, errors of the sensors are modelled using first-order Markov processes and error data analysis is undertaken to determine the respective time constants and variances. To maintain the stability and performance of the ESCKF, which is likely to deteriorate when the assumed error and noise characteristics do not reflect the true ones, a fuzzy logic based scheme is used to make these values adaptive. The choice of fuzzy membership functions for this scheme is first carried out using a heuristic approach and further refined using a MOGA method. *Copyright©2004 IFAC*

Keywords: Autonomous underwater vehicles, navigation, visual motion, Kalman filters, fuzzy logic, genetic algorithm, multiobjective optimization

1. INTRODUCTION

A deep mobile target has been converted into a rudimentary autonomous underwater vehicle (AUV) known as *Hammerhead*. A three year co-

operative research project funded by the Engineering and Physical Sciences Research Council involving both the University of Plymouth (UoP) and Cranfield University (CU) has the objectives of designing and developing an interactive nav-

igation system consisting of a visual navigation subsystem (VNS) and an inertial navigation system/global positioning subsystem (INS/GPS) to interact with an appropriate guidance and control system. VNS has been widely adopted as a navigation methodology for AUVs as it has the capabilities to provide high precision and high quality measurements from image data to derive accurate relative position information. Advanced VNS applications can attempt to provide AUV global position updates, while simultaneously creating a mosaic, or composite image map of a seabed (Fleischer *et al.*, 1997) and matching current image with viewing mosaic map. VNS has also been used for tracking and cable following (Balasuriya and Ura, 1999). In this work various image-processing techniques to extract position measurements or to identify a specific feature of an object from live video imagery are used. Given the high resolution of digital imaging, measurement accuracies on the order of millimetres and precise feature identification can be achieved. However, the methods employed are limited in regimes where the object or terrain of interest is within both the field of view (FoV) and visual range of the camera. In addition to that, there is a need for artificial light, which increases the expense and power consumption of the vehicles. The fusion of VNS with INS measurement data can be proposed as a potential solution to these problems, as INS measurements are not affected by the aforementioned factors (Fleischer *et al.*, 1997; Balasuriya and Ura, 1999). Through the technique of dead reckoning, the position of an AUV can be inferred by integrating the fused VNS and INS measurements. The problem here is that the dead reckoning is only accurate for short time durations; since the measurement noise is integrated along with the signals, the error on position accumulates quite quickly. Consequently, an external reset mechanism is required. The use of GPS to provide periodic updates and compensate drifting from the bias errors inherent when integrating INS heading for position, have been widely implemented in the navigation of AUVs. The work in this paper is an extension to the general integrated INS/GPS by fusing the VNS and INS data between intermittent GPS fixes. The navigation system that is being developed at the UoP is based on a multisensor data fusion (MSDF) technique that can produce accurate navigation information continuously in real time from a variety of low cost inertial sensors and a GPS receiver. During an actual mission this subsystem is enhanced by data from the intelligent viewing system developed by CU, with the purpose of aiding navigation by providing velocity estimates. Once the navigation data has been suitably processed it will be fed to the guidance and control system for the appropriate action. The aim of this paper is to describe

the present hardware/sensor configurations and techniques to combine measurement data from the VNS, INS and GPS to derive an estimated position of the *Hammerhead* AUV during both submerged and surface operations. The structure of this paper is as follows: the next two sections describe the current status of the *Hammerhead* VNS and INS/GPS development respectively and concluding remarks are given in section 4.

2. HAMMERHEAD VNS DEVELOPMENT: CURRENT STATUS

2.1 General description

The *Hammerhead* VNS is based on the laser stripe illumination (LSI) methodology previously developed at CU (Tetlow and Allwood, 1995), and will provide enhanced viewing of the seabed below the vehicle. However, it also provides real time data such as velocities, altitude and tracking information to the navigation system during the mission as well as gathering images to produce a post mission enhanced optical waterfall image of a surveyed area. There are several advantages of this type of approach over conventional imaging. LSI provides an improved image contrast at a given range. From computer simulations (Jaffe and Dunn, 1988) this type of system becomes limited at 5-7 attenuation lengths, compared with 2-4 for a conventionally illuminated system. This allows an increased deployment altitude for seabed surveys (3-18 metres) resulting in a greater swathe and hence greater area coverage. The images produced are approximately optically flat, meaning they exhibit even illumination. Furthermore, the structured nature of the light allows additional geometric information to be derived from the image and the stripe region can be extracted for each image to form a continuously evolving 2-D intensity waterfall image. However, LSI systems are more expensive and require a greater development resource than conventional systems. The viewing system comprises: a 100mW frequency doubled diode pumped Nd:YAG Laser (532nm), a low cost high sensitivity monochrome charged coupled device (CCD) camera with a wide angle lens, a single axis scanner and a tilt-compensated electronic compass (TCM2). Both the laser/scanner assembly and the camera are mounted within separate dedicated sections of the torpedo-shaped vehicle, with specially made plane ports to accommodate the optical path. The TCM2 compass uses two inclinometers to correct the output of three magnetometers for the declination angle error. The inclinometers, which are liquid-filled, are integral to the viewing system, providing tilt data for stabilisation when pitch and roll is experienced. The complete sensor subsystem can be split into the

three areas, with regard to their distinct utility: 1. velocity estimator; 2. active altitude sensor; 3. imaging capabilities.

A computer vision application is used to derive the required navigational and tracking information in real time from acquired images. An estimate of the instantaneous speed of the vehicle can be derived by using a 2-D correlation-based window-tracker, together with an integration of a range estimate from a laser-triangulation system by which image displacements are transformed into real-world displacements. The laser-triangulation technique requires the extraction of the vertical position of the laser stripe on an image to determine the range from the centre of the camera axis to the seabed. Together with the speed and range acquisition in real time, image quality can be checked and optimized by either changing the laser system parameters or demanding navigational changes from the vehicle. The complete video sequence is also recorded in digital-8 format and can be post-processed to produce a continuous 2-D intensity waterfall image or 3-D range images of the seabed with dimensional data and referenced against accurate positional information. These optical maps can be used to classify and locate particular objects that are of interest. This can be implemented in the image-processing suite or by manual inspection of the mosaic. In the autumn of 2002 a set of constrained motion trials were performed at the IFREMER facility in Brest. As well as the validation of a viewing model and system calibration, these trials created a useful archive of test files, allowing much of the future development to be possible from the dry laboratory. Furthermore, the measurements necessary to build a ground truth model were acquired in parallel to the image and vehicle specific data. This is used in ascertaining the accuracy of the vision-based navigation routines. A description of these experiments, including the construction, analysis and limitations of the ground truth model is given in a previous paper (Dalglish *et al.*, 2003). The next sub-section briefly describe the sensor in terms of one navigational components, i.e. the velocity estimator. Readers interested in the second component, the active altitude sensor, are referred to Loebis *et al.* (2003a). Some recent results are presented and the means by which the outputs are to be integrated with the MSDF algorithm (described later in the paper) is discussed. More detail concerning each component and the imaging capabilities will be given in subsequent papers.

2.2 Velocity estimator

As an implementation of window-based tracking (Anandan, 1989) using the LabVIEW IMAQ™

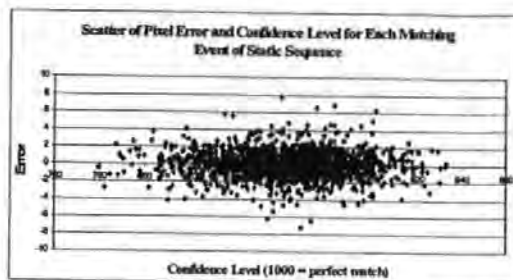


Fig. 1. Scatter plot of error vs. confidence level

Image Processing toolbox, this algorithm performs an intelligent correlation between an initial stored image region and subsequent image regions where image displacement is measured directly. The tracker uses a non-uniform sampling technique where only a few points that represent the overall content of the image are extracted. Moreover, an adaptive search strategy based on previous displacements is also incorporated to further improve tracking efficiency. Each successful matching event outputs a confidence level based on the degree of success of the correlation. Fig. 1 illustrates the distribution of matching events for the single resolution case and the corresponding pixel error, where sub-pixel accuracy is used. This was based on a static subsea sequence of two thousand 'lossy' JPEG images. The stand-off distance was eight metres. The window size was 20×20 pixels. As a more realistic alternative to adding Gaussian noise to each pixel independently, it is planned to use an 'artificial turbidity' environment in the Cranfield test tank to assess the static performance of the single resolution tracker under increasing noise. It is intended to use the confidence value as an indication of image quality, where as the confidence value degrades, the desired altitude of the AUV needs to be reduced. Outliers are detected in a smoothing stage.

3. HAMMERHEAD INS/GPS DEVELOPMENT: CURRENT STATUS

3.1 ESKF Modelling

Brown and Hwang (1997) discuss the advantages of the ESKF method over the total state Kalman filter. The most important advantage is that any nonlinear relationship between the process dynamics in the inertial system and the measurement relationships can be removed in a differencing operation, and the filter becomes linear. This linearity condition is required by the Kalman filter. This condition can also lead to a faster codes execution as linearisation operations are relatively slow to execute. KF algorithms are widely available in the literature. The interested reader can refer to Brown and Hwang (1997). Works on ESKF

$$\mathbf{F} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -\sin \hat{\psi}^g \cdot \hat{u}^h - \cos \hat{\psi}^g \cdot \hat{v}^h & 0 & \cos \psi_m^g & -\sin \psi_m^g \\ 0 & 0 & 0 & 0 & 0 & \cos \hat{\psi}^g \cdot \hat{u}^h - \sin \hat{\psi}^g \cdot \hat{v}^h & 0 & \sin \psi_m^g & \cos \psi_m^g \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{\cos \phi_m^h}{\cos \theta_m^h} & 0 & 0 \\ 0 & 0 & 0 & \frac{-1}{\tau_x} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{-1}{\tau_y} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{-1}{\tau_\psi} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{-1}{\tau_\psi} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-1}{\tau_u} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-1}{\tau_v} \end{bmatrix} \quad (3)$$

however, are very limited especially in the field of AUV navigation systems. An example of this work can be found in Gustaffson *et al.* (2001). Like in the KF, the ESCKF algorithm can be divided into two major parts: the measurement update and the time update. In ESCKF, the measurement update is obtained by subtracting the direct measurements from the computed version of the same quantities. By doing this, the true values cancel each other out and what remains is the difference between the measurement errors and drift errors. In the time update, the estimates are obtained by subtracting the estimated drift errors from the forward filter pass from the computed version of the same quantities. In this paper, measurement errors from an accelerometer and a gyroscope (assembled in an inertial measurement unit (IMU)), a TCM2 electronic compass and a GPS receiver unit are estimated and modelled using first order Markov processes which are defined in the following manner:

$$\dot{x} = -\frac{1}{\tau} \cdot x + \gamma \quad (4)$$

In (4), x is the error process to be modelled, τ is the time constant of the assumed Markov process and γ is white noise. For modelling purposes, all sensor data have been collected in static conditions for a period of approximately 2.5 hours. For the same purpose, three different frames of reference are defined. The body-fixed (b) frame of reference is aligned to the axes of the AUV, where forward-starboard-down correspond to $x - y - z$. These need to be transformed to the geographical (g) frame of reference, where $x - y - z$ correspond to North-East-Down. For these particular application, the measurements in question are 3D accelerations, as well as angular rates measured by the IMU. Earth-centred Earth-fixed (ECEF) frame is where the GPS latitude and longitude are defined. The elements of the state of the ESCKF are defined as follows:

$$\mathbf{x} = [x_d^g \ y_d^g \ \psi_d^g \ x_e^g \ y_e^g \ \psi_e^g \ r_e^b \ u_e^h \ v_e^h] \quad (5)$$

In (5) the subscripts d and e denote drift and sensor errors respectively. Superscripts g , b and h denote geographical frame, body-fixed-frame and horizontal frame respectively. Drift errors in position, x_d^g and y_d^g stem from the error in the integrated acceleration (u_e^h and v_e^h), and compass error ψ_e^g . ψ_d^g is heading drift error which comes from the error in the integrated yaw rate. Measurement errors in the position blend are respectively represented by the states x_e^g and y_e^g . Finally, state r_e^b represents gyroscope's yaw error. The process matrix (\mathbf{F}) is given in (3). In (3), m denotes measured value. Due to page limitation, the derivation of this matrix is not given in this paper.

The variance of the process noise for a Markov error model can be described as in Brown and Hwang (1997),

$$\text{variance}[w_k] = (1 - e^{-\frac{\Delta t}{\tau}}) \cdot \text{variance}[x_k] \quad (6)$$

In (6), $e^{-\frac{\Delta t}{\tau}}$ is the state transition parameter for the Markov error model. Δt is the discrete time interval and τ is the time constant. By taking the approximation: $e^{-\beta \Delta t} \approx 1 - \beta \cdot \Delta t$, where $\beta = \tau^{-1}$, the following is true:

$$\text{variance}[w_k] = (2\beta \Delta t - (\beta \Delta t)^2) \cdot \text{variance}[x_k] \quad (7)$$

According to the process model, the heading drift error state represents the integrated yaw error state, in effect an integrated Markov process. The process noise covariance matrix for these two states can be defined as in Brown and Hwang (1997),

$$\mathbf{Q} = \begin{bmatrix} E[\psi_d^g \psi_d^g] & E[\psi_d^g r_e^b] \\ E[\psi_d^g r_e^b] & E[r_e^b r_e^b] \end{bmatrix} \quad (8)$$

where

$$Q_{11} = \frac{2\sigma^2}{\beta} \left[\Delta t - \frac{2}{\beta}(1 - \phi) + \frac{1}{2\beta}(1 - \phi^2) \right] \quad (9)$$

$$Q_{12} = Q_{21} = 2\sigma^2 \left[\frac{1}{\beta}(1 - \phi) + \frac{1}{2\beta}(1 - \phi^2) \right] \quad (10)$$

$$Q_{22} = \sigma^2(1 - \phi^2) \quad (11)$$

where β is the inverse of the Markov time constant, σ^2 is the process noise variance of the yaw rate Markov error and $\phi = e^{-\frac{\Delta t}{\tau}}$ and defined as before.

For the first two states in the process model, x_d^g and y_d^g , the analysis is more complicated and for this reason, the noise covariance matrix of the two states are obtained from an empirical result, and provision for the adjustment method has been made and will be reported in the future.

The measurement matrix \mathbf{H} relates the available measurement updates to the element in the state vector and takes the following form:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \end{bmatrix} \quad (12)$$

The measurement noise covariance matrix \mathbf{R}_k is determined empirically and given as:

$$\mathbf{R}_k = \begin{bmatrix} \sigma_{X-Position}^2 & 0 & 0 \\ 0 & \sigma_{Y-Position}^2 & 0 \\ 0 & 0 & \sigma_{Heading}^2 \end{bmatrix} \quad (13)$$

where $\sigma_{X-Position}^2$, $\sigma_{Y-Position}^2$, $\sigma_{Heading}^2$ are the variance in X,Y direction and heading respectively. These values will be adapted using the algorithm discussed in the next sub-section.

3.2 Fuzzy error state complementary Kalman filter (FESCKF)

In this sub-section, an on-line innovation-based adaptive scheme of the ESCKF to adjust the \mathbf{R} matrix employing the principles of fuzzy logic is presented. The fuzzy logic is chosen mainly because of its simplicity. This motivates the interest in the topic, as testified by related papers which have been appearing in the literature (Loebis *et al.*, 2003b; Escamilla-Ambrosio and Mort, 2001). The FESCKF proposed herein is based on the IAE approach using a technique known as covariance-matching (Mehra, 1970). The basic idea behind the technique is to make the actual value of the covariance of the innovation sequences match its theoretical value.

The actual covariance is defined as an approximation of the \mathbf{Inn}_k sample covariance through

averaging inside a moving estimation window of size M which takes the following form:

$$\hat{\mathbf{C}}_{Inn_k} = \frac{1}{M} \sum_{j=j_0}^k \mathbf{Inn}_k \cdot \mathbf{Inn}_k^T \quad (14)$$

where $j_0 = k - M + 1$ is the first sample inside the estimation window. An empirical experiment is conducted to choose the window size M . From experimentation it was found that a good size for the moving window in (14) is 15. The theoretical covariance of the innovation sequence is defined as

$$\mathbf{S}_k = \mathbf{H}_k \cdot \mathbf{P}_k^- \cdot \mathbf{H}_k^T + \mathbf{R}_k \quad (15)$$

The logic of the adaptation algorithm using covariance matching technique can be qualitatively described as follows. If the actual covariance value $\hat{\mathbf{C}}_{Inn_k}$ is observed, whose value is within the range predicted by theory \mathbf{S}_k and the difference is very near to zero, this indicates that both covariances match almost perfectly and only a small change is needed to be made on the value of \mathbf{R} . If the actual covariance is greater than its theoretical value, the value of \mathbf{R} should be decreased. On the contrary, if $\hat{\mathbf{C}}_{Inn_k}$ is less than \mathbf{S}_k , the value of \mathbf{R} should be increased. This adjustment mechanism lends itself very well to being dealt with using a fuzzy-logic approach based on rules of the kind:

$$IF \langle antecedent \rangle THEN \langle consequent \rangle \quad (16)$$

where antecedent and consequent are of the form $\nu \in O_i$, $\kappa \in L_i$, $i = 1, 2, \dots$ respectively. Where ν and κ are the input and output variables, respectively, and O_i and L_i are the fuzzy sets.

To implement the above covariance matching technique using the fuzzy logic approach, a new variable called delta_k , is defined to detect the discrepancy between $\hat{\mathbf{C}}_{Inn_k}$ and \mathbf{S}_k . The following fuzzy rules of the kind (16) are used:

$$IF \langle \text{delta}_k \cong 0 \rangle THEN \langle \mathbf{R}_k \text{ is unchanged} \rangle \quad (17)$$

$$IF \langle \text{delta}_k > 0 \rangle THEN \langle \mathbf{R}_k \text{ is decreased} \rangle \quad (18)$$

$$IF \langle \text{delta}_k < 0 \rangle THEN \langle \mathbf{R}_k \text{ is increased} \rangle \quad (19)$$

Thus \mathbf{R} is adjusted according to,

$$\mathbf{R}_k = \mathbf{R}_{k-1} + \Delta \mathbf{R}_k \quad (20)$$

where $\Delta \mathbf{R}_k$ is added or subtracted from \mathbf{R} at each instant of time. Here delta_k is the input to the fuzzy inference system (FIS) and $\Delta \mathbf{R}_k$ is the output.

3.3 Fuzzy membership functions optimisation

MOGA is used here to optimize the membership functions of the FKF. To translate the FKF

membership functions to a representation useful as genetic material, they are parameterised with real-valued variables. Each of these variables constitutes a gene of the chromosomes for the MOGA. Boundaries of chromosomes are required for the creation of chromosomes in the right limits so that the MOGA is not misled to some other area of search space. The technique adopted in this paper is to define the boundaries of the output membership functions according to the furthest points and the crossover points of two adjacent membership functions. In other words, the boundaries of FKF consist of three real-valued chromosomes (*Chs*), as in Figure 2. The trapezoidal membership

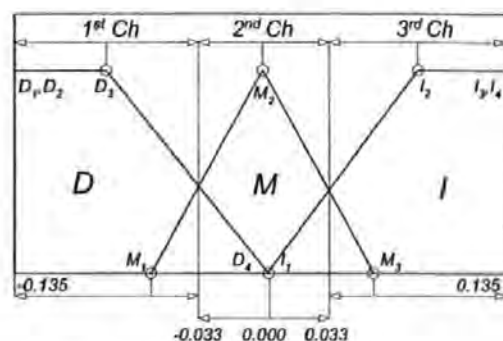


Fig. 2. Membership function and boundaries of R_k

functions' two furthest points, -0.135 (D_1), -0.135 (D_2) and 0.135 (I_3), 0.135 (I_4) of FKF, remain the same in the GA's description to allow a similar representation as the fuzzy system's definition. As can be seen from Figure 2, D_3 and M_1 can change value in the 1st *Ch* boundary, D_4 , M_2 and I_1 in the 2nd *Ch* boundary, and finally, M_3 and I_2 in 3rd *Ch*.

4. CONCLUDING REMARKS

A novel method to obtain an accurate AUV navigation system is proposed. The method is based on the ESCKF coupled with fuzzy logic to adjust the value of measurement noise covariance matrix R during both surface and underwater mission whereby different combination of sensors, i.e. INS/GPS and INS/VNS respectively, are used. MOGA is proposed to further refine the result.

REFERENCES

Anandan, P. (1989). A Computational Framework and an Algorithm for the Measurement of Visual Motion. *International Journal of Computer Vision* 2, 283-310.

Balasuriya, B. A. A. P. and T. Ura (1999). Sensor Fusion Technique for Cable Following by Autonomous Underwater Vehicles. In: *Proceedings of the 1999 IEEE International Conference on Control Applications*. Kohala Coast-Island of Hawaii, Hawaii, USA. pp. 1779-1784.

Brown, R. G. and P. Y. C. Hwang (1997). *Introduction to Random Signals and Applied Kalman Filtering*. 3rd Ed. John Wiley and Sons.

Dalgleish, F. R., S. Tetlow and R. L. Allwood (2003). Preliminary Experiments in the Development of a Laser-based Imaging Sensor for AUV Navigation. In: *Proceedings of the 1st IFAC Workshop on Guidance and Control of Underwater Vehicles*. Newport, South Wales, UK. pp. 239-244.

Escamilla-Ambrosio, P. J. and N. Mort (2001). A Hybrid Kalman Filter-Fuzzy Logic Multisensor Data Fusion Architecture with Fault Tolerant Characteristics. In: *Proc. of the 2001 International Conference on Artificial Intelligence*. Las Vegas, NV, USA. pp. 361-367.

Fleischer, S. D., M. R. Rock and R. Burton (1997). Global Position Determination and Vehicle Path Estimation from a Vision Sensor for Real Time Video Mosaicking and Navigation. In: *Proceedings of the Oceans '97 MTS/IEEE*. Halifax, Nova Scotia, Canada. pp. 641-647.

Gustafsson, E., E. An and S. Smith (2001). A Postprocessing Kalman Smoother for Underwater Vehicle Navigation. In: *Proc. 12th International Symposium on Unmanned Un-thetered Submersible Technology*. New Hampshire, NH, USA. pp. 1-7.

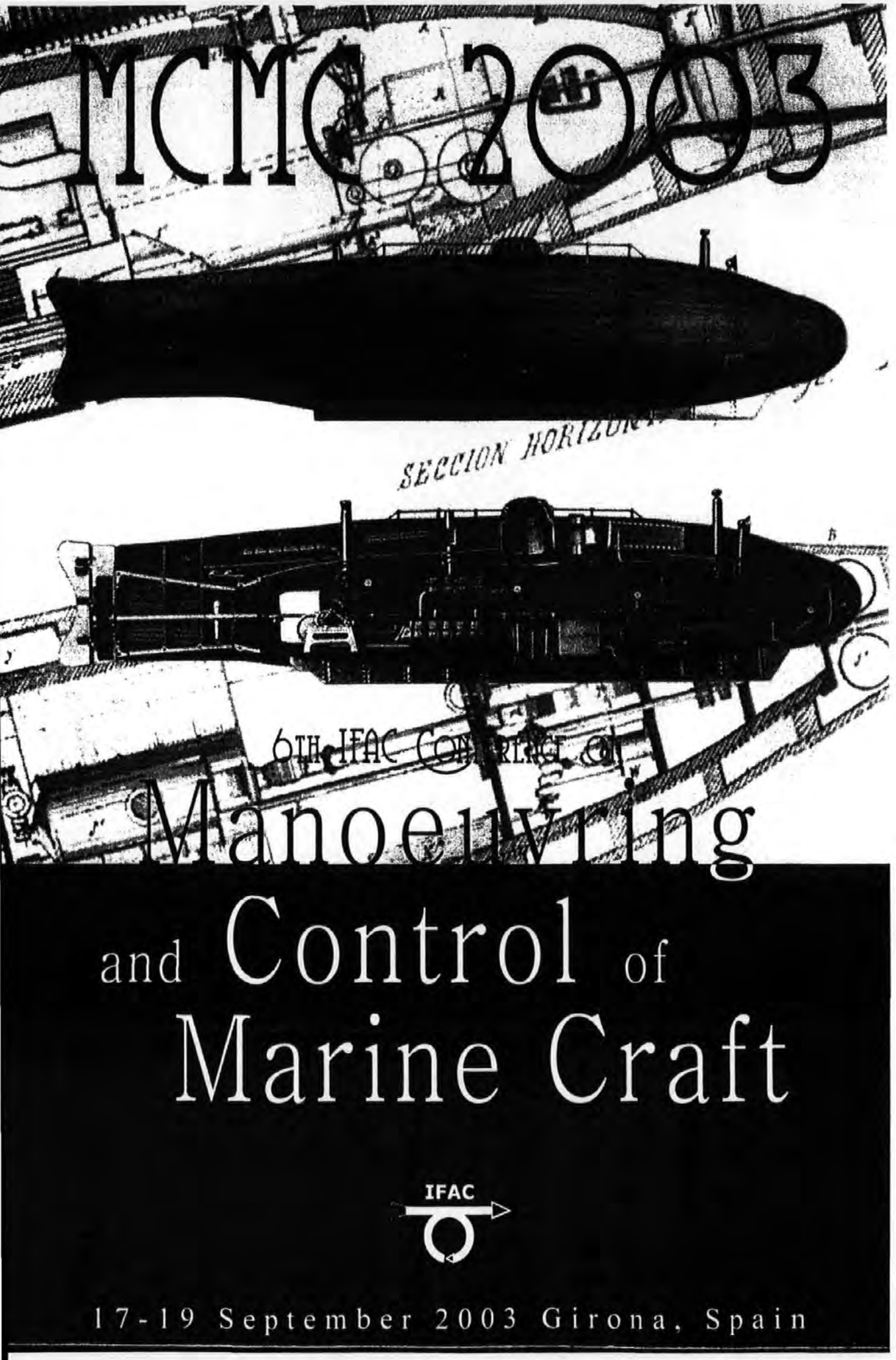
Jaffe, J. and C. Dunn (1988). A Model-Based Comparison of Underwater Imaging Systems. *SPIE Ocean Optics IX* 925, 344-350.

Loebis, D., F. R. Dalgleish, R. Sutton, S. Tetlow, J. Chudley and R. L. Alwood (2003a). An Integrated Approach in the Design of a Navigation System for an AUV. In: *Proc. 6th IFAC Conference on Manoeuvring and Control of Marine Craft*. Girona, Spain. pp. 329-334.

Loebis, D., R. Sutton and J. Chudley (2003b). A Fuzzy Kalman Filter for Accurate Navigation of an Autonomous Underwater Vehicle. In: *Proc. 1st IFAC Workshop on Guidance and Control of Underwater Vehicles*. Newport, South Wales, UK. pp. 161-166.

Mehra, R. K. (1970). On the Identification of Variances and Adaptive Kalman Filtering. *IEEE Transactions on Automatic Control* AC-16(1), 12-21.

Tetlow, S. and R. L. Allwood (1995). Development and Applications of a Novel Underwater Laser Illumination System. *Underwater Technology* 21, 13-20.



MCMC 2003

SECCION HORIZONTAL

6TH IFAC CONFERENCE

Manoeuvring and Control of Marine Craft



17-19 September 2003 Girona, Spain

A FUZZY KALMAN FILTER OPTIMIZED USING A GENETIC ALGORITHM FOR ACCURATE NAVIGATION OF AN AUTONOMOUS UNDERWATER VEHICLE

D. Loebis, J. Chudley and R. Sutton

*Marine and Industrial Dynamic Analysis Research Group
Department of Mechanical and Marine Engineering
The University of Plymouth
Drake Circus, Plymouth, PL4 8AA, United Kingdom*

Abstract: The Kalman filter (KF) has been a popular method for integrating data produced by inertial navigation system (INS) sensor suites and global positioning systems (GPSs) onboard autonomous underwater vehicles to provide optimal estimates of their position and attitude. In this paper the data from a variety of INS sensors are first fused together by means of a linear Kalman filter (LKF) before being integrated with GPS data through the use of an extended Kalman filter (EKF). The use of a fuzzy-rule-based adaptation scheme to cope with the divergence problem caused by the insufficiently known *a priori* filter statistics is also explored. The choice of fuzzy membership functions for the adaptation scheme is first carried out using a heuristic approach. Genetic algorithm (GA) techniques are then used to optimise the parameters of the membership functions with respect to a certain performance criteria in order to improve the overall accuracy of the integrated navigation system. Copyright © 2003 IFAC

Keywords: Autonomous underwater vehicles, navigation, sensor fusion, Kalman filters, extended Kalman filters, fuzzy logic, genetic algorithm

1. INTRODUCTION

In the past few decades, there have been numerous worldwide research and development activities in order to explore the oceans of the world. As an ocean is an inherently hostile and hazardous environment, the need for an underwater robotic system, especially one with high reliability and a fully built-in intelligence, becomes apparent. The autonomous underwater vehicle (AUV) class of vessel meets these requirements.

To achieve truly autonomous behaviour, an AUV must be able to locate itself accurately during an operating scenario using only its onboard sensors. In the past, fusing of inertial navigation system (INS) sensors and the integration with a global positioning

system (GPS) through the use of a conventional linear Kalman filter (LKF) (appendix A) and an extended Kalman filter (EKF) has been a popular method of localisation of an AUV. However, a significant difficulty in designing a KF (refers to both LKF and EKF) can often be traced to incomplete *a priori* knowledge of the process covariance matrix (Q) and measurement noise covariance matrix (R). In most practical applications, these matrices are initially estimated or even unknown. The problem here is that the optimality of the estimation algorithm in the KF setting is closely connected to the quality of *a priori* information about the process and measurement noise (Mehra, 1970). It has been shown that insufficiently known *a priori* filter statistics can on the one hand reduce the precision of the estimated filter states or introduces biases to their estimates. In

addition, incorrect *a priori* information can lead to practical divergence of the filter (Fitzgerald, 1971). From the aforementioned it may be argued that the conventional KF with fixed R and/or Q should be replaced by an adaptive estimation formulation.

In this paper, an innovation adaptive estimation (IAE) approach (Mehra, 1970) coupled with fuzzy logic techniques is used to adjust the R matrix of the KF. Here the innovation Inn_k at sample time k in the KF algorithm is the difference between the real measurement z_k , received by the filter and its estimated (predicted) value \hat{z}_k , and is computed as follows:

$$Inn_k = z_k - \hat{z}_k \quad (1)$$

The predicted measurement is the projection of the filter predicted states \hat{x}_k^- onto the measurement space through the measurement design matrix H_k , i.e.

$$\hat{z}_k = H_k \hat{x}_k^- \quad (2)$$

Innovation represents additional information available to the filter as a result of the new measurement z_k . The occurrence of data with statistics different from the *a priori* information will first show up in the innovation vector. For this reason the innovation sequence represent the information content in the new observation and is considered the most relevant source of information to the filter adaptation.

The fuzzy logic membership functions for the IAE approach are established by a combination of knowledge, experience and observation and may thus not be optimal. Additionally, fine-tuning of its performance is still a matter of trial and error. Many studies have shown that genetic algorithms (GAs) have the ability to find fuzzy membership functions closer to optimal solutions and may be made to implement self-tuning and adaptive schemes (Cordón *et al.*, 1997). However, this paper is the first known use of the GA for the optimisation of the membership function of a fuzzy system in the noise adaptation of a KF.

2. THE ADAPTIVE ESTIMATION ALGORITHM

2.1 Fuzzy Kalman filter.

In this sub-section, an on-line innovation-based adaptive scheme of the KF to adjust the R matrix employing the principles of fuzzy logic is presented. The fuzzy logic is chosen mainly because of its simplicity. This motivates the interest in the topic, as testified by related articles which have been appearing in the literature (Kobayashi *et al.*, 1998).

The fuzzy logic Kalman filter (FKF) proposed in this paper is based on the IAE approach using a

technique known as covariance-matching (Mehra, 1970). The basic idea behind the technique is to make the actual value of the covariance of the innovation sequences match its theoretical value.

The actual covariance is defined as an approximation of the Inn_k sample covariance through averaging inside a moving estimation window of size N (Mohamed and Schwarz, 1999) which takes the following form:

$$\hat{C}_{r_k} = \frac{1}{M} \sum_{i=i_0}^N Inn_k Inn_k^T, \quad (3)$$

where $i_0 = k - M + 1$ is the first sample inside the estimation window. An empirical experiment is conducted to choose the window size M . From experimentation it was found that a good size for the moving window in (3) is 15.

The theoretical covariance of the innovation sequence is defined as

$$S_k = H_k P_k^- H_k^T + R_k, \quad (4)$$

The logic of the adaptation algorithm using covariance matching technique can be qualitatively described as follows. If the actual covariance value \hat{C}_{r_k} is observed, whose value is within the range predicted by theory S_k and the difference is very near to zero, this indicates that both covariances match almost perfectly and only a small change is needed to be made on the value of R . If the actual covariance is greater than its theoretical value, the value of R should be decreased. On the contrary, if \hat{C}_{r_k} is less than S_k , the value of R should be increased. This adjustment mechanism lends itself very well to being dealt with using a fuzzy-logic approach based on rules of the kind:

$$\text{IF } \langle \text{antecedent} \rangle \text{ THEN } \langle \text{consequent} \rangle, \quad (5)$$

where antecedent and consequent are of the form $N \in M_i, \mathcal{T} \in N_i, i = 1, 2, \dots$ respectively, where N and \mathcal{T} are the input and output variables, respectively, and M_i and N_i are the fuzzy sets.

To implement the above covariance matching technique using the fuzzy logic approach, a new variable called *delta*, is defined to detect the discrepancy between \hat{C}_{r_k} and S_k . The following three fuzzy rules of the kind (5) are used:

$$\text{IF } \langle \text{delta}_k \equiv 0 \rangle \text{ THEN } \langle R_k \text{ is unchanged} \rangle, \quad (6)$$

$$\text{IF } \langle \text{delta}_k > 0 \rangle \text{ THEN } \langle R_k \text{ is decreased} \rangle, \quad (7)$$

$$\text{IF } \langle \text{delta}_k < 0 \rangle \text{ THEN } \langle R_k \text{ is increased} \rangle \quad (8)$$

Thus R is adjusted according to,

$$R_k = R_{k-1} + \Delta R_k, \quad (9)$$

where ΔR_k is added or subtracted from R at each instant of time. Here $\delta\Delta R_k$ is the input to the fuzzy inference system (FIS) and ΔR_k is the output.

On the basis of the above adaptation hypothesis, the FIS can be implemented using three fuzzy sets for $\delta\Delta R_k$; N = Negative, Z = Zero and P = Positive. For ΔR_k the fuzzy sets are specified as I = Increase, M = Maintain and D = Decrease. The membership functions of these fuzzy sets which are first designed using a heuristic approach are shown in Fig. 1.

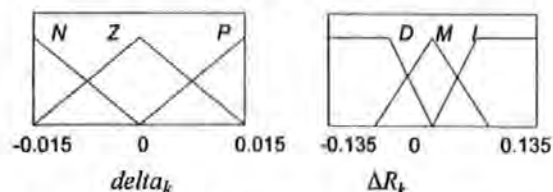


Fig. 1. Membership function of $\delta\Delta R_k$ and ΔR_k

2.2 Fuzzy logic observer

To monitor the performance of a FKF, another FIS called the fuzzy logic observer (FLO) (Escamilla-Ambrosio and Mort, 2001) is used. The FLO assigns a weight or degree of confidence denoted as c_k , a number on the interval $[0,1]$, to the FKF state estimate. The FLO is implemented using two inputs: the values of $|\delta\Delta R_k|$ and R_k . The membership functions of these variables are shown in Fig. 2.

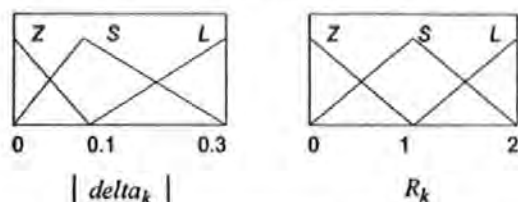


Fig. 2. Membership function of $|\delta\Delta R_k|$ and R_k

The fuzzy labels for the membership functions: Z = Zero, S = Small and L = Large. Three fuzzy singletons are defined for the output c_k and are labelled as G = Good, AV = Average and P = Poor with values 1, 0.5 and 0 respectively. The basic heuristic hypothesis for the FLO is as follows: if the value of $|\delta\Delta R_k|$ is near to zero and the value of R_k is near to zero, then the FKF works almost perfectly and the state estimate of the FKF is assigned a weight near 1. On the contrary if one or both of these values increases far from zero, it means that the FKF performance is degrading and the FLO assigns a weight near 0. Table 1 gives the complete fuzzy rule base of each FLO.

2.3 Fuzzy membership functions optimisation

GAs as function optimisers are global optimisation techniques based on natural selection (Goldberg, 1989). GAs are here presented as a tool to optimise

Table 1 Fuzzy rule base FLO

R_k	Z	S	L
$ \delta\Delta R_k $			
Z	G	G	AV
S	G	AV	P
L	AV	P	P

the FKF membership functions with respect to a certain objective function. Although there are many possible variants of basic GA, the fundamental underlying mechanism operates on a population of individuals, and consists of three operations: (1) evaluation of individual fitness in the population, (2) selection of the fittest individuals, and (3) recombination and mutation. The initial population $P(0)$ is chosen randomly and the individuals resulting from these three operations form the next generation's population. The process is iterated until the system ceases to improve or a termination condition is satisfied. Fig. 3. shows the structure of a simple GA.

```

begin (I)
  t = 0;
  initialise P (t)
  evaluate P (t)
  while ~(termination-condition) do
    begin (2)
      select P(t)
      recombine P(t)
      mutate P (t)
      evaluate P (t)
      t = t + 1;
    end (2)
  end (I)

```

Fig. 3. Structure of a GA

To translate the FKF membership functions to a representation useful as genetic material, they are parameterised with real-valued variables. Each of these variables constitutes a gene of the chromosomes for the GA. Boundaries of chromosomes are required for the creation of chromosomes in the right limits so that the GA is not misled to some other area of search space. The technique adapted in this paper is to define the boundaries of the output membership functions according to the furthest points and the crossover points of two adjacent membership functions. In other words, the boundaries of FKF consist of three real-valued chromosomes (Chs), as in Fig. 4.

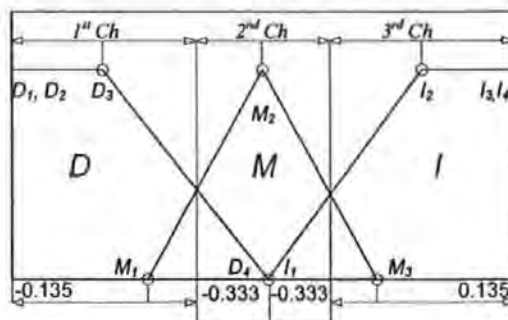


Fig. 4. Membership function and boundaries of R_k

The trapezoidal membership functions' two furthest points, -0.135 (D_1), -0.135 (D_2) and 0.135 (I_3), 0.135 (I_4) of FKF, remain the same in the GA's description to allow a similar representation as the fuzzy system's definition. As can be seen from Fig. 4., D_3 and M_1 can change value in the 1st Ch boundary, D_4 , M_2 and I_1 in the 2nd Ch boundary, and finally, M_3 and I_2 in 3rd Ch. Table 2 shows the encoding used for optimisation of the membership functions.

Table 2 FKF boundaries

Limit	Parameter		
	D_3, M_1	D_4, M_2, I_1	M_3, I_2
Upper Limit	-0.135	-0.033	0.033
Lower Limit	-0.033	0.033	0.135

3. SENSOR FUSION OF INS SENSOR DATA

In this section, the FKF algorithm is applied to a linearised steering model of an AUV at forward speed 1.3 ms^{-1} (Lea, 1998) as following:

$$\begin{bmatrix} \dot{v}(t) \\ \dot{r}(t) \\ \dot{\psi}(t) \end{bmatrix} = \begin{bmatrix} -2.09 & 0.376 & 0 \\ -7.96 & -8.69 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v(t) \\ r(t) \\ \psi(t) \end{bmatrix} + \begin{bmatrix} 1.07 \\ -14.1 \\ 0 \end{bmatrix} \delta_r(t) + w(t), \quad (10)$$

$$z_k = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ r(t) \\ \psi(t) \end{bmatrix} + v(t), \quad (11)$$

where $v(t)$, $r(t)$ and $\psi(t)$ represent the sway velocity, yaw rate of turn and yaw angle. The $w(t)$ and $v(t)$ are both zero mean white noise for the system and measurement models respectively and $\delta_r(t)$ the rudder deflection. A sample time of 0.125s is used to discretised the linearised model. The initial conditions are $[\hat{v}_0 \ \hat{r}_0 \ \hat{\psi}_0]^T = [0 \ 0 \ 0]^T$, $P_0 = 0.01 \ I_3$ and Q is made constant as $0.01 \ I_3$. A sinusoidal input was applied to the rudder. Four yaw sensors with different noise characteristics are considered to measure the response of the vehicle.

The actual value of R for each sensor is assumed unknown but its initial value is selected as 1. The FKF algorithm optimized using GA with parameters shown in Table 3 was then implemented and simulation results are shown in the next section.

Table 3 GA Parameters

Parameters	Values
Number of Generation	15
Number of Individual per Generation	10
Generation Gap in Selection Operation	0.95
Rate in Recombination Operation	0.8
Rate in Mutation Operation	0.09

3.1 Simulation Result

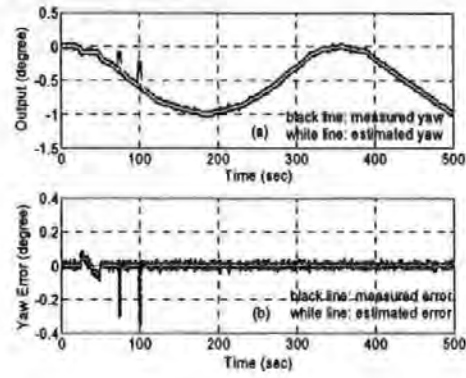


Fig. 5. (a) Measured and estimated yaw output, (b) measured and estimated yaw error of sensor 1.

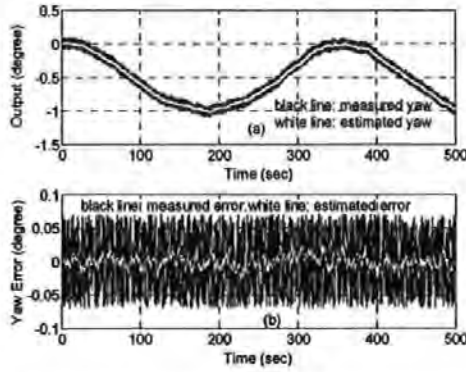


Fig. 6. (a) Measured and estimated yaw output, (b) measured and estimated yaw error of sensor 2.

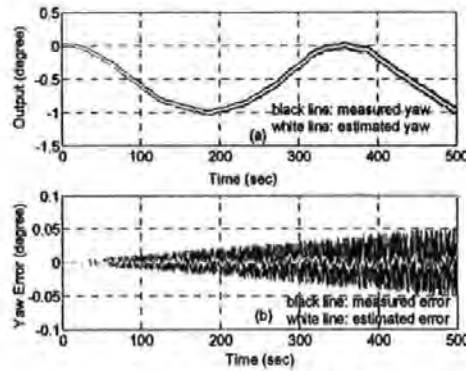


Fig. 7. (a) Measured and estimated yaw output, (b) measured and estimated yaw error of sensor 3.

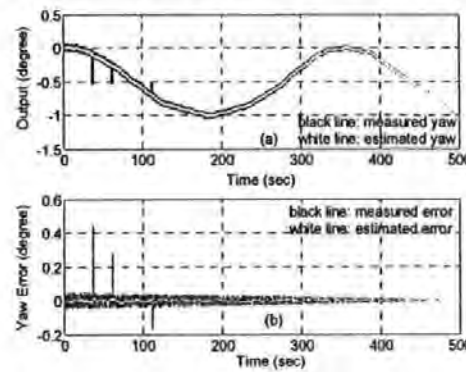


Fig. 8. (a) Measured and estimated yaw output, (b) measured and estimated yaw error of sensor 4.

Fig. 5, and Fig. 6. are the simulation result showing the response of the AUV observed by sensors with constant Gaussian noise, while Fig. 7. and Fig. 8. by sensors with uniform noise increasing and decreasing with time respectively.

To fuse the estimated yaw, a centre of gravity method is used,

$$\hat{z}_k = \frac{\sum_{i=1}^n \hat{z}_{k_i} c_{k_i}}{\sum_{i=1}^n c_{k_i}} \quad (12)$$

where \hat{z}_{k_i} is the output of the i -th FKF ($i=1,2,3,4$) and c_{k_i} is the respective weight at instant time k , which is shown in Fig. 9. Fig. 10. shows the comparison of the actual and the fused estimated yaw.

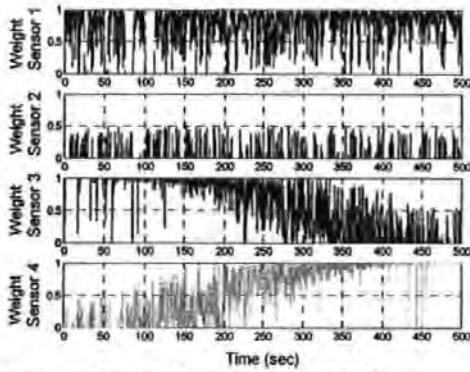


Fig. 9. Weight of sensor 1, 2, 3 and 4.

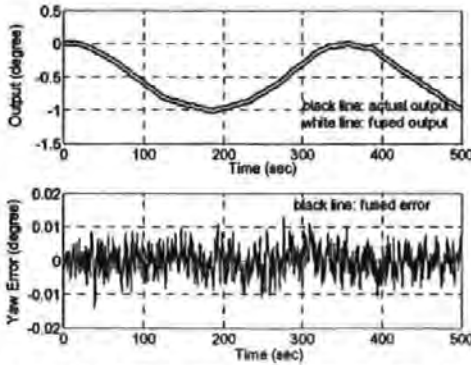


Fig. 10. (a) Actual and fused yaw output, (b) fused error.

Finally, the following performance measure are adopted for comparison purposes,

$$J_n = \sqrt{\frac{1}{n} \sum_{k=1}^n (z_{k_i} - z)^2}, \quad (13)$$

$$J_n = \sqrt{\frac{1}{n} \sum_{k=1}^n (z_{k_i} - \hat{z}_k)^2}, \quad (14)$$

where z_{k_i} is the actual value of the yaw, z_k is the measured yaw, \hat{z}_k is the estimated yaw at an instant of time k and n = number of samples. Table 4 shows the comparison of performance of each individual

FKF and those obtained by the proposed INS sensor fusion method with and without fuzzy system optimized using GA techniques.

Table 4 Comparison of performance

Sensor	Performance (degree)				
	J_n	Non-GA	J_n GA		
			1 st	2 nd	3 rd
Sensor 1	1.5291	1.0942	1.0758	1.0849	1.0809
Sensor 2	2.2908	0.5018	0.5028	0.5025	0.5026
Sensor 3	0.9933	0.2833	0.2856	0.2844	0.2856
Sensor 4	1.1996	0.2905	0.2901	0.2903	0.2903
Fused		0.2005	0.1940	0.1940	0.1940

4. GPS/INS NAVIGATION

In this section, the fused estimated yaw obtained previously is treated as a single imaginary yaw sensor and used by other INS sensors to transform data from body co-ordinate to Earth co-ordinate frame where integration with GPS data is performed.

A continuous time model of the vehicle motion appropriate to this problem is taken to be

$$\dot{X}(t) = F(X(t)) + W(t) \quad (15)$$

$$Z(t) = H(X(t)) + V(t) \quad (16)$$

Denoted by $X(t) = [\lambda(t) \ \varphi(t) \ \psi(t) \ \dot{\psi}(t) \ u(t) \ v(t)]^T$ is the model states. $\lambda(t)$ and $\varphi(t)$ are the longitude and latitude of the AUV position in Earth co-ordinate frame which are obtained from a GPS receiver, $\psi(t)$ is the yaw angle obtained from the imaginary yaw sensor, $\dot{\psi}(t)$ is yaw rate, $u(t)$ and $v(t)$ are the surge and sway velocity respectively.

In this system model, F and H are both continuous function, continuously differentiable in $X(t)$. The $W(t)$ and $V(t)$ are both zero mean white noise for the system and measurement models respectively.

The model states are related through the following kinematically based set of functions ($F(X(t))$ in Eq. 16):

$$\dot{\lambda}(t) = 0, \quad (17)$$

$$\dot{\varphi}(t) = 0, \quad (18)$$

$$\dot{\psi}(t) = r(t), \quad (19)$$

$$\dot{r}(t) = 0, \quad (20)$$

$$\dot{\lambda}(t) = u(t) \cos \psi(t) - v(t) \sin \psi(t), \quad (21)$$

$$\dot{\varphi}(t) = u(t) \sin \psi(t) + v(t) \cos \psi(t), \quad (22)$$

The output measurements are related through the states by the identity matrix $H(X(t))$. To obtain an EKF with an effective state prediction equation in a simple form, the continuous time model of (17)-(22) have been linearised about the current state estimates, producing the following form:

$$\dot{X}(t) = AX(t) + W(t), \quad (23)$$

$$Z(t) = CX(t) + V(t), \quad (24)$$

where $A =$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -u(t)\sin\psi(t) - v(t)\cos\psi(t) & 0 & \cos\psi(t) & -\sin\psi(t) \\ 0 & 0 & u(t)\cos\psi(t) - v(t)\sin\psi(t) & 0 & \sin\psi(t) & \cos\psi(t) \end{bmatrix}$$

and C is an identity matrix. Subsequent discretisation with period $T = 0.125$ s of the linearised model results in an EKF algorithm similar to the LKF algorithms in appendix A, only this time the A matrix is updated at every iteration. The initial conditions are $P_0 = 0.01 I_6$ and Q is made constant as $\text{diag}[0 \ 0 \ 0.000001 \ 0.01 \ 0.01 \ 0.01]$. The actual value of R is assumed unknown but its initial value is selected as $\text{diag}[100 \ 0 \ 0 \ 0.1 \ 0.1]$.

The FKF algorithm from section 2.1 is then implemented, only this time the adaptation of the (i, i) -th element of R_k is made in accordance with the (i, i) -th element of ΔR_k . Here a single-input-single-output (SISO) FIS as shown in Fig. 1., is used sequentially to generate the correction factors for the elements in the main diagonal of R_k as the following,

$$R_k(i, i) = R_{k-1}(i, i) + \Delta R_k, \quad (25)$$

With the proposed FKF adaptation, the RMSE for longitude is 2.2890 m and 2.1137 m for latitude compared to 2.5322 m for longitude and 2.1921 m for latitude without adaptation.

5. SUMMARY AND CONCLUSION

The problem with incomplete *a priori* knowledge of Q (process covariance matrix) and R (measurement covariance matrix) is considered. In this paper, an adaptive Kalman filter approach, based on the filter innovation sequence coupled with fuzzy logic optimized using genetic algorithm is discussed as an alternative for fusing INS sensor data and integrating INS/GPS position information. Implementation of this approach to a linearised steering model of an AUV, whose responses are measured with sensors with different noise characteristics, has shown a promising result in improving the estimation of the individual KF. The use of FLO also plays an important role in determining the weight or degree of confidence of the FKF output.

APPENDIX A: Kalman filter Equations

Given a discrete-time controlled process described by the linear stochastic difference equations:

$$x_{k+1} = A_k x_k + B_k u_k + w_k \quad (A1)$$

$$z_k = H_k x_k + v_k \quad (A2)$$

where x_k is an $n \times 1$ system state vector, A_k is an $n \times n$ transition matrix, u_k is an $l \times 1$ vector of the input forcing function, B_k is an $n \times l$ matrix, w_k is an $n \times 1$ process noise vector, z_k is a $m \times 1$ measurement vector, H_k is a $m \times n$ measurement matrix and v_k is a $m \times 1$ measurement noise vector. Both the w_k and v_k are assumed to be uncorrelated zero-mean Gaussian white noise sequences with covariance given by

$$E[w_k w_k^T] = \begin{cases} Q_k, & i = k \\ 0, & i \neq k \end{cases} \quad (A3)$$

$$E[v_k v_k^T] = \begin{cases} R_k, & i = k \\ 0, & i \neq k \end{cases} \quad (A4)$$

$$E[w_k v_i^T] = 0, \text{ for all } k \text{ and } i \quad (A5)$$

The KF algorithm can be organised into time update and measurement update equations,

Time update equations:

$$\hat{x}_{k+1}^- = A_k \hat{x}_k + B_k u_k \quad (A6)$$

$$P_{k+1}^- = A_k P_k A_k^T + Q_k \quad (A7)$$

Measurement update equations:

$$K_k = P_k^- H_k^T [H_k P_k^- H_k^T + R_k]^{-1} \quad (A8)$$

$$\hat{x}_k = \hat{x}_k^- + K_k [z_k - H_k \hat{x}_k^-] \quad (A9)$$

$$P_k = [I - K_k H_k] P_k^- \quad (A10)$$

The measurement update equations incorporate a new observation into the *a priori* estimate from the time update equations to obtain an improved *a posteriori* estimate. In the time and measurement update equations, \hat{x}_k is an estimate of the system state vector x_k , and P_k is the covariance matrix of the state estimation error.

REFERENCES

- Cordón, O., F. Herrera and M. Lozano (1997). On the Combination of Fuzzy Logic and Evolutionary Computation: A Short Review and Bibliography. In: *Fuzzy Evolutionary Computation*, (W. Pedrycz, (Ed)), pp. 57-77. Kluwer Academic.
- Escamilla-Ambrosio, P.J. and N. Mort. (2001). A Hybrid Kalman Filter-Fuzzy Logic Multisensor Data Fusion Architecture with Fault Tolerant Characteristics. *Proc. of 2001 International Conference on Artificial Intelligence (IC-AI'2001)*, pp. 361-367, Las Vegas, NV, USA.
- Fitzgerald, R.J. (1971). Divergence of the Kalman Filter. *IEEE Transaction on Automatic Control*, vol. AC-16, no. 6, pp. 736-747.
- Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, Reading, Massachusetts.
- Kobayashi, K., K.C. Cheok, K. Watanabe and F. Muneke. (1998). Accurate Differential Global Positioning System via Fuzzy Logic Kalman Filter Sensor Fusion Technique. *IEEE Transactions on Industrial Electronics*, vol. 45, no. 3, pp. 510-518.
- Lea, R. K. (1998). Control of a Tethered Underwater Flight Vehicle. *PhD Thesis*, University of Southampton.
- Mehra, R.K. (1970). On the Identification of Variances and Adaptive Kalman Filtering. *IEEE Transactions on Automatic Control*, vol. AC-15, no. 2, pp.175-184.
- Mohamed, A.H. and K.P. Schwarz. (1999). Adaptive Kalman Filtering for INS/GPS. *Journal of Geodesy*, vol. 73, pp. 193-203.

AN INTEGRATED APPROACH IN THE DESIGN OF A NAVIGATION SYSTEM FOR AN AUV

D. Loebis[†], F. R. Dalglish*, R. Sutton[†], S. Tetlow*, J. Chudley[†], and R. L. Allwood*

[†]*Marine and Industrial Dynamic Analysis Research Group
Department of Mechanical and Marine Engineering
The University of Plymouth
Drake Circus, Plymouth, PL4 8AA, United Kingdom*

**Offshore Technology Centre
Cranfield University
Bedfordshire, Cranfield, MK43 0AL, United Kingdom*

Abstract: An intelligent navigation system for an autonomous underwater vehicle known as *Hammerhead* is being developed by the University of Plymouth and Cranfield University. The *Hammerhead* navigation system is an integrated low-cost vision navigation subsystem and an inertial navigation system/global positioning subsystem. It is being developed to demonstrate the feasibility of using an integrated visual navigation system and inertial navigation system to navigate between intermittent global positioning system fixes whilst providing an enhanced imaging and tracking capability. This paper describes the present hardware/sensors composition, software design and preliminary experiment to combine measurement data from all the on-board sensors to derive an estimated position and orientation of the autonomous underwater vehicle during both submerged and surface operations. Copyright © 2003 IFAC

Keywords: Autonomous underwater vehicles, real time, inertial navigation, global positioning systems, image sensors, sensor fusion, data acquisition, Kalman filters, fuzzy logic.

1. INTRODUCTION

A deep mobile target has been converted into a rudimentary autonomous underwater vehicle (AUV) known as *Hammerhead*. A three year co-operative research project funded by the Engineering and Physical Sciences Research Council involving both the University of Plymouth (UoP) and Cranfield University (CU) has the objectives of designing and developing an interactive navigation system consists of a visual navigation subsystem (VNS) and an inertial navigation system/global positioning subsystem (INS/GPS) to interact with an appropriate guidance and control system.

VNS has been widely adopted as a navigation methodology for AUVs as it has the capabilities to provide high precision and high quality measurements

from image data to derive accurate relative position information. Advanced VNS applications can attempt to provide AUV global position updates, while simultaneously creating a mosaic, or composite image map of a seabed (Fleischer *et al.*, 1997) and matching current image with viewing mosaic map. There are also works using VNS for tracking and cable following (Balasuriya and Ura, 2001). In these works various image-processing techniques to extract position measurements or to identify a specific feature of an object from live video imagery are used. Given the high resolution of digital imaging, measurement accuracies on the order of millimetres and precise feature identification can be achieved. However, the methods employed in these works are limited in regimes where the object or terrain of interest is within both the field of view (FoV) and visual range of the camera. In addition to that, there is a need for

artificial light, which increases the expense and power consumption of the vehicles. The fusion of VNS with INS measurement data can be proposed as a potential solution to these problems, as INS measurements are not affected by the aforementioned factors (Fleischer *et al.*, 1997; Balasuriya and Ura, 2001). Through the technique of dead reckoning, the position of an AUV can be inferred by integrating the fused VNS and INS measurements. The problem here is that the dead reckoning is only accurate for short time durations; since the measurement noise is integrated along with the signals, the error on position accumulates quite quickly. Consequently, an external reset mechanism is required. The use of GPS to provide periodic updates and compensate drifting from the bias errors inherent when integrating INS heading for position, have been widely implemented in the navigation of AUVs. The work in this paper is an extension to the general integrated INS/GPS by fusing the VNS and INS data between intermittent GPS fixes.

The navigation system that is being developed at the UoP is based on a multisensor data fusion (MSDF) technique that can produce accurate navigation information continuously in real time from a variety of low cost inertial sensors and a GPS receiver. During an actual mission this subsystem is enhanced by data from the intelligent viewing system developed by CU, with the purpose of aiding navigation by providing velocity and altitude estimates. Once the navigation data has been suitably processed it will be fed to the guidance and control system for the appropriate action.

The aim of this paper is to describe the present hardware/sensors configuration, software design and preliminary experiments to combine measurement data from the VNS, INS and GPS to derive an estimated position of the *Hammerhead* AUV during both submerged and surface operations. The structure of this paper is as follows: the next two sections describe the current status of the *Hammerhead* VNS and INS/GPS development respectively. A proposed experiment scenario of an integrated VNS+INS/GPS is then discussed in section 4. Conclusion and future works are given in section 5.

2. HAMMERHEAD VNS DEVELOPMENT: CURRENT STATUS

2.1 General description

The *Hammerhead* VNS is based on the laser stripe illumination (LSI) methodology previously developed at CU (Tetlow and Allwood, 1995), and will provide enhanced viewing of the seabed below the vehicle. However, it also provides real time data such as altitude, velocities and tracking information to the navigation system during the mission as well as gathering images to produce a post mission enhanced optical waterfall image of a surveyed area.

There are several advantages of this type of approach over conventional imaging. LSI provides an improved image contrast at a given range. From computer simulations (Jaffe and Dunn, 1988) this type of

system becomes limited at 5-7 attenuation lengths, compared with 2-4 for a conventionally illuminated system. This allows an increased deployment altitude for seabed surveys (3-18 metres) resulting in a greater swathe and hence greater area coverage. The images produced are approximately optically flat, meaning they exhibit even illumination. Furthermore, the structured nature of the light allows additional geometric information to be derived from the image and the stripe region can be extracted for each image to form a continuously evolving 2-D intensity waterfall image. However, LSI systems are more expensive and require a greater development resource than conventional systems.

The viewing system comprises: a 100mW frequency doubled diode pumped Nd:YAG Laser (532nm), a low cost high sensitivity monochrome charged coupled device (CCD) camera with a wide angle lens, a single axis scanner and a tilt-compensated electronic compass (TCM2). Both the laser/scanner assembly and the camera are mounted within separate dedicated sections of the torpedo-shaped vehicle, with specially made plane ports to accommodate the optical path. The TCM2 compass uses two inclinometers to correct the output of three magnetometers for the declination angle error. The inclinometers, which are liquid-filled, are integral to the viewing system, providing tilt data for stabilisation when pitch and roll is experienced.

The complete sensor subsystem can be split into the three areas, with regard to their distinct utility: 1. velocity estimator; 2. active altitude sensor; 3. imaging capabilities

A computer vision application is used to derive the required navigational and tracking information in real time from acquired images. An estimate of the instantaneous speed of the vehicle can be derived by using a 2-D correlation-based window-tracker, together with an integration of a range estimate from a laser-triangulation system by which image displacements are transformed into real-world displacements. The laser-triangulation technique requires the extraction of the vertical position of the laser stripe on an image to determine the range from the centre of the camera axis to the seabed.

Together with the speed and range acquisition in real time, image quality can be checked and optimised by either changing the laser system parameters or demanding navigational changes from the vehicle. The complete video sequence is also recorded in digital-8 format and can be post-processed to produce a continuous 2-D intensity waterfall image or 3-D range images of the seabed with dimensional data and referenced against accurate positional information. These optical maps can be used to classify and locate particular objects that are of interest. This can be implemented in the image-processing suite or by manual inspection of the mosaic.

In the autumn of 2002 a set of constrained motion trials were performed at the IFREMER facility in Brest. As well as the validation of a viewing model and system calibration, these trials created a useful

archive of test files, allowing much of the future development to be possible from the dry laboratory. Furthermore, the measurements necessary to build a ground truth model were acquired in parallel to the image and vehicle specific data. This is used in ascertaining the accuracy of the vision-based navigation routines. A description of these experiments, including the construction, analysis and limitations of the ground truth model is given in a previous paper (Dalglish, *et al.*, 2003).

The next sub-sections briefly describe the sensor in terms of the two navigational components. Some recent results are presented and the means by which the outputs are to be integrated with the MSDF algorithm (described later in the paper) is discussed. More detail concerning each component and the imaging capabilities will be given in subsequent papers.

2.2 Velocity estimator

As an implementation of *window-based tracking* (Anandan, 1989) using the LabVIEW IMAQ™ Image Processing toolbox, this algorithm performs an intelligent correlation between an initial stored image region and subsequent image regions where image displacement is measured directly. The tracker uses a non-uniform sampling technique where only a few points that represent the overall content of the image are extracted. Moreover, an adaptive search strategy based on previous displacements is also incorporated to further improve tracking efficiency.

Each successful matching event outputs a confidence level based on the degree of success of the correlation. Fig. 1. illustrates the distribution of matching events for the single resolution case and the corresponding pixel error, where sub-pixel accuracy is used. This was based on a static subsea sequence of two thousand 'lossy' JPEG images. The stand-off distance was eight metres. The window size was 20 x 20 pixels.

As a more realistic alternative to adding Gaussian noise to each pixel independently, it is planned to use an 'artificial turbidity' environment in the Cranfield test tank to assess the static performance of the single resolution tracker under increasing noise. It is intended to use the confidence value as an indication of image quality, where as the confidence value degrades, the desired altitude of the AUV needs to be reduced. Outliers are detected in a smoothing stage.

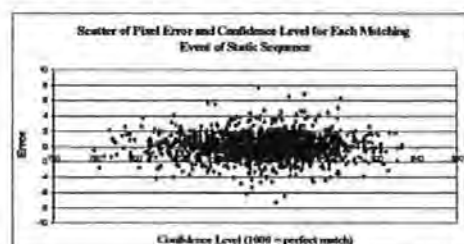


Fig. 1. Scatter plot of error vs. confidence level

The standard deviation and the mean error are 1.44 pixels and 0.03 pixels, which corresponds to 4.3cm

and 0.1cm respectively. This satisfies the Gaussian distribution condition which is desirable for the application to MSDF methods using a Kalman filter approach.

Additionally, for the Kalman filter to work properly there is a need to have a good knowledge of the measurement noise covariance matrix (R). For this system the initial value of R has been derived, and the fuzzy logic techniques discussed in section 3 are going to be used to tune the R value as it changes. Two similar static sequences can be used to produce the initial value of R .

The further development of this tracker is intended to include a robust feature identification stage, where tracking windows are placed around features deemed worth tracking, as they first pass into the FoV of the camera. This process will be described in a separate paper.

In order to transform the X and Y image velocities to real world velocities, a translational transformation is performed using estimated altitude. As well as placing a restriction on the size of the tracking window, this makes the assumption that the optical axis is perpendicular to the plane of the seabed.

The next sub-section describes the analysis of the laser triangulation process to determine the altitude of the vehicle above the seabed.

2.3 Active altitude sensor

Determining range by triangulation of a laser beam and optical sensor is a well understood and tested method. Perhaps the most engaging aspect is that of calibration. For subsea systems this has been approached in different ways (Chantler *et al.*, 1997; Spours, 2000). From the geometry of the system (Dalglish *et al.*, 2003), the measurement of the laser exit angle, the baseline and the use of the lens manufacturers data, an algebraic solution to derive altitude from stripe position in the image can be obtained. However, due to the errors in each of these values, the result is always inaccurate, particularly at increased range, and consequently, multipoint calibration routines are used, where 3rd or 4th order polynomials are used to describe the best fit of the calibration data, often using several relationships for each different range of altitude.

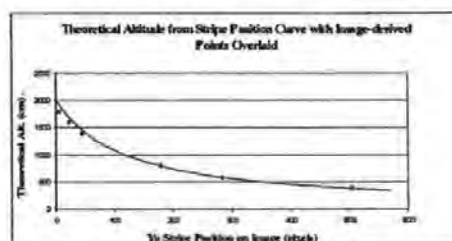


Fig. 2. Theoretical altitude from stripe position curve with image derived points overlaid

Fig. 2. shows the theoretical curve based on the geometry of the viewing system, with six points overlaid which were based on actual measurements

from which images were taken in clear filtered seawater. The extraction of the stripe was performed using intensity thresholding. It can be seen how the accuracy falls off between 14 metres and 18 metres altitude. This is mainly due to both the non-linearity of the lens as the stripe approaches the extreme of the FoV and pixel round-off errors.

However, once operating in higher turbidity water, the stripe widens and the scatter field becomes more distinct (Tetlow, 1993). It is then that the extraction of the stripe becomes more difficult, and the accuracy of the laser triangulation will suffer even once the system is properly calibrated. In this case it is necessary for the computer vision application to instruct the AUV control system to fly at a lower altitude.

3. HAMMERHEAD INS/GPS DEVELOPMENT: CURRENT STATUS

3.1 General description

The current hardware consists of an inertial measurement unit (IMU), a TCM2 compass, a GPS receiver, a pressure gauge, a shaft encoder, a laptop and a PCMCIA to 4-RS232 converter. The initial design of INS/GPS software is based on an adaptive Kalman filter (KF), which will be discussed in more detail in subsequent paragraphs.

The KF is a computational algorithm that processes measured data with the objective of producing minimum error estimates of the states of the system. Although it is termed optimal filter, there are practical limitations to the KF that may lead to its divergence (Fitzgerald, 1971). Divergence may be attributed to incomplete *a priori* knowledge of the process noise covariance matrix (Q) and the R as the measurement noise covariance matrix. For example: if R and/or Q are too small at the beginning of the estimation process, the uncertainty tube around the true value will tighten and a biased solution will result. If R and/or Q are too large, filter divergence and longer estimation of the filter could result. From the aforementioned it may be argued that the KF with fixed R and/or Q should be replaced by an adaptive estimation formulation. A fuzzy Kalman filter (FKF) technique (Loebis, *et al.*, 2003) is an improvement to the conventional KF, whereby the requirements to have a complete *a priori* knowledge of the R covariance matrices are relaxed. In this preliminary simulation work, a linear Kalman filter (LKF) and an extended Kalman filter (EKF) coupled with fuzzy logic techniques is used to fuse sensor data and make appropriate adaptation to the initial value of R .

This algorithm takes two general steps. First, the coupled LKF and fuzzy logic are utilised to fuse the response of a linearised dynamic model of an AUV after a sequence of commands is sent to its control surface. Here, four sensors measuring the response of the system are fused and form an input to the EKF. The EKF plays a role as a state estimator of a non-linear system that describes the relationship of position and velocity in body and inertial co-ordinate

system when the AUV navigates both without GPS updates *i.e.*, pure dead reckoning navigation during a submerged operation and navigates with GPS updates *i.e.*, integrated INS/GPS during a surface operation. In this preliminary simulation work, (Loebis, *et al.*, 2003), the method has shown a promising result.

It should be noted however, that this method requires the dynamic model of the system from which the states can be estimated. The problem here is that input-output relationships that enable one to model a complete dynamic behaviour of an AUV are not always available and an alternative method to loosen up this requirement is needed. To this end, an extension to the initial work on FKF (Loebis, *et al.*, 2003) is proposed. The idea behind the proposed work is that the process and measurement model of the system are mainly derived from non-linear kinematic equations of motion and only an EKF coupled with fuzzy logic algorithm will be used as a MSDF algorithm. The discussion of the algorithm is begun with the derivation of the process and measurement model and makes up the topic for the next sub-section.

3.2 Process and measurement model

Angular rate, velocity and acceleration with respect to body co-ordinate frame in x , y and z axis produced by the IMU are denoted as $[p, q, r]^T$, $[u, v, w]^T$ and $[a_x, a_y, a_z]^T$ respectively. Angular displacement of the IMU and TCM2 with respect to an inertial co-ordinate frame is described using Euler angles roll, pitch and yaw which are represented as $[\phi, \theta, \psi]^T$. The rotation matrix that transforms velocity vectors $[u, v, w]^T$ in the body co-ordinate frame to velocity vectors in the inertial co-ordinate frame is written as

$$R = \begin{bmatrix} \xi_\theta \eta_\psi & \eta_\phi \eta_\theta \xi_\psi - \xi_\phi \eta_\psi & \xi_\phi \eta_\theta \xi_\psi + \eta_\phi \eta_\psi \\ \xi_\theta \eta_\psi & \eta_\phi \eta_\theta \eta_\psi + \xi_\phi \xi_\psi & \xi_\phi \eta_\theta \eta_\psi - \eta_\phi \xi_\psi \\ -\eta_\theta & \eta_\phi \xi_\theta & \xi_\phi \xi_\theta \end{bmatrix} \quad (1)$$

where $\xi_{(\cdot)} = \cos(\cdot)$ and $\eta_{(\cdot)} = \sin(\cdot)$ and (\cdot) represents the Euler angle component.

The three inertial co-ordinate frame Euler angle rotation rates $[\dot{\phi}, \dot{\theta}, \dot{\psi}]^T$ are obtained from the body co-ordinate frame rotation rates $[p, q, r]^T$ by the non-orthogonal linear transformation in (2).

$$\dot{T} = \begin{bmatrix} 1 & \eta_\phi \eta_\theta / \xi_\theta & \xi_\phi \eta_\theta / \xi_\theta \\ 0 & \xi_\phi & -\eta_\phi \\ 0 & \eta_\phi / \xi_\theta & \xi_\phi / \xi_\theta \end{bmatrix} \quad (2)$$

The process model plays a critical role in the localisation system performance since the AUV's localisation relies on it and the dead reckoning measurements entirely while absolute observations from GPS are not available. The non-linear systems represented in matrix form as in (3), (4) and (5) are utilised as the kinematic model of the vehicle:

$$\begin{bmatrix} f_{x_k} \\ f_{y_k} \\ f_{z_k} \end{bmatrix} = \begin{bmatrix} x_{k+1} \\ y_{k+1} \\ z_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix} + \tau \cdot R_k \cdot \begin{bmatrix} u_k \\ v_k \\ w_k \end{bmatrix} \quad (3)$$

$$\begin{bmatrix} f_{u_k} \\ f_{v_k} \\ f_{w_k} \end{bmatrix} = \begin{bmatrix} u_{k+1} \\ v_{k+1} \\ w_{k+1} \end{bmatrix} = \begin{bmatrix} u_k \\ v_k \\ w_k \end{bmatrix} + \tau \cdot \begin{bmatrix} a_{x_k} \\ a_{y_k} \\ a_{z_k} \end{bmatrix} \quad (4)$$

$$\begin{bmatrix} f_{\theta_k} \\ f_{\phi_k} \\ f_{\psi_k} \end{bmatrix} = \begin{bmatrix} \theta_{k+1} \\ \phi_{k+1} \\ \psi_{k+1} \end{bmatrix} = \begin{bmatrix} \theta_k \\ \phi_k \\ \psi_k \end{bmatrix} + \tau \cdot T_k \cdot \begin{bmatrix} p_{x_k} \\ q_{y_k} \\ r_{z_k} \end{bmatrix} \quad (5)$$

where $[x_k, y_k, z_k]^T$ is the position in inertial coordinate frame, $[u_k, v_k, w_k]^T$ and $[\phi_k, \theta_k, \psi_k]^T$ are as before and k is discrete-time index and τ is the sampling interval. Thus the system state vector may be written as $x_k = [\phi_k, \theta_k, \psi_k, u_k, v_k, w_k, \phi_k, \theta_k, \psi_k]^T$, the input vector as $u_k = [a_{x_k}, a_{y_k}, a_{z_k}, p_{x_k}, q_{y_k}, r_{z_k}]^T$. The system matrix (A_k) and input gain matrix (B_k) are represented by the Jacobian $\left[\frac{\partial f}{\partial x}\right]$ and $\left[\frac{\partial f}{\partial u}\right]$ and are evaluated respectively at \hat{x}_k and u_k . The H matrix is simply an $I_{9 \times 9}$ matrix when GPS updates are available. In case of GPS updates unavailability, if H 's components can be represented as $H_{[1,1]}$, $H_{[1,2]}$ and $H_{[2,1]}$ are zero with $H_{[3,3]} \dots H_{[9,9]}$ are one. In the next sub-section, a discussion on the proposed algorithm is given.

3.3 Fuzzy extended Kalman filter

Having derived the process and measurement model, this section discusses how they are to be used in the proposed MSDF technique. This technique, the so called fuzzy extended Kalman filter (FEKF) enables the fusion of sensor data without the availability of the dynamic model of a system and resolves the limitations of the preliminary MSDF method (Loebis, *et al.*, 2003). The algorithm of standard EKF itself is very well-defined and interested readers can refer to previous work (Brown and Hwang, 1997).

The FEKF is based on an early innovation adaptive estimation (IAE) approach (Mehra, 1970) coupled with fuzzy logic techniques. Here the innovation at sample time k in the FEKF algorithm is the difference between the real measurement z_k , received by the filter and its estimated (predicted) value \hat{z}_k , and is computed as follows:

$$Inn_k = z_k - \hat{z}_k \quad (6)$$

The predicted measurement is the projection of the filter predicted states \hat{x}_k^- onto the measurement space through the measurement design function, *i.e.*

$$\hat{z}_k = h(f(\hat{x}_k^-)) \quad (7)$$

Innovation represents additional information available to the filter as a result of the new measurement. The occurrence of data with statistics different from the *a priori* information will first show up in the innovation vector. For this reason the innovation sequence represent the information content in the new observation and is considered the most relevant source of information to the filter adaptation.

The fuzzy logic is chosen mainly because of its simplicity. This motivates the interest in the topic, as testified by related articles, which have been appearing in the literature. The FEKF proposed in this paper is based on the IAE approach using a technique known as covariance-matching (Mehra, 1970). The basic idea behind this technique is to make the actual value of the covariance of the innovation sequences match its theoretical value.

The actual covariance is defined as an approximation of the Inn_k sample covariance through averaging inside a moving estimation window of size N which takes the following form:

$$\hat{C}_k = \frac{1}{M} \sum_{i=i_0}^N Inn_k Inn_k^T \quad (8)$$

where $i_0 = k - M + 1$ is the first sample inside the estimation window. An empirical experiment is conducted to choose the window size M . From experimentation it was found in the previous work (Loebis, *et al.*, 2003) that a good size for the moving window in (8) is 15.

The theoretical covariance of the innovation sequence is defined as:

$$S_k = H_k P_k^- H_k^T + R_k \quad (9)$$

The logic of the adaptation algorithm using covariance matching technique can be qualitatively described as follows. If the actual covariance value \hat{C}_k is observed, whose value is within the range predicted by theory S_k and the difference is very near to zero, this indicates that both covariances match almost perfectly and only a small change is needed to be made on the value of R . If the actual covariance is greater than its theoretical value, the value of R should be decreased. On the contrary, if \hat{C}_k is less than S_k , the value of R should be increased. This adjustment mechanism lends itself very well to being dealt with using a fuzzy-logic approach, based on rules of the kind:

$$\text{IF } \langle \text{antecedent} \rangle \text{ THEN } \langle \text{consequent} \rangle \quad (10)$$

where antecedent and consequent are of the form $\aleph \in M_i$, $\Im \in N_i$, $i = 1, 2, \dots$ respectively, where \aleph and \Im are the input and output variables, respectively, and M_i and N_i are the fuzzy sets.

To implement the above covariance matching technique using the fuzzy logic approach, a new variable called δ_k , is defined to detect the discrepancy between \hat{C}_k and S_k . The following three fuzzy rules of the kind (10) are used:

IF $\langle \delta_k \approx 0 \rangle$ THEN $\langle R_k \text{ is unchanged} \rangle$ (11)

IF $\langle \delta_k > 0 \rangle$ THEN $\langle R_k \text{ is decreased} \rangle$ (12)

IF $\langle \delta_k < 0 \rangle$ THEN $\langle R_k \text{ is increased} \rangle$ (13)

Thus R is adjusted according to,

$$R_k = R_{k-1} + \Delta R_k \quad (14)$$

where ΔR_k is added or subtracted from R at each instant of time. Here δ_k is the input to the fuzzy inference system (FIS) and ΔR_k is the output. On the basis of the above adaptation hypothesis, the FIS can be implemented using three fuzzy sets for δ_k ; $N = \text{Negative}$, $Z = \text{Zero}$ and $P = \text{Positive}$. For ΔR_k the fuzzy sets are specified as $I = \text{Increase}$, $M = \text{Maintain}$ and $D = \text{Decrease}$.

4. HAMMERHEAD VNS+INS/GPS INTEGRATION EXPERIMENT

For the purpose of VNS+INS/GPS integration, several open water experiments with a predefined mission scenario will be performed (Fig. 3). The purpose of the experiment is to acquire real time sensor measurement data and visual information. This will enable the creation of simulation environment test files. In this experiment, an initial GPS signal will be acquired before the vehicle submerges. When underwater, the vehicle is steered automatically to move in a spiral- or snake track- like trajectory and the INS measurement data plus visual images of the seabed will be continuously acquired. At a certain point in time, the CCD camera will detect and identify pre-deployed visual landmarks, whose position in 3D have been surveyed accurately before the actual mission taking place. These landmarks will be used to verify the accuracy of the position obtained using the integrated INS/VNS that will be developed afterwards in a simulation environment. Having completed the predefined trajectory, the vehicle will be steered to come up to the surface, where another GPS signal will be acquired.

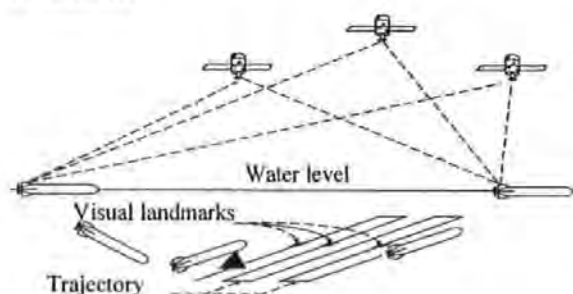


Fig. 3. VNS+INS/GPS mission scenario

The TCM2 and IMU angular displacement data acquired in this experiment will be fused using the

FEKF technique using the process and measurement model as described earlier. Herein, the authors will focus on using the technique to adjust the R matrix only. The same technique is then applied to fuse the redundant velocity measurement data from VNS and IMU. Results will be reported in future papers.

5. CONCLUSION AND FUTURE WORKS

The background and the justification of the work have been presented. The hardware configurations and preliminary experiment for the VNS are discussed followed by a brief discussion on the hardware of the INS/GPS plus the initial MSDF algorithm to fuse the sensor data in this subsystem. An extension of this algorithm together with a plan for a future real-time trial to implement the algorithm has also been presented. In addition, the authors have made provision to deploy other sensors on-board the vehicle to provide a more accurate vertical position estimation. These sensors are a sonar altimeter and an upward looking sonar which will be fused with the laser-derived altitude output and the pressure sensor depth output respectively.

REFERENCES

- Anandan, P. (1989). A Computational Framework and an Algorithm for the Measurement of Visual Motion. *International Journal of Computer Vision*, vol. 2, pp. 283-310.
- Balasuriya, A. and T. Ura (2001). Autonomous Underwater Vehicle Navigation Scheme for Cable Following. *2001 IEEE Intelligent Transportation Systems Conference Proceedings*, pp. 519-524, Oakland, CA, USA.
- Brown, R.G. and P.Y.C. Hwang (1997). *Introduction to Random Signals and Applied Kalman Filtering*, Chapter 9. John Wiley & Sons Inc., New York.
- Chantler, M.J., J. Clark and M. Umasuthan (1997). Calibration and Operation of an Underwater Laser Triangulation Sensor: The Varying Baseline Problem. *Optical Engineering*, vol. 36, no. 9, pp. 2604-2611.
- Dalglish, F.R., S. Tetlow and R. L. Allwood (2003). Preliminary Experiments in the Development of a Laser-based Imaging Sensor for AUV Navigation. *Proceedings of the 1st IFAC Workshop on Guidance and Control of Underwater Vehicles*, Newport, South Wales, UK, pp. 239-244.
- Fitzgerald, R.J. (1971). Divergence of the Kalman Filter. *IEEE Transaction on Automatic Control*, vol. AC-16, no. 6, pp. 736-747.
- Fleischer, S.D., M.R. Rock and R. Burton (1997). Global Position Determination and Vehicle Path Estimation from a Vision Sensor for Real Time Video Mosaicking and Navigation. *Proceedings of the Oceans '97 MTS/IEEE*, pp. 641-647, Halifax, Nova Scotia, Canada.
- Jaffe, J. and C. Dunn (1988). A Model-Based Comparison of Underwater Imaging Systems. *SPIE*, vol. 925, *Ocean Optics IX*, pp. 344-350.
- Loebis, D., R. Sutton and J. Chudley (2003). A Fuzzy Kalman Filter for Accurate Navigation of an Autonomous Underwater Vehicle. *Proceedings of the 1st IFAC Workshop on Guidance and Control of Underwater Vehicles*, Newport, South Wales, UK, pp. 161-166.
- Mehra, R.K. (1970). On the Identification of Variances and Adaptive Kalman Filtering. *IEEE Transactions on Automatic Control*, vol. AC-15, no. 2, pp. 175-184.
- Spours, J. (2000). The Use of a Structured Laser Light System to Ascertain Three Dimensional Measurements of Underwater Work Sites. *PhD Thesis, Cranfield University, UK*.
- Tetlow, S. (1993). Use of Laser Light Stripes to Reduce Backscatter in an Underwater Viewing System. *PhD Thesis, Cranfield University, UK*.
- Tetlow, S. and R. L. Allwood (1995). Development and Applications of a Novel Underwater Laser Illumination System. *Underwater Technology*, vol. 21, no. 2, pp. 13-20.

GUIDANCE AND CONTROL OF UNDERWATER VEHICLES 2003

*A Proceedings volume from the IFAC Workshop
Newport, South Wales, UK, 9 - 11 April 2003*

Edited by
G. N. ROBERTS, R. SUTTON
and
R. ALLEN

A FUZZY KALMAN FILTER FOR ACCURATE NAVIGATION OF AN AUTONOMOUS UNDERWATER VEHICLE

D. Loebis, R. Sutton and J. Chudley

*Marine and Industrial Dynamic Analysis Research Group
Department of Mechanical and Marine Engineering
The University of Plymouth
Drake Circus, Plymouth, PL4 8AA, United Kingdom*

Abstract: An autonomous underwater vehicle requires a navigation system suite that has the capability to provide optimal estimates of the position and attitude of the vehicle. To this purpose, in this paper the data from a variety of inertial navigation systems (INS) sensors are first fused together by means of a linear Kalman filter (LKF) before being integrated with global positioning system (GPS) data through the use of an extended Kalman filter (EKF). To cope with divergence problem caused by the insufficiently known *a priori* filter statistics, the performance of both the KF and EKF are improved by introducing a fuzzy-rule-based adaptation scheme. Copyright © 2003 IFAC

Keywords: Autonomous underwater vehicles, navigation, sensor fusion, Kalman filters, extended Kalman filters, fuzzy logic.

1. INTRODUCTION

In the past few decades, there have been numerous worldwide research and development activities in order to explore the oceans of the world. As an ocean is an inherently hostile and hazardous environment, the need for an underwater robotic system, especially one with high reliability and a fully built-in intelligence, becomes apparent. The autonomous underwater vehicle (AUV) class of vessel meets these requirements. Tasks performed by AUVs include cable tracking and inspection (Asakawa *et al.*, 2000) and seafloor mapping and ordnance location (Wright *et al.*, 1996).

To achieve truly autonomous behaviour, an AUV must be able to locate itself accurately during an operating scenario using only its onboard sensors. In the past, fusing of inertial navigation system (INS) sensors and the integration with the global positioning system (GPS) through the use of a conventional linear Kalman filter (LKF), whose

algorithm is given in appendix A, and an extended Kalman filter (EKF) have been a popular method of localisation of an AUV (Yun *et al.*, 1999). However, a significant difficulty in designing a KF (refers to both LKF and EKF) can often be traced to incomplete *a priori* knowledge of the process covariance matrix (Q) and measurement covariance matrix (R). In most practical applications, these matrices are initially estimated or even unknown. The problem here is that the optimality of the estimation algorithm in the KF setting is closely connected to the quality of *a priori* information about the process and measurement noise (Mehra, 1970). It has been shown that insufficiently known *a priori* filter statistics can on the one hand reduce the precision of the estimated filter states or introduces biases to their estimates. In addition, inaccurate *a priori* information can lead to practical divergence of the filter (Fitzgerald, 1971). For example: if R and/or Q are too small at the beginning of the estimation process, the uncertainty tube around the true value will tighten and a biased solution will result. If R

and/or Q are too large, filter divergence and longer estimation of the filter could result. From the aforementioned it may be argued that the conventional KF with fixed R and/or Q should be replaced by an adaptive estimation formulation.

In this paper, an innovation adaptive estimation (IAE) approach (Mehra, 1970) coupled with fuzzy logic techniques is used to adjust the R matrix of the KF. Here the innovation Inn_k at sample time k in the KF algorithm is the difference between the real measurement z_k , Eq. (A2), received by the filter and its estimated (predicted) value \hat{z}_k , and is computed as follows:

$$Inn_k = z_k - \hat{z}_k \quad (1)$$

The predicted measurement is the projection of the filter predicted states \hat{x}_k^- onto the measurement space through the measurement design matrix H_k , i.e.

$$\hat{z}_k = H_k \hat{x}_k^- \quad (2)$$

Innovation represents additional information available to the filter as a result of the new measurement z_k . The occurrence of data with statistics different from the *a priori* information will first show up in the innovation vector. For this reason the innovation sequence represent the information content in the new observation and is considered the most relevant source of information to the filter adaptation.

2. THE ADAPTIVE ESTIMATION ALGORITHM

2.1 Fuzzy Kalman filter.

In this section, an on-line innovation-based adaptive scheme of the KF to adjust the R matrix employing the principles of fuzzy logic is presented. The fuzzy logic is chosen mainly because of its simplicity. This motivates the interest in the topic, as testified by related articles which have been appearing in the literature (Kobayashi *et al.*, 1998; Jetto *et al.*, 1999).

The fuzzy logic Kalman filter (FKF) proposed in this paper is based on the IAE approach using a technique known as covariance-matching (Mehra, 1970). The basic idea behind this technique is to make the actual value of the covariance of the innovation sequences match its theoretical value.

The actual covariance is defined as an approximation of the Inn_k sample covariance through averaging inside a moving estimation window of size N (Escamilla-Ambrosio and Mort, 2001) which takes the following form:

$$\hat{C}_k = \frac{1}{M} \sum_{i=i_0}^N Inn_k Inn_k^T, \quad (3)$$

where $i_0 = k - M + 1$ is the first sample inside the estimation window. An empirical experiment is conducted to choose the window size M . The window size should be long enough to capture the dynamics of the innovation sequences as to provide statistical smoothing and prevent the occurrence of bias in the approximation. From experimentation it was found that a good size for the moving window in (3) is 15.

The theoretical covariance of the innovation sequence is defined as

$$S_k = H_k P_k^- H_k^T + R_k, \quad (4)$$

which is part of the KF algorithm, Eq. A8.

The logic of the adaptation algorithm using covariance matching technique can be qualitatively described as follows. If the actual covariance value \hat{C}_k is observed, whose value is within the range predicted by theory S_k and the difference is very near to zero, this indicates that both covariances match almost perfectly and only a small change is needed to be made on the value of R . If the actual covariance is greater than its theoretical value, the value of R should be decreased. On the contrary, if \hat{C}_k is less than S_k , the value of R should be increased. This adjustment mechanism lends itself very well to being dealt with using a fuzzy-logic approach based on rules of the kind:

$$\text{IF } \langle \text{antecedent} \rangle \text{ THEN } \langle \text{consequent} \rangle, \quad (5)$$

where antecedent and consequent are of the form $x \in M_i$, $y \in N_j$, $i = 1, 2, \dots$ respectively, where x and y are the input and output variables, respectively, and M_i and N_j are the fuzzy sets.

To implement the above covariance matching technique using the fuzzy logic approach, a new variable called *delta*, is defined to detect the discrepancy between \hat{C}_k and S_k . The following three fuzzy rules of the kind (5) are used:

$$\text{IF } \langle \text{delta}_k \equiv 0 \rangle \text{ THEN } \langle R_k \text{ is unchanged} \rangle, \quad (6)$$

$$\text{IF } \langle \text{delta}_k > 0 \rangle \text{ THEN } \langle R_k \text{ is decreased} \rangle, \quad (7)$$

$$\text{IF } \langle \text{delta}_k < 0 \rangle \text{ THEN } \langle R_k \text{ is increased} \rangle \quad (8)$$

Thus R is adjusted according to,

$$R_k = R_{k-1} + \Delta R_k, \quad (9)$$

where ΔR_k is added or subtracted from R at each instant of time. Here *delta*_k is the input to the fuzzy inference system (FIS) and ΔR_k is the output.

On the basis of the above adaptation hypothesis, the FIS can be implemented using three fuzzy sets for

$\delta\alpha_k$; N = Negative, Z = Zero and P = Positive. For ΔR_k the fuzzy sets are specified as I = Increase, M = Maintain and D = Decrease. The membership functions of these fuzzy sets are shown in Figure 1.

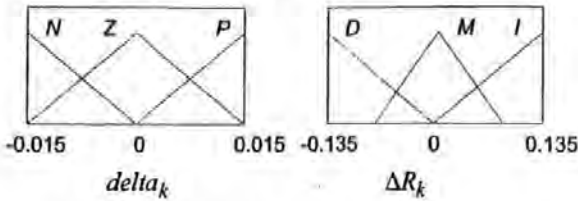


Fig. 1 Membership function of $\delta\alpha_k$ and ΔR_k

2.2 Sensor fault diagnostic and recovery algorithm.

In addition to the adaptation procedure, the FKF has been equipped with the sensor fault diagnostic and recovery algorithm as proposed by Escamilla-Ambrosio and Mort (2001). The basic idea behind this algorithm is that the amplitude of the actual value of the Inn_k and its theoretical value ($\sqrt{S_k}$) for a sensor without any fault must be around 1, but it increases abruptly if a transient or persistent fault is present in the measurement data. For this purpose a variable $InnC_k$ is defined as,

$$InnC_k = \frac{|Inn_k|}{\sqrt{S_k}} \quad (10)$$

Thus, if the value of $InnC_k$ is greater or equal than a threshold (α) then a transient fault is declared and Inn_k is assigned a value of 0. If $InnC_k$ is still greater than α for an instant of time, the persistent fault is declared and Inn_k is assigned a value of $\sqrt{S_k}$ multiplied by a random number. From experimentation it was found that the good value of α is 1.2.

2.3 Fuzzy logic observer

To monitor the performance of a FKF, another FIS called the fuzzy logic observer (FLO) is used. The FLO assigns a degree of confidence denoted as c_k , a number on the interval $[0,1]$, to the FKF state estimate. The FLO is implemented using two inputs: the values of $|\delta\alpha_k|$ and R_k . The membership functions of these variables are shown in Figure 2.

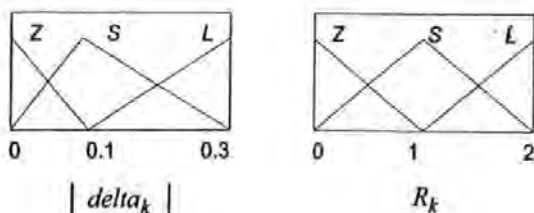


Fig. 2. Membership function of $|\delta\alpha_k|$ and R_k

The fuzzy labels for the membership functions: Z = Zero, S = Small and L = Large. Three fuzzy singletons are defined for the output c_k and are labelled as G = Good, AV = Average and P = Poor with values 1, 0.5 and 0 respectively. The basic heuristic hypothesis for the FLO is as follows: if the value of $|\delta\alpha_k|$ is near to zero and the value of R_k is near to zero, then the FKF works almost perfectly and the state estimate of the FKF is assigned a degree of confidence near 1. On the contrary if one or both of these values increases far from zero, it means that the FKF performance is degrading and the FLO assigns a degree of confidence near 0. Table 1 gives the complete fuzzy rule base of each FLO.

Table 1 Fuzzy rule base FLO

R_k	Z	S	L
$ \delta\alpha_k $			
Z	G	G	AV
S	G	AV	P
L	AV	P	P

3. SENSOR FUSION OF INS SENSOR DATA

In this section, the FKF algorithm is applied to a linearised steering model of an AUV at forward speed $u = 1.3 \text{ ms}^{-1}$ (Lea, 1998) as following:

$$\begin{bmatrix} \dot{v}(t) \\ \dot{r}(t) \\ \dot{\mathcal{L}}(t) \end{bmatrix} = \begin{bmatrix} -2.09 & 0.376 & 0 \\ -7.96 & -8.69 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v(t) \\ r(t) \\ \mathcal{L}(t) \end{bmatrix} + \begin{bmatrix} 1.07 \\ -14.1 \\ 0 \end{bmatrix} \ddot{a}_r(t) + w(t), \quad (11)$$

$$z_k = [0 \ 0 \ 1] \begin{bmatrix} v(t) \\ r(t) \\ \mathcal{L}(t) \end{bmatrix} + v(t), \quad (12)$$

where $v(t)$, $r(t)$ and $\mathcal{L}(t)$ represent the sway velocity, yaw rate of turn and yaw angle. The $w(t)$ and $v(t)$ are both zero mean white noise for the system and measurement models respectively and $\ddot{a}_r(t)$ the rudder deflection. A sample time of 0.125s is used to discretised the linearised model. The initial conditions are $[\hat{v}_0 \ \hat{r}_0 \ \hat{\mathcal{L}}_0]^T = [0 \ 0 \ 0]^T$, $P_0 = 0.01 I_3$ and Q is made constant as $0.01 I_3$. A sinusoidal input was applied to the rudder. Four yaw sensors with different noise characteristics are considered to measure the response of the vehicle. The actual value of R for each sensor is assumed unknown but its initial value is selected as 1. The FKF algorithm was then implemented and simulation results are shown in the next section.

3.1 Simulation Result

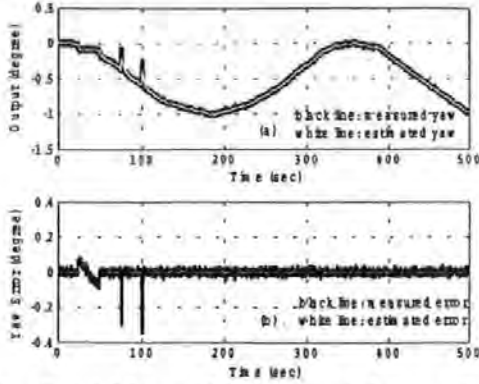


Fig. 3. (a) Measured and estimated yaw output, (b) measured and estimated yaw error of sensor 1.

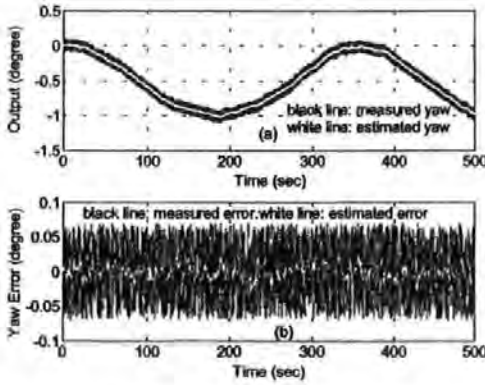


Fig. 4. (a) Measured and estimated yaw output, (b) measured and estimated yaw error of sensor 2.

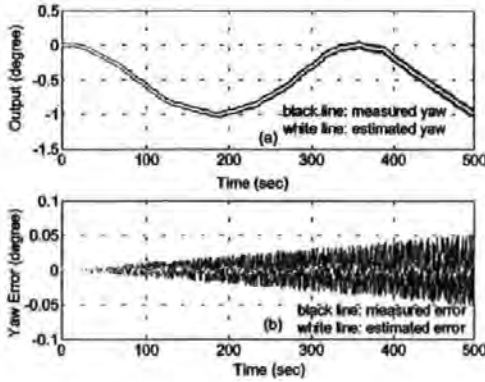


Fig. 5. (a) Measured and estimated yaw output, (b) measured and estimated yaw error of sensor 3.

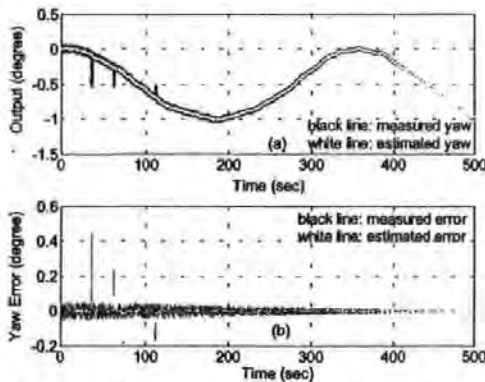


Fig. 6. (a) Measured and estimated yaw output, (b) measured and estimated yaw error of sensor 4.

Figure 3, and Figure 4, are the simulation result showing the response of the AUV observed by sensors with constant Gaussian noise, while Figure 5 and Figure 6, by sensors with uniform noise increasing and decreasing with time respectively.

To fuse the estimated yaw, a centre of gravity method is used,

$$\hat{z}_k = \frac{\sum_{i=1}^n \hat{z}_{ki} c_{ki}}{\sum_{i=1}^n c_{ki}} \quad (13)$$

where \hat{z}_{ki} is the output of the i -th FKF ($i=1,2,3,4$) and c_{ki} is the respective degree of confidence at instant time k . Figure 7, shows the comparison of the actual and the fused estimated yaw.

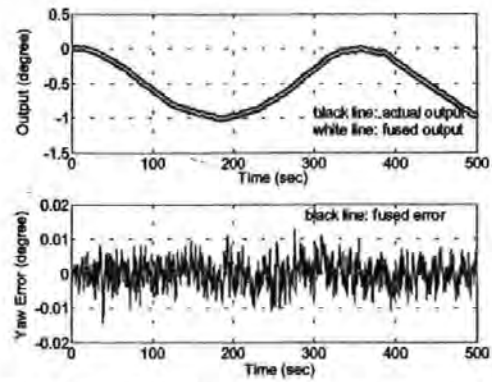


Fig. 7. (a) Actual and fused yaw output, (b) fused error.

Finally, the following performance measure are adopted for comparison purposes,

$$J_{zy} = \sqrt{\frac{1}{n} \sum_{k=1}^n (z_{ak} - z_k)^2} \quad (14)$$

$$J_{ze} = \sqrt{\frac{1}{n} \sum_{k=1}^n (z_{ak} - \hat{z}_k)^2} \quad (15)$$

where z_{ak} is the actual value of the yaw, z_k is the measured yaw, \hat{z}_k is the estimated yaw at an instant of time k and n = number of samples.

Table 2 shows the comparison of performance of each individual FKF and those obtained by the proposed INS sensor fusion method.

Table 2 Comparison of performance

Sensor	Performance	
	J_{zy} (degree)	J_{ze} (degree)
Sensor 1	1.5291	1.0935
Sensor 2	2.2908	0.5018
Sensor 3	0.9933	0.2870

Sensor 4	1.1996	0.2920
Sensor Fusion		0.1994

4. GPS/INS NAVIGATION

In this section, the fused estimated yaw obtained previously is treated as a single imaginary yaw sensor and used by other INS sensors to transform data from body co-ordinate to Earth co-ordinate frame where integration with GPS data is performed.

A continuous time model of the vehicle motion appropriate to this problem is taken to be

$$\dot{X}(t) = F(X(t)) + W(t) \quad (16)$$

$$Z(t) = H(X(t)) + V(t) \quad (17)$$

Denoted by $X(t) = [Long(t) Lat(t) \phi_e(t) r(t) u(t) v(t)]^T$ is the model states. $Long(t)$ and $Lat(t)$ are the longitude and latitude of the AUV position in Earth co-ordinate frame which are obtained from a GPS receiver, $\phi_e(t)$ is the yaw angle obtained from the imaginary yaw sensor, $r(t)$ is yaw rate, $u(t)$ and $v(t)$ are the surge and sway velocity respectively.

In this system model, F and H are both continuous function, continuously differentiable in $X(t)$. The $W(t)$ and $V(t)$ are both zero mean white noise for the system and measurement models respectively.

The model states are related through the following kinematically based set of functions ($F(X(t))$ in Eq. 16):

$$\dot{u}(t) = 0, \quad (18)$$

$$\dot{v}(t) = 0, \quad (19)$$

$$\dot{\phi}_e(t) = r(t), \quad (20)$$

$$\dot{r}(t) = 0, \quad (21)$$

$$\dot{Long}(t) = u(t) \cos \phi_e(t) - v(t) \sin \phi_e(t), \quad (22)$$

$$\dot{Lat}(t) = u(t) \sin \phi_e(t) + v(t) \cos \phi_e(t), \quad (23)$$

The output measurements are related through the states by the identity matrix $H(X(t))$. To obtain an EKF with an effective state prediction equation in a simple form, the continuous time model of (18)-(23) have been linearised about the current state estimates, producing the following form:

$$\dot{X}(t) = AX(t) + W(t), \quad (24)$$

$$Z(t) = CX(t) + V(t), \quad (25)$$

where $A =$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -u(t) \sin \phi_e(t) - v(t) \cos \phi_e(t) & 0 & \cos \phi_e(t) & -\sin \phi_e(t) \\ 0 & 0 & u(t) \cos \phi_e(t) - v(t) \sin \phi_e(t) & 0 & \sin \phi_e(t) & \cos \phi_e(t) \end{bmatrix}$$

and C is an identity matrix. Subsequent discretisation with period $T = 0.125s$ of the linearised model results in an EKF algorithm similar to the LKF algorithms in appendix A, only this time the A matrix is updated at every iteration. The initial conditions are $P_0 = 0.01 I_6$ and Q is made constant as $\text{diag}[0 \ 0 \ 0.000001 \ 0.01 \ 0.01 \ 0.01]$. The actual value of R is assumed unknown but its initial value is selected as $\text{diag}[100 \ 100 \ 0 \ 0 \ 0.1 \ 0.1]$.

The FKF algorithm from section 2.1 is then implemented, only this time the adaptation of the (i,i) -th element of R_k is made in accordance with the (i,i) -th element of ΔR_k . Here a single-input-single-output (SISO) FIS as shown in Figure 1., is used sequentially to generate the correction factors for the elements in the main diagonal of R_k as the following,

$$R_k(i,i) = R_{k-1}(i,i) + \Delta R_k, \quad (26)$$

4.1 Simulation Results

Figure 8. and Figure 9. show the AUV trajectory obtained using GPS, INS sensors and GPS/INS integration. As the initial value of R for both the $Long(t)$ and $Lat(t)$ is 100, the EKF algorithm puts less weight on the position obtained by GPS and more on the prediction of position obtained from dead reckoning method (using INS sensors data). This results in an estimated trajectory with RMSE of 2.5311 m for longitude and 2.1917 m for latitude.

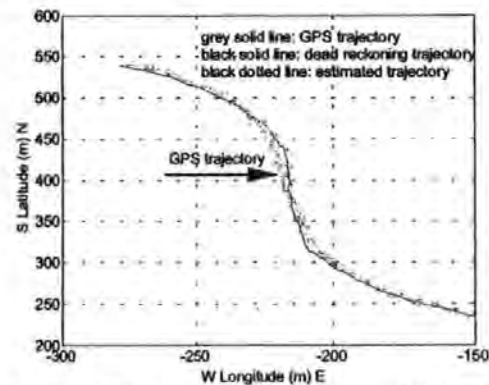


Fig. 8. AUV trajectory obtained using GPS, INS sensors (dead reckoning method) and GPS/INS using EKF without adaptation.

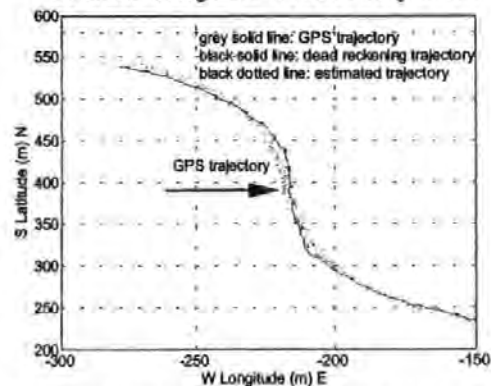


Fig. 9. AUV trajectory obtained using GPS, INS sensors (dead reckoning method) and GPS/INS using EKF with adaptation.

With the proposed FKF adaptation, the RMSE for longitude is 2.2883 m and 2.1134 m for latitude. This accuracy can increase as the distance travelled by the AUV increases.

5. SUMMARY AND CONCLUSION

The problem with incomplete *a priori* knowledge of Q (process covariance matrix) and R (measurement covariance matrix) is considered. In this paper, an adaptive Kalman filter approach, based on the filter innovation sequence coupled with fuzzy logic techniques is discussed as an alternative for fusing INS sensor data and integrating INS/GPS position information. Implementation of this approach to a linearised steering model of an AUV, whose responses are measured with sensors with different noise characteristics, has shown a promising result in improving the estimation of the individual KF. The use of FLO also plays an important role in determining the degree of confidence of the FKF output.

The choice of membership functions used in the FISs is carried out using a heuristic approach. This was time consuming. Currently the authors are exploring the use of adaptive search techniques for optimising the membership functions.

APPENDIX A

Kalman filter Equations

Given a discrete-time controlled process described by the linear stochastic difference equations:

$$x_k = A_k x_{k-1} + B_k u_k + w_k \quad (A1)$$

$$z_k = H_k x_k + v_k \quad (A2)$$

Where x_k is an $n \times 1$ system state vector, A_k is an $n \times n$ transition matrix, u_k is an $l \times 1$ vector of the input forcing function, B_k is an $n \times l$ matrix, w_k is an $n \times 1$ process noise vector, H_k is a $m \times n$ measurement matrix and v_k is a $m \times 1$ measurement noise vector. Both the w_k and v_k are assumed to be uncorrelated zero-mean Gaussian white noise sequences with covariance given by

$$E[w_k w_i^T] = \begin{cases} Q, & i = k \\ 0, & i \neq k \end{cases} \quad (A3)$$

$$E[v_k v_i^T] = \begin{cases} R, & i = k \\ 0, & i \neq k \end{cases} \quad (A4)$$

$$E[w_k v_i^T] = 0, \text{ for all } k \text{ and } i \quad (A5)$$

The KF algorithm can be organised into time update and measurement update equations,

Time update equations:

$$\hat{x}_{k+1}^- = A_k \hat{x}_k + B_k u_k \quad (A6)$$

$$P_{k+1}^- = A_k P_k A_k^T + Q_k \quad (A7)$$

Measurement update equations:

$$K_k = P_k^- H_k^T [H_k P_k^- H_k^T + R_k]^{-1} \quad (A8)$$

$$\hat{x}_k = \hat{x}_k^- + K_k [z_k - H_k \hat{x}_k^-] \quad (A9)$$

$$P_k = [I - K_k H_k] P_k^- \quad (A10)$$

The measurement update equations incorporate a new observation into the *a priori* estimate from the time update equations to obtain an improved *a posteriori* estimate. In the time and measurement update equations, \hat{x}_k is an estimate of the system state vector x_k , and P_k is the covariance matrix of the state estimation error.

REFERENCES

- Asakawa, K., J. Kojima, Y. Kato, S. Matsumoto and N. Kato (2000). Autonomous Underwater Vehicle Aqua Explorer 2 for Inspection of Underwater Cables. *Proceedings of the 2000 International Symposium on Underwater Technology*, pp. 242-247, Tokyo, Japan.
- Escamilla-Ambrosio, P.J. and N. Mort (2001). A Hybrid Kalman Filter-Fuzzy Logic Multisensor Data Fusion Architecture with Fault Tolerant Characteristics. *Proceedings of 2001 International Conference on Artificial Intelligence (IC-AI'2001)*, pp. 361-367, Las Vegas, NV, USA.
- Fitzgerald, R.J. (1971). Divergence of the Kalman Filter. *IEEE Transaction on Automatic Control*, vol. AC-16, no. 6, pp. 736-747.
- Jetto, L., S. Longhi and D. Vitali (1999). Localization of a Wheeled Mobile Robot by Sensor Data Fusion Based on a Fuzzy Logic Adapted Kalman Filter. *Control Engineering Practice*, vol. 7, pp. 763-771.
- Kobayashi, K., K.C. Cheok, K. Watanabe and F. Muneke (1998). Accurate Differential Global Positioning System via Fuzzy Logic Kalman Filter Sensor Fusion Technique. *IEEE Transactions on Industrial Electronics*, vol. 45, no. 3, pp. 510-518.
- Lea, R. K. (1998). Control of a Tethered Underwater Flight Vehicle. *PhD Thesis*, University of Southampton.
- Mehra, R.K. (1970). On the Identification of Variances and Adaptive Kalman Filtering. *IEEE Transactions on Automatic Control*, vol. AC-15, no. 2, pp. 175-184.
- Wright, J., K. Scott, C. Tien-Hsin, B. Lau, J. Lathrop and J. McCormick (1996). Multi-Sensor Data Fusion for Seafloor Mapping and Ordnance Location. *Proceedings of the 1996 Symposium on Autonomous Underwater Vehicle Technology*, pp. 167-175, Monterey, CA, USA.
- Yun, X., R.E. Bachmann, R.B. McGhee, R.H. Whalen, R.L. Roberts, R.G. Knapp, A.J. Healey and M.J. Zyda (1999). Testing and Evaluation of an Integrated GPS/INS System for Small AUV Navigation. *IEEE Journal of Oceanic Engineering*, vol. 24, no. 3, 396 - 404.

Journal of Marine Engineering and Technology

Nº A1 2002 ISSN 1476-1548



DART

Proceedings of The Institute of Marine Engineering, Science and Technology

Review of multisensor data fusion techniques and their application to autonomous underwater vehicle navigation

D Loebis, R Sutton and J Chudley
Marine and Industrial Dynamic Analysis Group
Department of Mechanical and Marine Engineering
The University of Plymouth

A key problem with autonomous underwater vehicles is being able to navigate in a generally unknown environment. The available underwater sensor suites have a limited capability to cope with such a navigation problem. In practice, no single sensor in the underwater environment can provide the level of accuracy, reliability and the coverage of information necessary to perform underwater navigation. Therefore there is a need to use a number of sensors and combine their information to provide the necessary navigation capability in a synergetic manner. This may be achieved by employing multisensor data fusion (MSDF) techniques and these are the subject of the material presented in this paper.

INTRODUCTION

The oceans cover 70% of the Earth's surface and contain an abundance of living and non-living resources that remain largely untapped waiting to be discovered. However, a number of complex issues, mainly caused by the nature of underwater environments, make exploration and protection of these resources difficult to perform. In the past few decades, various world-wide research and development activities in underwater robotic systems have increased in order to meet this challenge. One class of these systems is tethered and remotely operated and referred to as remotely operated vehicles (ROVs). Extensive use of ROVs is currently limited to a few applications because of very high operational costs and the need for human presence in conducting a mission. The demand for a more sophisticated underwater robotic technology that minimises the cost and eliminates the need for human operator and is therefore capable of operating autonomously, becomes apparent. These requirements led to the development of autonomous underwater vehicles (AUVs).

To achieve truly autonomous behaviour, an AUV must be able to navigate accurately within an area of operation. In order to achieve this, an AUV needs to employ a navigation sensor with a high level of accuracy and reliability. However, in

practice, as will be discussed in the next section, a single sensor alone may not be sufficient to provide an accurate and reliable navigation system, as it can only operate efficiently under certain conditions or it has inherent limitations when operating in underwater environments. It is therefore necessary to use a number of sensors and combine their information to provide the necessary navigation capability. To achieve this, a multisensor data fusion (MSDF) approach, which combines data from multiple sensors and related information from associated databases, can be used.

The aim of this paper is to survey previous work and recent development in AUV navigation and to introduce MSDF techniques as a means of improving the AUV's navigation capability. The structure of this paper is as follows: the next section describes the navigation systems that are currently being used in AUVs. MSDF is then discussed, whilst MSDF using specific sensor combinations applied to the navigation of AUVs follow.

AUTONOMOUS UNDERWATER VEHICLE NAVIGATION

Navigation systems used by AUVs that are discussed here include dead reckoning, radio, optical, acoustic and terrain-relative navigation.

Dead Reckoning Navigation

Dead reckoning is a mathematical means to determine position estimates when the vehicle starts from a known point and moves at known velocities. The present position is equal to the time integral of the velocity. Measurement of the vector velocity components of the vehicle is usually accomplished with a compass (to obtain direction) and a water speed sensor (to obtain magnitude). The principal problem is that the presence of an ocean current can add a velocity component to the vehicle, which is not detected by the speed sensor.

An inertial navigation system (INS) is a dead reckoning technique that obtains position estimates by integrating the signal from an accelerometer, which measures the vehicle's acceleration. The vehicle position is obtained in principle by double integration of the acceleration. The orientation of the accelerometer is governed by means of a gyroscope, which maintains either a fixed or turning position as prescribed by some steering function. The orientation may also, in principle, be determined by integration of the angular rates of the gyroscope. Both the accelerometer and the gyroscope depend on inertia for their operation.

A dead reckoning navigation system is attractive mainly because it uses sensors that are self-contained and able to provide fast dynamic measurements. Unfortunately in practice, this integration leads to unbounded growth in position error with time due to the noise associated with the measurement and the nonlinearity of the sensors, and there is no built-in method for reducing this error. Depending on the sensors used and the specific vehicle mission, the navigational error can grow rapidly to the point where either the mission will not produce useful data or it will not be achievable at all.

Two types of dead reckoning sensors have been widely employed in AUVs: inertial measurements units (IMUs) and Doppler velocity sonar (DVS). Many very accurate IMUs have been developed for submarines. However, these are typically very expensive devices and are used only in naval vehicles. Lower cost IMUs have been used in AUVs¹. However, due to the low acceleration encountered in autonomous underwater vehicles, these units are not normally of sufficient accuracy to provide stand-alone navigation.

DVS sensors provide measurement of a velocity vector with respect to the sea floor. These sensors normally comprise three or more separate sound beams allowing construction of a full three-dimensional velocity vector. Typically, these instruments have specifications of about 1% of the distance travelled². However, these results can only be achieved when the speed of sound in the AUVs area of operation does not vary significantly as a result of changes in the salinity, temperature and density of the water. Therefore, as in the IMU case, these units are not normally used to provide stand-alone navigation.

Radio Navigation

Radio navigation systems mainly use the global positioning system (GPS)³. The GPS is a satellite-based navigational system that provides the most accurate open ocean navigation available. GPS consists of a constellation of 24 satellites that orbit the earth in 12 hours. There are six orbital planes (with nominally four satellites in each) equally spaced (60 degrees apart) and inclined at about 55 degrees with respect to the equatorial plane³. This constellation provides the user with between five and eight satellites visible from any point on the earth.

The GPS-based navigation system is used extensively in surface vessels as these vehicles can directly receive signals radiated by the GPS. Unfortunately, these signals have a limited water-penetrating capability. Therefore to receive the signals, an antenna associated with an AUV employing a GPS system must be clear and free of water. There are three possible antenna configurations to meet this requirement. These are fixed, retractable, or expendable antennas⁴. A fixed antenna is a non-moving antenna placed on the outside of the AUV. The AUV has to surface to expose this antenna and stay surfaced until the required information has been received and processed adequately. A retractable antenna is one that the AUV would deploy while still submerged. When the required information is received, the antenna is retracted back to the AUV. The expendable antenna works along the same principle as the retractable antenna, except that it is used once and discarded. When required, another antenna would be deployed.

These antenna configurations require the AUV either to surface or to rise to a shallow depth, but there are several disadvantages⁵. For an AUV to receive radio signals, it must interrupt its mission, expend time and energy climbing and/or surfacing, risk its safety for up to a minute on the surface or in a shallow depth of water getting the fix, which is especially dangerous in a hostile environment, then expend additional time and energy submerging to resume the mission. Even if an extremely accurate fix is obtained, the vehicle location uncertainty can grow significantly during descent before the mission is ever resumed. Therefore there is a need to combine information obtained by a GPS navigation system with other underwater navigation sensors when the AUV operates underwater to maintain good navigation capability.

Optical Navigation

In the context of optical imaging for navigation, the underwater environment is a very special place. The reason for this is that, in addition to visual-sensing issues that must be addressed in land and space-based vehicles, there are also issues specific to underwater imaging. These issues include limited range of visibility, brightness and contrast variation, and non-uniform illumination⁶. Limited range of visibility is caused by the attenuation of light in water by absorption and scattering by suspended matter. Light absorption and scattering cause the amount of reflected light to exponentially decay as a function of distance to scene surfaces. The absorption and scattering of light also affect image brightness and contrast. Objects far away appear dark; as they move nearer, their brightness and contrast increase. Changes in image intensity brightness and contrast can cause many image processing techniques to fail. If some type of intensity normalisation is not performed, brightness and contrast differences between images make it difficult to realise that the same scenery or object is being viewed⁶. Non-uniform illumination refers to the limitation of artificial light sources to provide uniform illumination of the entire scene under observation. A classic example that demonstrates the difficulties non-uniform lighting can cause is the imaging of a planar, perpendicular surface using a collocated camera/light source. In this case, the image centre will appear brighter than the image border. If the camera and light source are moved relative to the scene, both the absolute and relative brightness of each pixel in the image will change. Simple effects such as these can degrade

correspondence (image matching) performance; more complicated effects such as shadowing can cause significant difficulties for most image correspondence techniques⁶.

Optical-based navigation involves the estimation of 3D motion from time varying imagery^{7,8}. Most techniques for this purpose require knowledge of relevant 2D geometric information in an image sequence. The current state-of-the-art in optical-based navigation is essentially a form of dead reckoning⁹. This method works by creating a mosaic where a series of images are taken from a video stream and aligned with each other to form a chain of images along the vehicle path. When a new image is about to be added to a mosaic, it must be properly aligned with the last image in the chain of images comprising the mosaic. To accomplish this, the two images are compared, and the displacement vector between the two image centres is calculated. Therefore, to determine the current vehicle position, it would be possible to compute the total distance travelled by summing the image displacement measurements along the image chain⁹. As with the INS discussed earlier, this method has a fundamental problem: the unbounded propagation of errors on vehicle position over time. This random walk-effect is due to the accumulation of image alignment errors as the length of mosaic increases (Fig 1). Therefore, as in the INS case, this navigation method is not normally used to provide stand-alone navigation.

Acoustic Navigation

Acoustic navigation is the most widely accepted form of AUV navigation, and a variety of systems have been both researched and tested. Most require an engineered environment, meaning that something has been added to the environment to aid navigation. The distance between acoustic baselines is generally used to define an acoustic positioning system; that is the distance between the active sensing elements. Three types of system have been primarily employed; ultra short baseline (USBL), short baseline (SBL) and long baseline (LBL) with distance between acoustic baselines less than 10 cm, between 20 to 50m and between 100 to 6000m respectively¹⁰.

USBL systems (Fig 2a) employ a single beacon on the bottom of the seafloor which emits acoustic pulses without being interrogated from an AUV. The onboard AUV equipment consists of a two-dimensional hydrophone array mounted on the bottom of the AUV. USBL systems measure the time- or phase difference of the arrival of an acoustic pulse between individual elements of the hydrophones. This time- or phase difference is used to determine the bearing from the USBL transceiver to the beacon. If a time-of-flight interrogation technique is used, a range to that beacon will also be available from the USBL system. In SBL (Fig 2b) three or more transceivers are rigidly mounted on the hull of the AUV, making either an equilateral or a right-angled triangle. The distance between each transceiver is precisely known. A bearing to the transponder is derived from the detection of the relative time-of-arrival as an acoustic pulse passes each of the transceivers. If the time-of-flight interrogation technique is used, a range to that beacon will also be available from the SBL system. Any range and bearing position derived from USBL and SBL systems are with respect to the transceivers mounted on the AUV and, as such, the systems need a vertical reference unit (VRU), a gyroscope and, possibly, a surface navigation system to provide a position that is seafloor (Earth) referenced¹⁰.

In LBL navigation systems (Fig 2c), an array of acoustic

beacons separated by a range of 100m to a few kilometres is deployed on the seabed^{10,11}. The vehicle determines its position by listening to the pulses emitted from the beacons and recording the arrival times. The location of these beacons must be provided, and the vehicle must be able to detect and distinguish between their signals. The two major types of LBL navigation are described as spherical and hyperbolic. In spherical navigation, the vehicle interrogates the array by emitting its own pulse and then listens for the responses from the beacons. In hyperbolic LBL navigation, the vehicle does not interrogate the array, but instead listens passively to the synchronised pulses emitted by the beacons¹². Any range/range position derived from a LBL system is with respect to relative or absolute seafloor co-ordinates. As such a LBL system does not require a VRU or gyroscope¹⁰.

Terrain-Relative Navigation

For some applications of AUVs, the use of acoustic beacons is undesirable or impractical. In particular, the acoustic beacons must be pre-deployed for every mission and the vehicles can operate only over relatively short ranges, and they are far too expensive to be practical in low cost civilian AUV work. Also the accuracy of the acoustic signals tend to degrade due to noise and reverberation problem. This then motivates the use of onboard terrain sensors for the purpose of navigation of an AUV. An onboard sensor is used to obtain information on the terrain surrounding the vehicle in the form of features or landmarks. The vehicle maintains a map of these landmarks which may or may not have been provided a priori. As the vehicle moves through the environment, the landmark observations obtained from the terrain sensor are matched to the landmarks maintained in the map and used, in much the same way as beacon observations, to correct and update the estimated location of the vehicle. In underwater environments it is very rare that an a priori terrain map will exist. Unlike surface applications, satellite or aircraft imagery cannot be used to build an underwater terrain map. This then precludes the common use of digital terrain elevation data (DTED) as employed by systems such as terrain contour matching (TERCOM) used for cruise missiles¹³. This limitation then motivates the development of simultaneous localisation and mapping (SLAM) for AUV navigation (see Fig 3).

SLAM is the process of concurrently building a feature-based map of the environment and using this map to obtain estimates of the location of the vehicle. In essence, the vehicle relies heavily on its ability to extract useful navigation information from the data returned by its sensors. The vehicle typically starts at an unknown location with no a priori knowledge of landmark locations. From relative observations of landmarks, it simultaneously computes an estimate of vehicle location and an estimate of landmark locations. While continuing in motion, the vehicle builds a complete map of the landmarks and uses these to provide continuous estimates of the vehicle location. By tracking the relative position between the vehicle and identifiable features in the environment, both the position of the vehicle and the position of the features can be estimated simultaneously. The SLAM algorithm has recently seen a considerable amount of interest from AUV community as a tool to enable fully autonomous navigation^{14,15,16}.

MULTISENSOR DATA FUSION

It is clear from the previous discussion that information from

sensors used in one navigation system need to be combined or fused with information from sensors of other navigation systems to improve the overall accuracy of the system. To achieve this, MSDF techniques, which combine data from multiple sensors and related information from associated databases can be used^{17,18}. Varshney¹⁹ describes MSDF as the acquisition, processing and synergistic combination of information gathered by various knowledge sources and sensors to provide a better understanding of a phenomenon. In this section, a general introduction to MSDF is provided. The description on benefits of MSDF, problems and issues, levels of MSDF where fusion takes place and MSDF algorithms are presented.

Benefits of MSDF

In general, fusion of multisensor data provides significant advantages over single source data. The advantages can be summarised as follows^{19,20}:

1. *Improved system reliability and robustness.* Multiple sensors have inherent redundancy. Due to the availability of data from multiple sensors, uncertainty can be reduced, noise can be rejected and sensor failure can be tolerated.
2. *Extended coverage.* An increase in both spatial and temporal coverage of an observation is made possible by the use of multiple sensor systems. Multiple sensors can observe a region larger than the one observable by a single sensor.
3. *Increased confidence.* Joint data from multiple sensors confirm the set of hypotheses about an object or event. The confirmation can be used to exclude some hypotheses to produce a reduced set of feasible options and as a result reduce the effort required to search for the best solution.
4. *Enhanced resolution.* Multiple sensors with different resolution can result in a greater resolution than a single sensor can achieve.

Problems and Issues

A technique for MSDF should consider several key issues, summarised below^{19,20}:

1. *Registration/data alignment.* Each sensor provides data in its local frame. The data from different sensors must be converted into a common reference frame before combination. This problem of aligning sensor reference frames is often referred to as a registration problem.
2. *Correspondence/data association.* Once the sensors are registered, there is still a need to establish which data features in one sensor refer to the same aspect environment of the sensor.
3. *Fusion.* The fusion of data from multiple sensors or a single sensor over time can take place at different levels of representation. A useful categorisation is to consider MSDF as taking place at signal-, pixel-, feature- and symbol levels of representation.
4. *Inference and estimation.* Once the data has been fused, it is necessary to infer the sensed data due to the inherent uncertainty in the combined measurements.
5. *Sensor Management.* Sensor management can take the form of active data gathering where the sensors are directed via feedback to specific fusion stage, physical reconfiguration of the spatial pattern of the sensors and sensor type, or algorithmic changes to the combination of data.

Levels of MSDF

The common fused representation may range from a low-level probability distribution for statistical inference to high level

logical proposition used in production rules for logical inference. Luo and Kay²² and Luo et al.²³ divide the levels of representation of MSDF into signal-, pixel-, feature- and symbol levels.

1. *Signal-level.* Signal-level fusion deals with the combination of signals from a group of similar sensors with the aim of deriving a single composite signal, usually of the same form as the original signals but with a higher quality. The signals produced by the sensors can be modelled as random variables corrupted by uncorrelated noise, with the fusion process considered as an estimation procedure. A high degree of spatial and temporal registration between the sensed data is necessary for fusion to take place.

2. *Pixel-level.* Pixel-level fusion deals with the combination of multiple images into a single image with a greater information content. The fused images can be modelled as a realisation of a stochastic process across the image, with the fusion process considered as an estimation procedure. In order for pixel-level to be feasible, the data provided by each sensor must be able to be registered at the pixel-level and, in most cases, must be sufficiently similar in terms of its resolution and information content.

3. *Feature-level.* Feature-level fusion deals with the combination of features derived from signals and images into meaningful internal representations or more reliable features. A feature provides for data abstraction and is created either through the attachment of some type of semantic meaning to the results of the processing of some spatial and/or temporal segment of the sensory data or through a combination of existing features. As compared to the signal- and pixel-level fusion, the sensor registration requirements for feature-level fusion are less stringent, with the result that the sensors can be distributed across different platform.

4. *Symbol-level.* Symbol-level fusion deals with the combination of symbols with an associated uncertainty measure, each representing some decision, into symbols representing composite decisions. A symbol derived from sensory information represents a decision that has been made concerning some aspect of the environment. The decision is usually made by matching features derived from the sensory information to a model. The sensor registration is usually not explicitly considered in symbol-level fusion because the spatial and temporal extent of the sensory information upon which a symbol is based has already been explicitly considered in the generation of the symbol.

MSDF Algorithms

This section presents fusion algorithms for MSDF. Luo et al.²³ classify MSDF algorithms as follows: estimation methods, classification methods, inference methods and artificial intelligence methods. Each of these methods will be discussed here and applications to AUV navigation are presented later.

1. *Estimation methods.* A general estimation method of fusion is to take a weighted average of redundant information provided by a group of sensors and use this as the fused value. While this method provides real-time processing capability of dynamic low-level data, the Kalman filter is generally preferred as it provides a method that is nearly equal in processing requirement and results in estimates for the fused data that are optimal in a statistical sense. Kalman filtering is an estimation method that combines all available measurement data, plus prior knowledge about the system and measuring devices, to produce an estimate of the state in such a manner as to minimise the error statistically²⁴. A detailed formulation of Kalman filter is given in

appendix A.

2. Classification methods. Classification methods involve partitioning of the multidimensional feature space (by geometrical or statistical boundaries) into distinct regions, each representing an identity class. In this method, the location of a feature vector to prespecified locations in feature space is compared. A similarity measure must be computed and each observation is compared to a priori classes. In the cluster analysis approach, geometrical relationships on a set of sample data in a training process are established²⁵. Other approaches include unsupervised or self-organised learning algorithms such as K-means clustering and the associated adaptive update rule, the Kohonen feature map²⁶. To fuse sensory data in an adaptive manner and allow to automatically adjust the granularity of the classifier and to maintain stability against category proliferation in the presence of drifting inputs and changing environments, ART, ARTMAP and Fuzzy ART network approaches can be used.

3. Inference methods. Bayesian inference and Dempster-Shafer evidential reasoning are the main approaches in inference methods. Bayesian inference provides formalism for MSDF that allows sensory data to be fused according to the rules of probability theory. This approach relies on the use of Bayes' rule where a relationship between the a priori probability of a hypothesis, the conditional probability of an observation given a hypothesis and the a posteriori probability of the hypothesis is provided¹⁸. An immediate problem in this approach is that the required knowledge of the a priori probability and the conditional probability may not be always available. Also in defining these probabilities, often subjective judgements are necessary²⁷. An extension to the Bayesian inference method, Dempster-Shafer evidential reasoning, overcomes these drawbacks by keeping track of an explicit probabilistic measure of the lack of information concerning a proposition's probability. The cost of this approach is the additional time required for computation.

4. Artificial intelligence methods. Artificial intelligence is a vast, loosely defined area encompassing various aspects of pattern recognition and image processing, natural language and speech processing, automated reasoning and a host of other disciplines. Fuzzy logic and neural network are two of the most widely used approaches in artificial intelligence methods for combining multisensor data. Fuzzy logic involves extension of Boolean set theory and Boolean logic to a continuous-valued logic via the concept of membership functions to quantify imprecise concepts. Neural network is a method designed to mimic a theory of how biological nervous systems work. In this method, an individual neuron takes weighted input from a number of sources, perform a simple function and then produces a single output when the required threshold is reached. Neurons can be trained to represent sensor data and, through associate recall, complex combinations of the neurons can be activated in response to different sensor stimuli²³.

APPLICATIONS OF MULTISENSOR DATA FUSION

The discussion here focuses on a variety of approaches to the fusion of information from combinations of different types of sensors.

Inertial and GPS-Based Systems

McGhee et al²⁸ describe a navigation system employed by the

Phoenix AUV using an inertial and differential GPS (DGPS) navigational suite to conduct shallow-water mine-detection and coastal environment monitoring missions. In the course of its mission, Phoenix combines signal-level information from a gyroscope, depth sensor, speed sensor, and a compass heading to predict its position while operating underwater. The vehicle surfaces periodically to obtain an update of its position from a DGPS fix and then submerges (Fig 4a). Problems with this setup concern the time required to acquire the DGPS data and the influence of water covering the DGPS antenna during position fixing were examined in Norton²⁹. The inertial navigation sensors described in McGhee et al²⁸ obtain accelerations and angular rates of change for the vehicle. A 'nine state' Kalman filter is used to process the data and to give the prediction of the vehicle position. The DGPS data is then used to update the predicted position resulting in an estimated position. The nine state Kalman filter can be divided into seven continuous-time states (three Euler angles, two horizontal velocities, and two horizontal positions) and two discrete-time states (estimated east and north current derived from the DGPS fixes). The method used to fuse sensory information discussed by McGhee et al²⁸ can be shown as in Fig 4b.

The main problem with the Kalman filter employed in McGhee et al²⁸ is the need for a tuning system to prevent filter divergence. This problem can be overcome by the use of artificial intelligence (AI) techniques as have been applied in helicopters³⁰, automobiles³¹ and target tracking system³² applications. Kobayashi et al³¹ wished to determine accurately the position of an automobile using DGPS. In their work, a fixed fuzzy rule based algorithm is used to tune the covariance factors of a Kalman filter. The shape and positioning of the various fuzzy sets on their respective universes of discourse having been decided by heuristic means. The main problem with the Kobayashi et al³¹ methodology is the reliance on trial and error to generate the fuzzy rule based algorithms. Similar comments can also be made concerning the robot positioning work of Jetto et al³³. To overcome such drawbacks genetic algorithms^{34,35} have been used to optimise fuzzy systems. Other intelligent optimisation techniques such as chemotaxis, alopex and simulated annealing have also been successfully employed in the design optimisation of fuzzy control systems^{36,37}.

Acoustic-Based Systems

Atwood et al³⁸ have built and tested an AUV that utilises a LBL navigation system with an innovative fix-finding algorithm and commercially-available hardware. They use a spherical navigation system, in which the vehicle actively interrogates acoustic transponders and calculates ranges from round trip transit times, resulting in a greater accuracy (about 1m) compared to the hyperbolic method proposed by Bellingham et al¹². In this system, the vehicle can use two operating modes, master mode and transponder mode. In the first mode, the vehicle triggers the acoustic transponders, which reply with an acoustic signal. The vehicle computer can then calculate distances and, applying acoustically measured depth, a position. Using the first mode, operation over an area of 1 km² is possible. In the second operating mode, a surface vessel triggers the vehicle, which in turn interrogates the transponders. Position of the AUV can then be calculated in the surface vessel through an established GPS position and knowledge of the relative positions of the AUV and the transponders. This procedure is called the *fish solution*, as it lets the operator on the ship monitor vehicle progress. The second mode is developed to have operational areas as large as 10 km².

In this work, Atwood et al³⁸ have solved the problem of fading or destructive interference of the acoustic signals produced by the transponders encountered by Bellingham et al³⁹. Atwood et al³⁸ principally combine sensor information at signal-level data.

Rendas and Lourtie⁴⁰ combine LBL navigation with dead reckoning and calls it a *hybrid system*. The vehicle travels between deployed baseline arrays, each consisting, for example, of four transponders, and uses acoustic navigation when in range of an array. Outside the range, it uses a sonar/Doppler sensor and depth information for autonomous navigation. The distances between the arrays must be carefully planned, because the accuracy of navigation in the autonomous mode deteriorates with time, depending on the quality of the sensing systems. The transition from one mode to another takes place automatically. When the vehicle is leaving the area where a particular baseline array is located, the number of range measurements it is able to receive will gradually decrease to zero, entering, in this way, the autonomous navigation mode. On the contrary, when it approaches an area where transponders are located, it receives an increasing number of distance measurements, switching from autonomous to local navigation mode. The system uses a variable dimension Kalman filter for both navigation modes. Where there is no detectable acceleration, the filter assumes uniform motion and estimates position and linear velocity. When there is acceleration, the filter switches to a larger order (manoeuvring model) and extends its state vector to include the accelerations. In this work, however, Rendas and Lourtie⁴⁰ have not taken into account the analytical approximations to the error evolution during autonomous navigation to determine the layout of the baseline arrays and to derive the constraints on path planning once a layout has been decided upon. Similar to Bellingham et al³⁸, the MSDF method used by Rendas and Lourtie⁴⁰ is an estimation method which fuses data from the navigation sensors at signal-level.

Acoustic- and Optical-Based Systems

Majumder et al^{14,15,16} reported the use of sonar and underwater cameras to construct a complete environmental map for navigation. A generic, multi-layered data fusion scheme is used to combine information from the two sensors. The general principle is that all sensor information is projected into a common state-space before the extraction of seabed features. Once projection has occurred, feature extraction and subsequent processing is based on a combined description of the environment. As robust features, such as points and lines turn out to be fragile in a natural underwater environment, Majumder et al found that this approach is better than extracting features from a single piece of sensor information followed by fusion. In this work, 'blobs' and blob-like patches are used as scene descriptors to segregate feature information from background noise and other errors. Majumder et al discussed both the Bayesian and extended Kalman filter (EKF) approaches to map-building and localisation in autonomous navigation systems. It was shown in this work that a significant problem in applying EKF is the difficulty of modelling natural environment features in a form that can be used in an EKF algorithm. Another formidable problem is the fragility of the EKF method when faced with incorrect associations of observations to landmarks. The limitations in using EKF to build a feature map of landmarks describing the environment were then resolved through the use of the Bayesian approach. The fusion process can be shown as in Fig 5. A significant problem with this approach lies on the stability of the algorithm when the vehicle is run over long distances and returning around a loop to the initial

vehicle location. This problem stems from the limitation in data association technique to correspond initially identified landmarks and the same landmarks viewed from the opposite side on the return visit. A potential solution to this problem is to use a probabilistic model to provide a very general description of landmarks form and shape.

Twin Burger 2, an AUV developed by the University of Tokyo, was designed to help monitor and carry out routine maintenance work of underwater cables^{41,42}. In doing so, the vehicle tracks the cable visually and provides human operators with visual information about the condition of the cable accordingly. Initially the vehicle employed a visual servoing system to track the cable and to navigate the AUV accordingly. However, due to undesirable optical behaviour underwater, there were many occasions where the cable was not sufficiently visible for the vision processor to track the cable. In addition, the vehicle can lose track of the cable when there were many similar cables appearing in the image. In order to overcome these problems, a multisensor fusion technique is proposed. The proposed sensor fusion technique uses dead reckoning position uncertainty with a 2D-position model of the cable to predict the region of interest in the image captured by a camera mounted on the AUV^{41,42}. The 2D-position model of the layout of the cable is generated by taking the position (x_i, y_i) of a few points along the cable. The 2D-position model of the cable is used to predict the most likely region of the cable in the image, which leads to a reduction in the amount of image data and a decrease in the image processing time. Additionally, due to the narrowing of the region of interest in the image, the chances of misinterpretation of similar features appearing in the image can be avoided. The 2D-position model is also used to generate navigation commands when the vision processor cannot recognise the cable in the environment. Similar to Majumder et al^{14,15,16}, the fusion process takes place at feature-level.

Scheizer⁴³ has reported a target detection and classification system using side scan sonar data and vision. Objects are detected by searching for highlights, textures, statistical anomalies and shadows. An artificial neural network-based classification system is used to assist the image-processing component. The classification process does not identify objects but rather labels them as foreground, background, highlight, or shadow highlight. The level of correct classification is reported to be 95% using a training set of 62 images. This technique, however, does not address the issue of feature- or object identification.

CONCLUDING REMARKS

It has been suggested in this paper, from the various examples given in AUV navigation, that information coming from a single navigation system is not sufficient to provide a good navigation capability. Therefore MSDF techniques which combine sensory information from other navigation systems to improve the navigation capability is essential. MSDF techniques which combine sensory information from inertial, radio and optical navigation system to track underwater cables is currently being developed in a three year co-operative project funded by EPSRC involving both the University of Plymouth and Cranfield University, UK. The navigation system that is being developed at the University of Plymouth utilises INS/GPS and will be enhanced by a vision-based navigation system being developed at Cranfield University.

During an underwater cable tracking mission, the position obtained by the INS will be combined with a dead reckoned position obtained from a vision-based navigation system. In

addition, the 2D-position model of the cable can also be included to predict the most likely region of the cable in the image, which leads to a reduction in the amount of image data and a decrease in the image processing time as discussed in^{41,42}. The vision system will be based on the laser stripe illumination methodology previously developed at Cranfield University⁴⁴, overcoming optical imaging problems such as range of visibility, brightness and contrast, and illumination of the seabed as discussed by Marks et al⁶. The combined positions are then used to identify the locations of the gathered images of the cable.

A significant improvement in accuracy of the navigation system is expected as the accumulated error using the combined INS and video-based navigation is to be minimised by the GPS update methodology as discussed by McGhee et al²⁸ and also as the Kalman filter will be made adaptive by using fuzzy logic techniques to prevent the filter divergence as discussed by Kobayashi et al³¹.

ACKNOWLEDGEMENT

This research has been supported by the Engineering and Physical Sciences Research Council (EPSRC), QinetiQ (Winfrith and Bingley), J&S Marine, South West Water PLC and Halliburton Subsea.

REFERENCES

1. Cox, R, and Wei, S. (1995). *Advances in the State of the Art for AUV Inertial Sensors and Navigation Systems*. IEEE Journal of Oceanic Engineering, 20 (4), pp. 361-366.
2. Bellingham, JG. (1992). *Capabilities of Autonomous Underwater Vehicles*. In J. Moore., Editor, Scientific and Environmental Collection with Autonomous Underwater Vehicles, pp. 7-14. MIT Sea Grant.
3. Ellowitz, HJ. (1992). *The Global Positioning System*. Microwave Journal, 35 (4), pp. 24-33.
4. Kwak, SH, McKeon, JB, Clynych, JR, and McGhee, RB. (1992). *Incorporation of Global Positioning System into Autonomous Underwater Vehicle Navigation*. Proceedings of the 1992 Symposium on Autonomous Underwater Vehicle Technology, Washington, DC, pp. 291-297.
5. Kwak, SH, Stevens, CD, Clynych, JR, McGhee, RB, and Whalen, RH. (1993). *An Experimental Investigation of GPS/INS Integration for Small AUV Navigation*. Proceedings of International Symposium on Unmanned Unthethered Submersible Technology, Durham, NH, pp. 239-251.
6. Marks, RL, Rock, SM, and Lee, MJ. (1994). *Real-Time Video Mosaicking of the Ocean Floor*. IEEE Journal of Oceanic Engineering, 20 (3), pp. 229-241.
7. Victor, JS, and Senteiro, J. (1994). *The Role of Vision for Underwater Vehicles*. Proceedings of the 1994 Symposium on Autonomous Underwater Technology, Cambridge, MA, pp. 28-35.
8. Hallset, JO. (1992). *A Vision System for an Autonomous Underwater Vehicle*. Proceedings of the 11th IAPR International Conference on Pattern Recognition, Los Alamitos, CA, pp. 320-323.
9. Huster, A, Fleischer, SD, and Rock, SM. (1998). *Demonstration of a Vision-Based Dead-Reckoning System for Navigation of an Underwater Vehicle*. Proceedings of the Oceans '98 MTS/IEEE, Nice, France, pp. 326-330.
10. Vickery, K. (1998). *Acoustic Positioning Systems: A Practical Overview of Current Systems*. Proceedings of the 1998 Workshop on Autonomous Underwater Vehicle, Cambridge, MA, pp. 5-17.
11. Geyer, EM, Creamer, PM, D'Appolito, JA, and Gains RG. (1987). *Characteristics and Capabilities of Navigation Systems for Unmanned Unthethered Submersibles*. Proceedings International Symposium on Unmanned Unthethered Submersible Technology, Durham, NH, pp. 320-347.
12. Bellingham, JG, Consi, TR, Tedrow, U, and Di Massa, D. (1992). *Hyperbolic Acoustic Navigation for Underwater Vehicles: Implementation and Demonstration*. 1992 IEEE Symposium on Autonomous Underwater Vehicle Technology, Washington, DC, pp. 304-309.
13. Golden, JP. (1980). *Terrain Contour Matching (TERCOM): A Cruise Missile Guidance Aid*. SPIE: Image Processing for Missile Guidance, 238, pp. 10-18.
14. Majumder, S, Scheduling, S, and Durrant-Whyte, HF. (2000). *Sensor Fusion and Map Building Localisation for Underwater Navigation*. Proceedings of the 7th International Symposium on Experimental Robotics, Honolulu, Hawaii.
15. Majumder, S, Scheduling, S, and Durrant-Whyte, HF. (2000). *Sensor Fusion and Map Building for Underwater Navigation*. Proceedings of the Australian Conference on Robotics and Automation, Melbourne, Australia, pp. 25-30.
16. Majumder, S, Scheduling, S, and Durrant-Whyte, HF. (2001). *Multisensor Data Fusion for Underwater Navigation*. Robotics and Autonomous Systems 35, pp. 97-108.
17. Llinas, J, and Waltz, E. (1990). *Multisensor Data Fusion*. Artech House: Boston, MA.
18. Hall, D. (1992). *Mathematical Techniques in Multisensor Data Fusion*. Artech House: Boston, MA.
19. Varshney, PK. (1997). *Multisensor Data Fusion*. Electronics and Communication Engineering Journal, 9 (6), pp. 245-253.
20. Harris, CJ, Bailey, A, and Dodd, TJ. (1998). *Multisensor Data Fusion in Defence and Aerospace*. The Aeronautical Journal, 102 (1015), pp. 229-241.
21. Hall, DL, and Llinas, J. (1997). *An Introduction to Multisensor Data Fusion*. Proceedings of the IEEE, 85 (1), pp. 6-23.
22. Luo, RC, and Kay, MG. (1990). *A Tutorial on Multisensor Integration and Fusion*. Proceedings of 16 Annual Conference of IEEE of Industrial Electronics Society, Pacific Grove, CA, pp. 707-722.
23. Luo, RC, Chih-Chen, Y, and Kuo, LS. (2002). *Multisensor Fusion and Integration: Approaches, Application and Future Research Directions*. IEEE Sensors Journal, 2 (2), pp. 107-119.
24. Brown, P, and Hwang, P. (1997). *Introduction to Random Signals and Applied Kalman Filtering*. Wiley & Sons, Inc.: Toronto, Canada.
25. Bracio, BR, Horn, W, and Moller, DPF. *Sensor Fusion in Biomedical Systems*. (1997). Proceedings of the 19th Annual International Conference of IEEE Engineering in Medicine and Biology and Society, Chicago, IL, pp. 1387-1390.
26. Kohonen, T. (1988). *Self-Organization and Associative Memory*. Springer-Verlag: New York.
27. Brooks, RR, and Iyengar, SS. (1998). *Multisensor Fusion: Fundamentals and Applications with Software*. Prentice Hall: Upper Saddle River, NJ.
28. McGhee, RB, Clynych, JR, Healey, AJ, Kwak, SH, Brutzman, DP, Yun, XP, Norton, NA, Whalen, RH, Bachmann, ER, Gay, DL, and Schubert, WR. (1995). *An Experimental Study of an Integrated GPS/INS System for Shallow-Water AUV Navigation*. 9th International Symposium on Unmanned Unthethered Submersible Technology, Durham, NH, pp. 153-167.
29. Norton, NA. (1994). *Evaluation of Hardware and Software for a Small Autonomous Underwater Vehicle Navigation System (SANS)*.

Master Thesis, Naval Postgraduate School, Monterey, California.

30. Doyle, RS, and Harris, CJ. (1996). *Multisensor Data Fusion for Helicopter Guidance Using Neurofuzzy Estimation Algorithms*. The Royal Aeronautical Society Journal, June, pp. 241-251.

31. Kobayashi, K, Cheok, KC, Watanabe, K, and Munekata, F. (1998). *Accurate Differential Global Positioning via Fuzzy Logic Kalman Filter Sensor Fusion Technique*. IEEE Transaction on Industrial Electronics, 45 (3), pp. 510-518.

32. McGinnity, S, and Irwin, G. (1997). *Nonlinear State Estimation Using Fuzzy Local Linear Models*. International Journal of Systems Science, 28 (7), pp. 643-656.

33. Jetto, L, Longhi, S., and Vitali, D. (1999). *Localisation of a Wheeled Mobile Robot by Sensor Data Fusion Based on a Fuzzy Logic Adapted Kalman Filter*. Control Engineering Practice, 7, pp. 763-771.

34. Pham, DT, and Karaboga, D. (1991). *Optimum Design of Fuzzy Controllers Using Genetic Algorithms*. Journal of Systems Engineering, pp. 114-118.

35. Sutton, R, and Marsden, GD. (1997). *A Fuzzy Autopilot Optimised Using a Genetic Algorithm*. Journal of Navigation, 50, pp. 120-131.

36. Sutton, R, Taylor, SDH, and Roberts, GN. (1996). *Neurofuzzy Techniques Applied to a Ship Autopilot Design*. Journal of Navigation, 49, pp. 410-429.

37. Sutton, R, Taylor, SDH, and Roberts, GN. (1997). *Tuning Fuzzy Ship Autopilots Using Artificial Neural Networks*. Transaction of Institute of Measurement and Control, 19 (2), pp. 94-106.

38. Atwood, DK, Leonard, JJ, Bellingham, JG, and Moran, BA. (1995). *An Acoustic Navigation system for Multi-Vehicle Operations*. 9th International Symposium on Unmanned Unthetered Submersible Technology, Durham, NH, pp. 202-208.

39. Bellingham, JG, Goudey, CA, Consi, TR, Bales, JW, Atwood, DK, Leonard, JJ, and Chrysosostomidis, (1994). *A Second Generation Survey AUV*. Proceedings of the 1994 Symposium on Autonomous Underwater Vehicle Technology, Cambridge, MA, pp. 148-155.

40. Rendas, MJ, and Lourtie, IMG. (1994). *Hybrid Navigation System for Long Range Navigation*. Proceedings of the 1994 Symposium on Autonomous Underwater Vehicle Technology, Cambridge, MA, pp. 353-359.

41. Balasuriya, BAAP, and Ura, T. (1999). *Multisensor Fusion for Autonomous Underwater Cable Tracking*. Oceans'99 MTS/IEEE, Seattle, WA, pp. 209-215.

42. Balasuriya, BAAP, and Ura, T. (1999). *Sensor Fusion Technique for Cable Following by Autonomous Underwater Vehicles*. Proceedings of the 1999 IEEE International Conference on Control Applications, Kohala Coast-Island of Hawaii, Hawaii, pp. 1779-1784.

43. Scheizer, PF, and Petlevitch, EJ. (1989). *Automatic Target Detection and Cueing System for an Autonomous Underwater Vehicle*. Proceedings 7th International Symposium Unmanned Unthetered Submersible Technology, Durham, NH, pp. 359-371.

44. Tetlow, S, and Allwood, RL. (1995). *Development and Applications of a Novel Underwater Laser Illumination System*. Underwater Technology, 21 (2), pp. 13-20.

APPENDIX A

The Kalman filter and the extended Kalman filter are the most popular tools proposed in the literature for MSDF in AUV navigation. If the AUV system can be described with a linear model and both the system and sensor error can be modelled as white Gaussian noise, a Kalman filter provides unique, statistically optimal, estimates for data of interest. In the Kalman filter formulation, the observation $z(k) \in \mathcal{R}^n$ are described (or approximated) by the linear model

$$z(k+1) = H(k+1)x(k+1) + v(k+1) \quad (1)$$

where $x \in \mathcal{R}^m$ is a state vector, $H \in \mathcal{R}^{n \times m}$ is an observation model, and $v \in \mathcal{R}^n$ is the observation noise. The state vector satisfies a linear discrete-time state transition equation

$$x(k+1) = F(k+1)x(k) + G(k+1)u(k+1) + w(k+1) \quad (2)$$

where $F \in \mathcal{R}^{m \times m}$ is the system model, $G \in \mathcal{R}^{m \times q}$ is the control model, $u \in \mathcal{R}^q$ is a known control input, and $w \in \mathcal{R}^m$ is the input noise.

Independent, zero mean and white noise processes are assumed,

$$\begin{aligned} E[w(k)] &= E[v(k)] = 0, E[w(k)w^T(j)] = Q(k)\delta_{kj}, \\ E[v(k)v^T(j)] &= R(k)\delta_{kj}, E[w(k)v^T(j)] = 0 \end{aligned} \quad (3)$$

where δ_{kj} is the Kronecker delta function

$$(\delta_{kj} = 0, k \neq j; \delta_{kj} = 1, k = j).$$

The optimal mean square error estimate of $x(k)$ given $z(1), \dots, z(j) (k \geq j)$ is

$$\hat{x}(k|j) = E[x(k)|z(1), \dots, z(j)] \quad (4)$$

and the conditional covariance matrix of $\hat{x}(k|j)$ is

$$P(k|j) = E[(x(k) - \hat{x}(k|j))(x(k) - \hat{x}(k|j))^T | z(1), \dots, z(j)] \quad (5)$$

The Kalman filter algorithm provides recursively an estimate $\hat{x}(k+1|k+1)$ in terms of the previous estimate $\hat{x}(k|k)$ and the most recent observation, $z(k+1)$. It involves a cycle of prediction and updating (see ref 27).

The measurement model for the EKF is

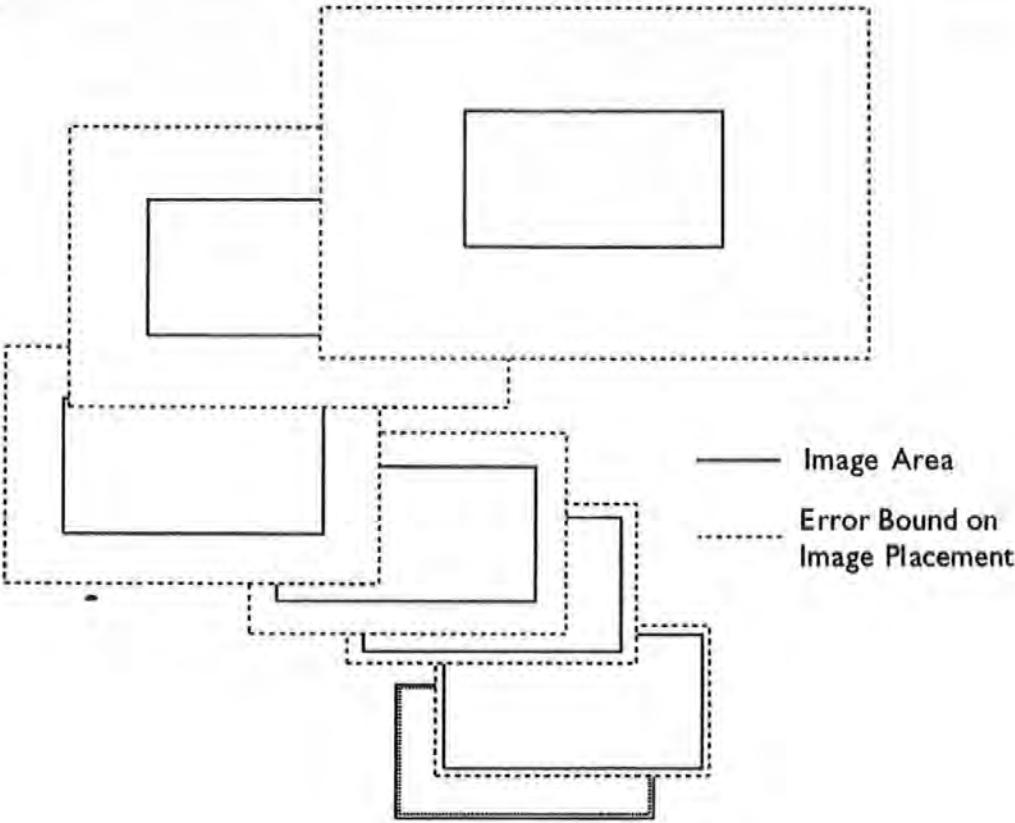
$$z(k+1) = h[k+1, x(k+1)] + v(k+1), \quad (6)$$

and the dynamics are assumed to be

$$x(k+1) = f(k+1, x(k), u(k+1)) + w(k+1) \quad (7)$$

The vector-valued function h and f are, in general, time varying. The EKF framework is developed through a series expansion of the nonlinear dynamics and the measurement equation.

FIGURES



As the length of the image chain comprising the mosaic increases, the error bound in placing the last image relative to the initial image (i.e. the origin) continues to grow according to a random walk

Fig 1: Error propagation in image chain as described by Huster et al.⁹

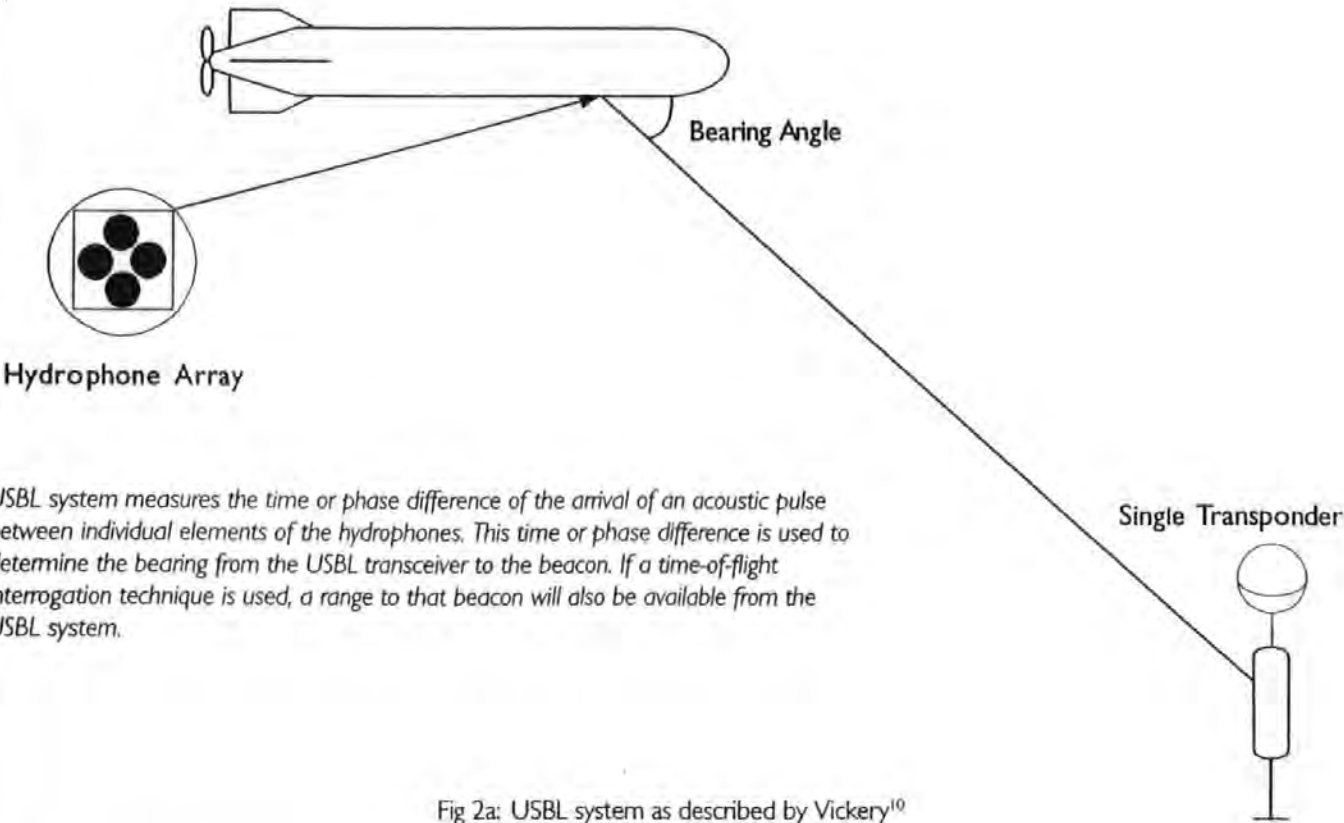
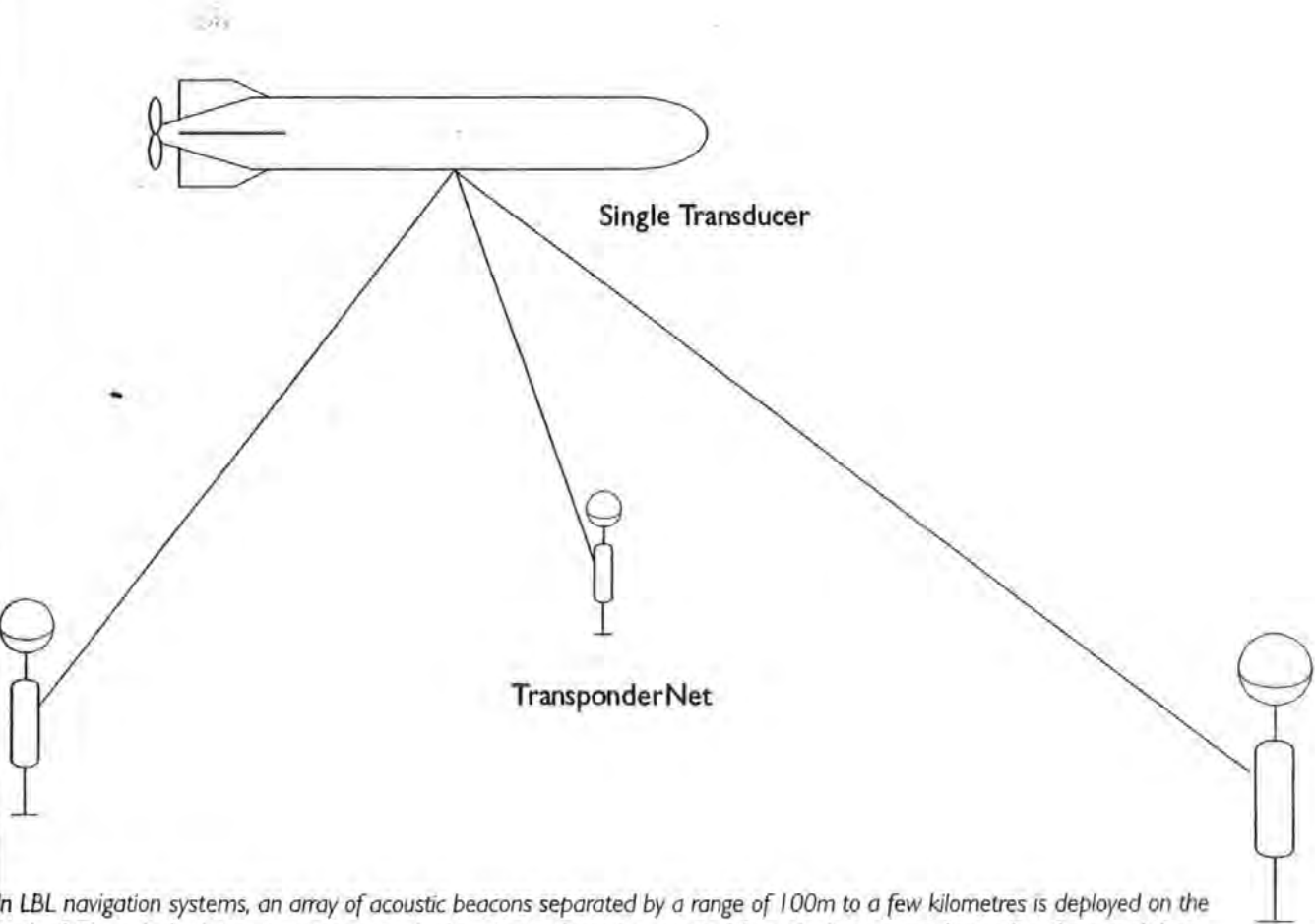
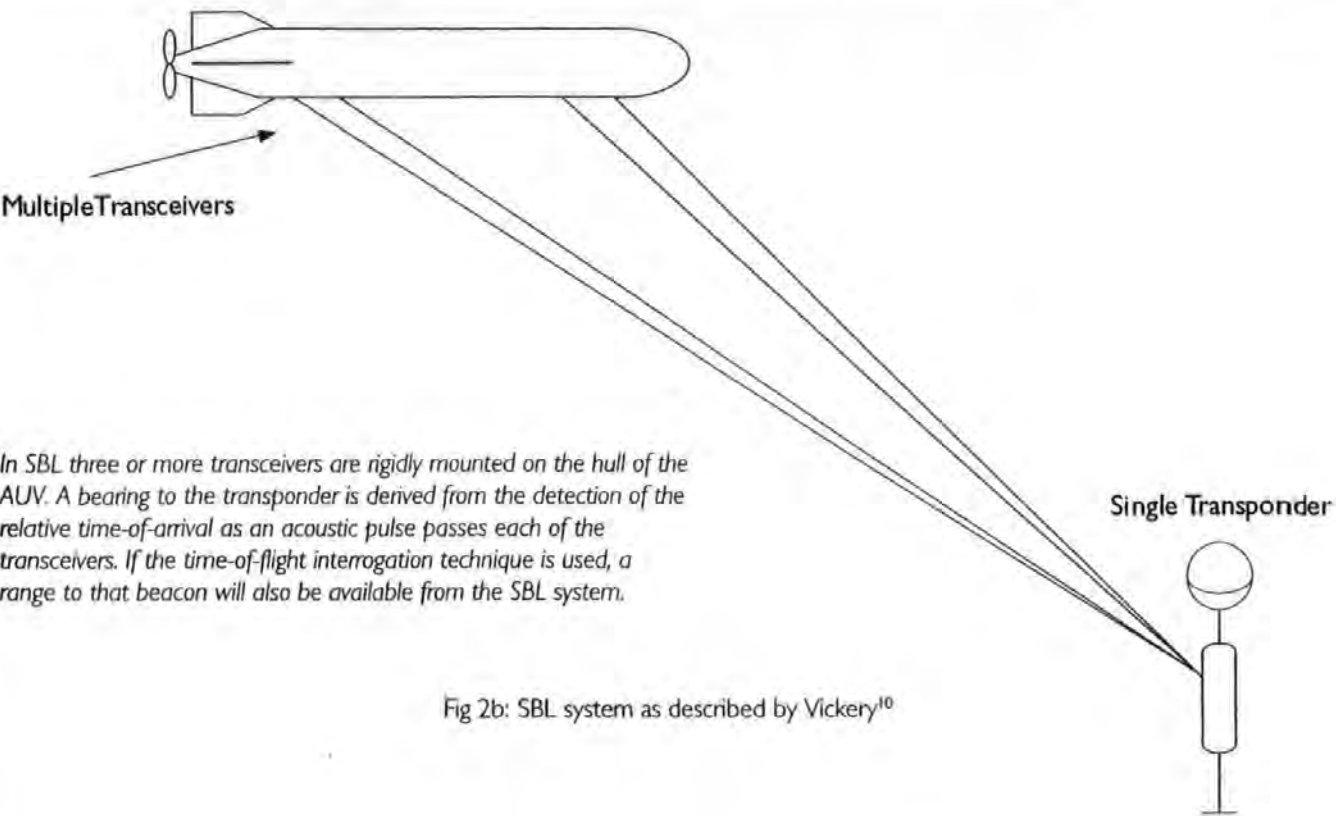
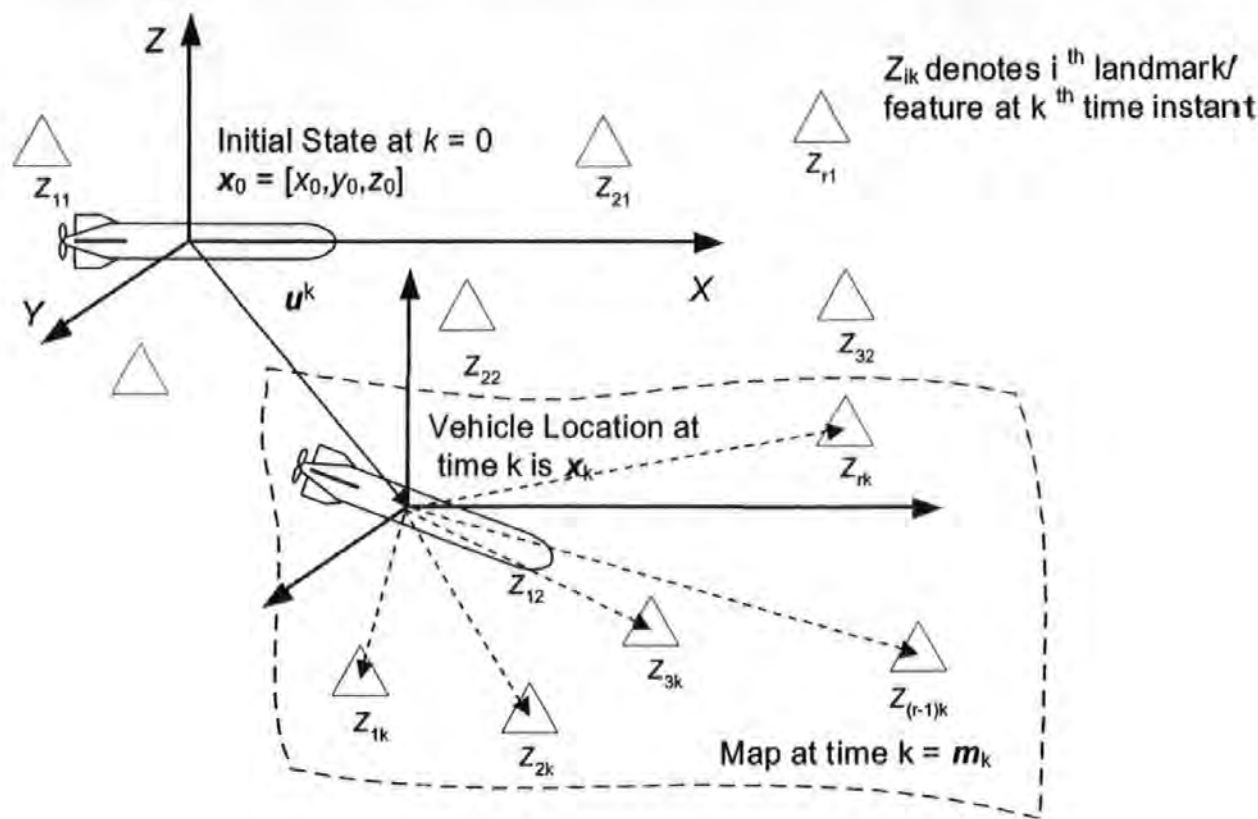


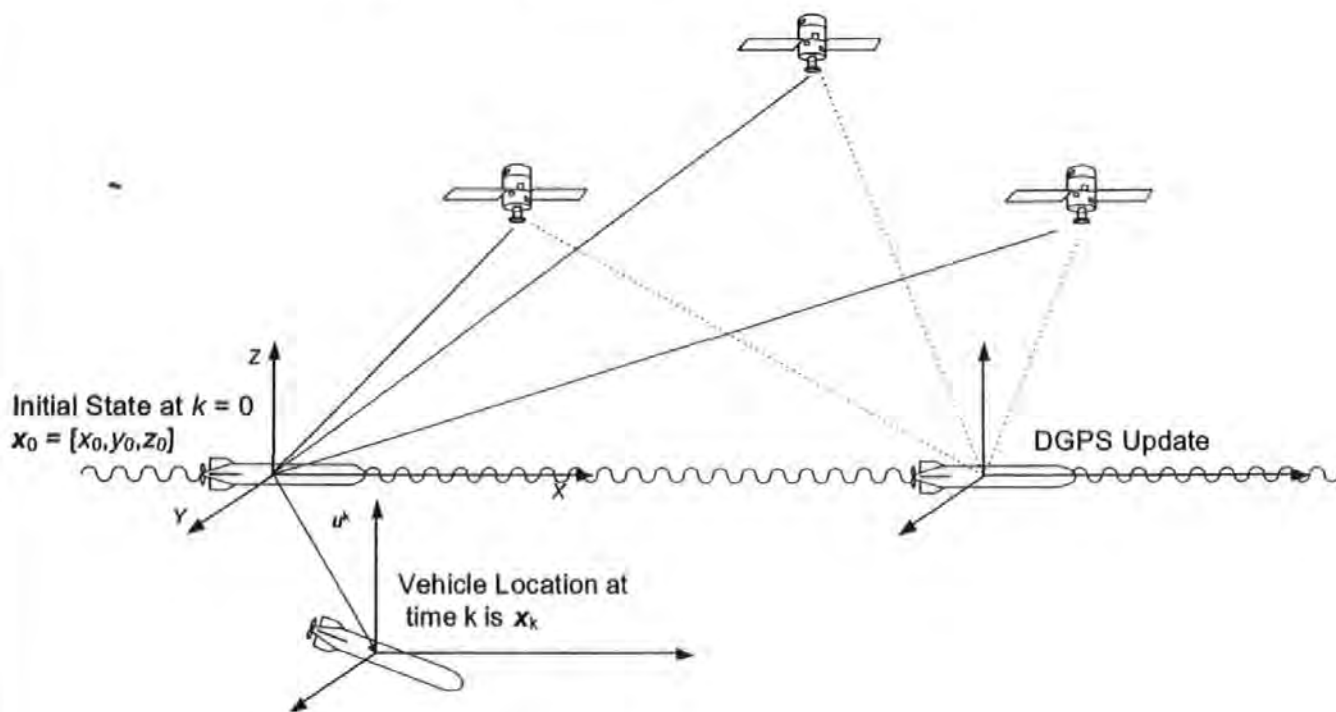
Fig 2a: USBL system as described by Vickery¹⁰





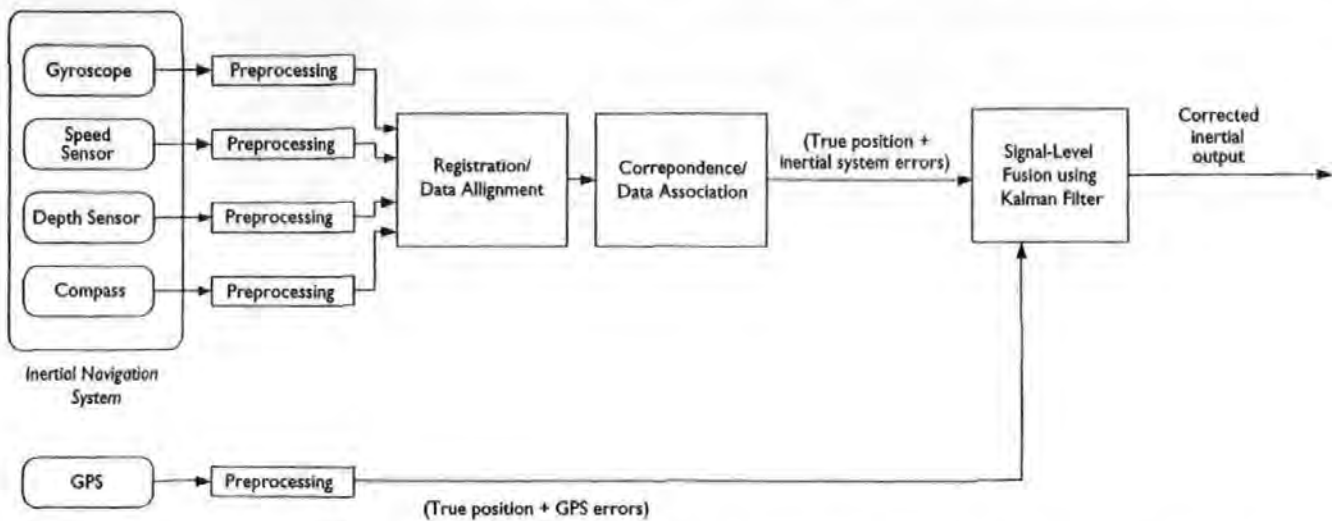
Relationship between the vehicle, features and map at any time k is shown above. Cartesian axes system is used to describe the vehicle location at any time k denoted by x_k . The vehicle states change as a result of the applied control input u_k . The map at any time k is defined as set of landmarks or features detected from the sensor observation z_k relative to the vehicle location.

Fig 3: SLAM algorithm as described by Majumder et al.^{14,15,16}



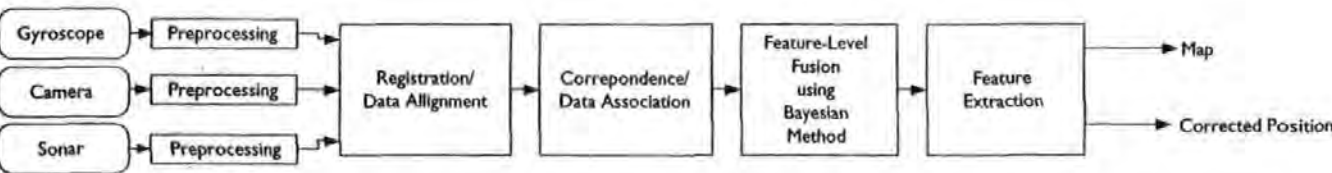
In the course of its mission Phoenix combines signal-level information INS to predict its position while operating underwater. The vehicle surfaces periodically to obtain an update of its position from a DGPS fix and then submerges.

Fig 4a: GPS/INS Navigation by McGhee et al.²⁸



In the course of its mission Phoenix combines signal-level information from a gyroscope, depth sensor, speed sensor, and a compass heading to predict its position while operating underwater. The vehicle surfaces periodically to obtain an update of its position from a DGPS fix and then submerges to resume its mission.

Fig 4b: MSDF in GPS/INS Navigation by McGhee et al.²⁸



The general principle is that all sensor information is projected into a common state-space before the extraction of seabed features. Once projection has occurred, feature extraction and subsequent processing is based on a combined description of the environment.

Fig 5: MSDF in SLAM algorithm as described by Majumder et al.^{14,15,16}

APPENDIX B

THE *HAMMERHEAD* STATE SPACE MODEL PROPERTIES

The properties of nine different *Hammerhead* state space models are shown in Figure B.1 - B.9. These include the fit between simulated and measured output, model residuals, step response, impulse response and pole-zero map. In this thesis, second order model is deemed to be sufficient for the analysis of the proposed MSDF techniques. This is based on the fact that the difference between the fit (between simulated and measured output) of second order system and higher order systems is infinitesimal. Autocorrelation function of the residual and the cross correlation between the residual and the input are also computed and displayed. The 95% confidence interval of these values are shown by the dashed curves. It is evident by visual inspection of Figure B.1(b), B.2(b) and B.3(b), that the second order model could produce an autocorrelation function that goes inside the confidence interval better than the third and fourth order models do. Further comparison between the second order model and the fifth to the tenth order models shows that more autocorrelation of these models lie inside the 95% confidence intervals. However, the use of these models could increase the level of complexity in the development of the proposed algorithm. This is also one of the reason behind the preference to use the second order model over the higher order models.

Further analysis can be made by observing the step response of these models in conjunction with their pole-zero plots. The step response, which are shown in Figure B.1(c)-B.9(c), displays a particular behaviour, which generally belongs to an integrator type of system where the output is produced by integrating the input. This behaviour is mainly caused by the presence of the poles at the axis of the unit circle ($z = 1$ or equivalent to $s = 0$), which along with the step input will form a ramp function, whose output magnitude can increase without bound as time progresses.

B.1 SECOND ORDER

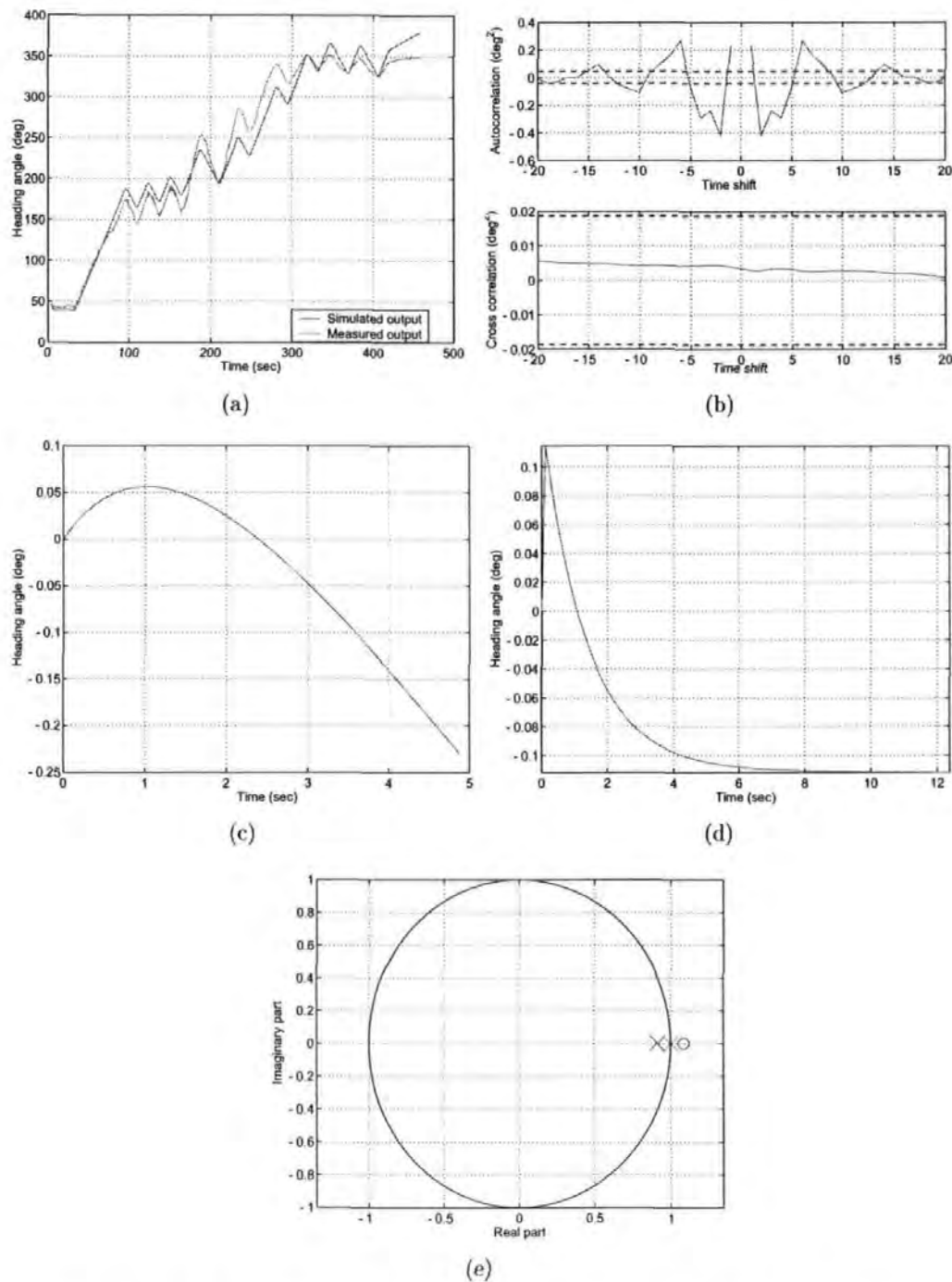


Figure B.1: (a) 83.1888% fit between simulated and measured output, (b) autocorrelation residuals and cross correlation of residuals and the input, (c) step response, (d) impulse response and (e) pole zero plot

B.2 THIRD ORDER

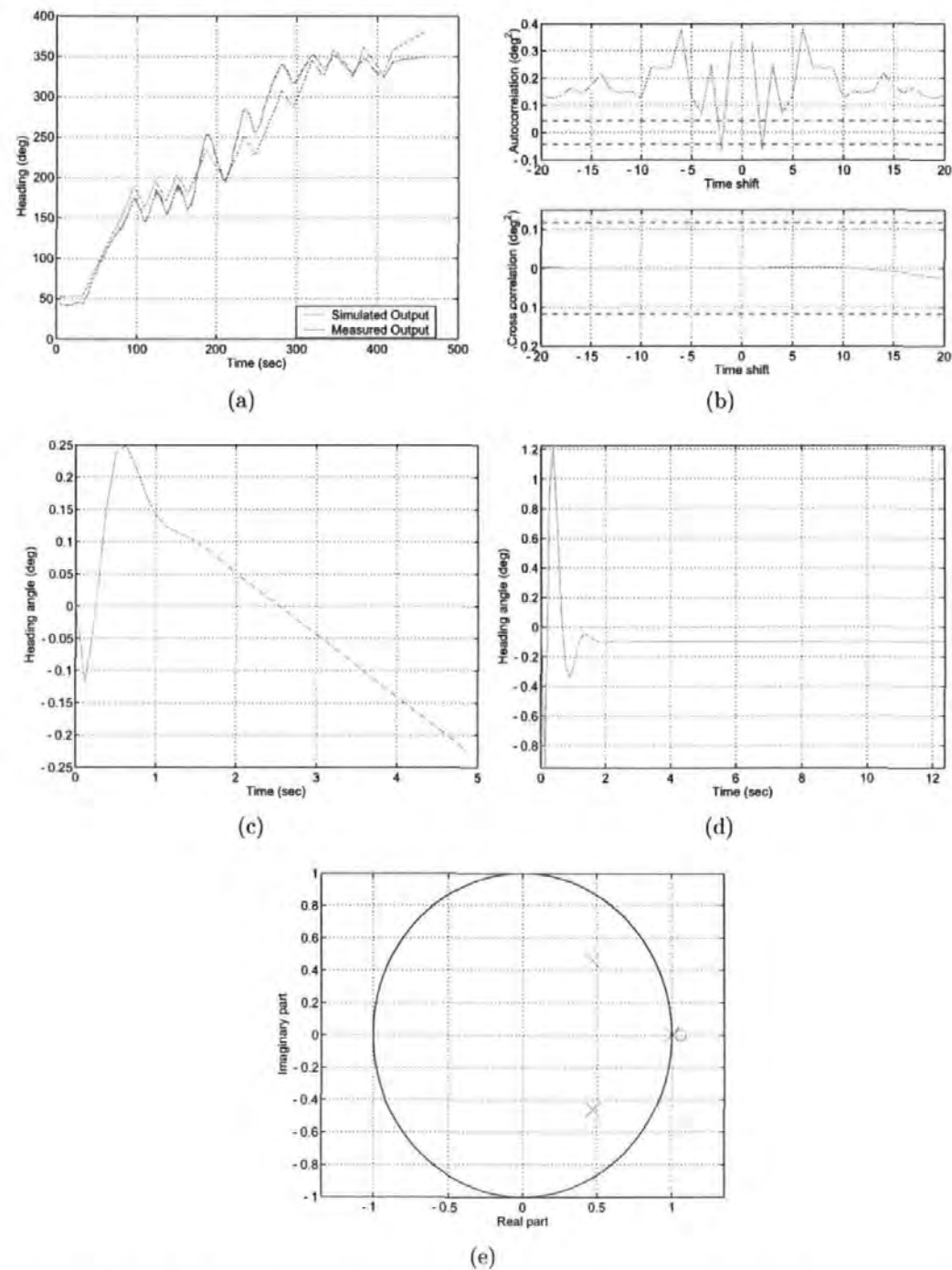


Figure B.2: (a) 81.9476% fit between simulated and measured output, (b) autocorrelation residuals and cross correlation of residuals and the input, (c) step response, (d) impulse response and (e) pole zero plot

B.3 FOURTH ORDER

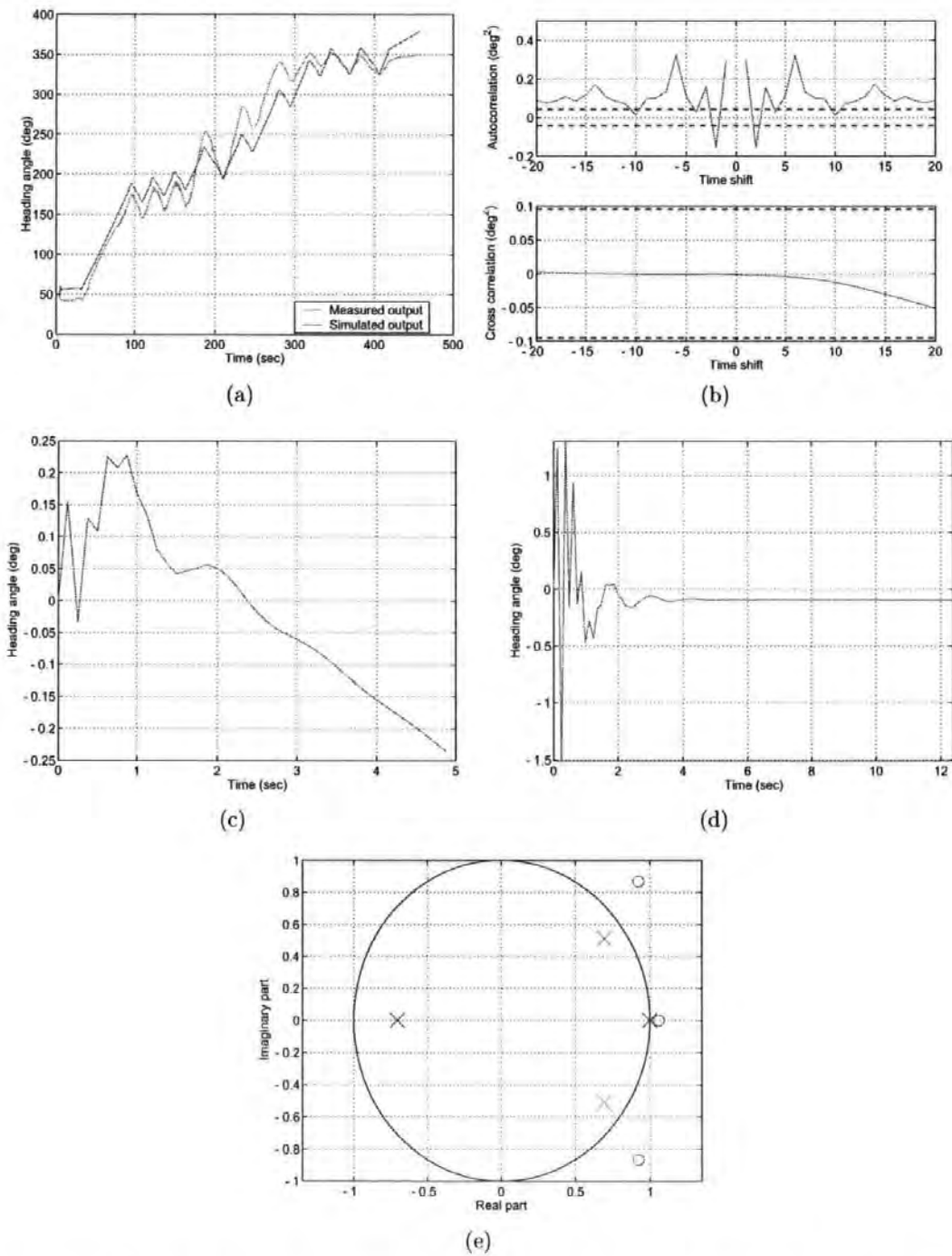


Figure B.3: (a) 81.5526% fit between simulated and measured output, (b) autocorrelation residuals and cross correlation of residuals and the input, (c) step response, (d) impulse response and (e) pole zero plot

B.4 FIFTH ORDER

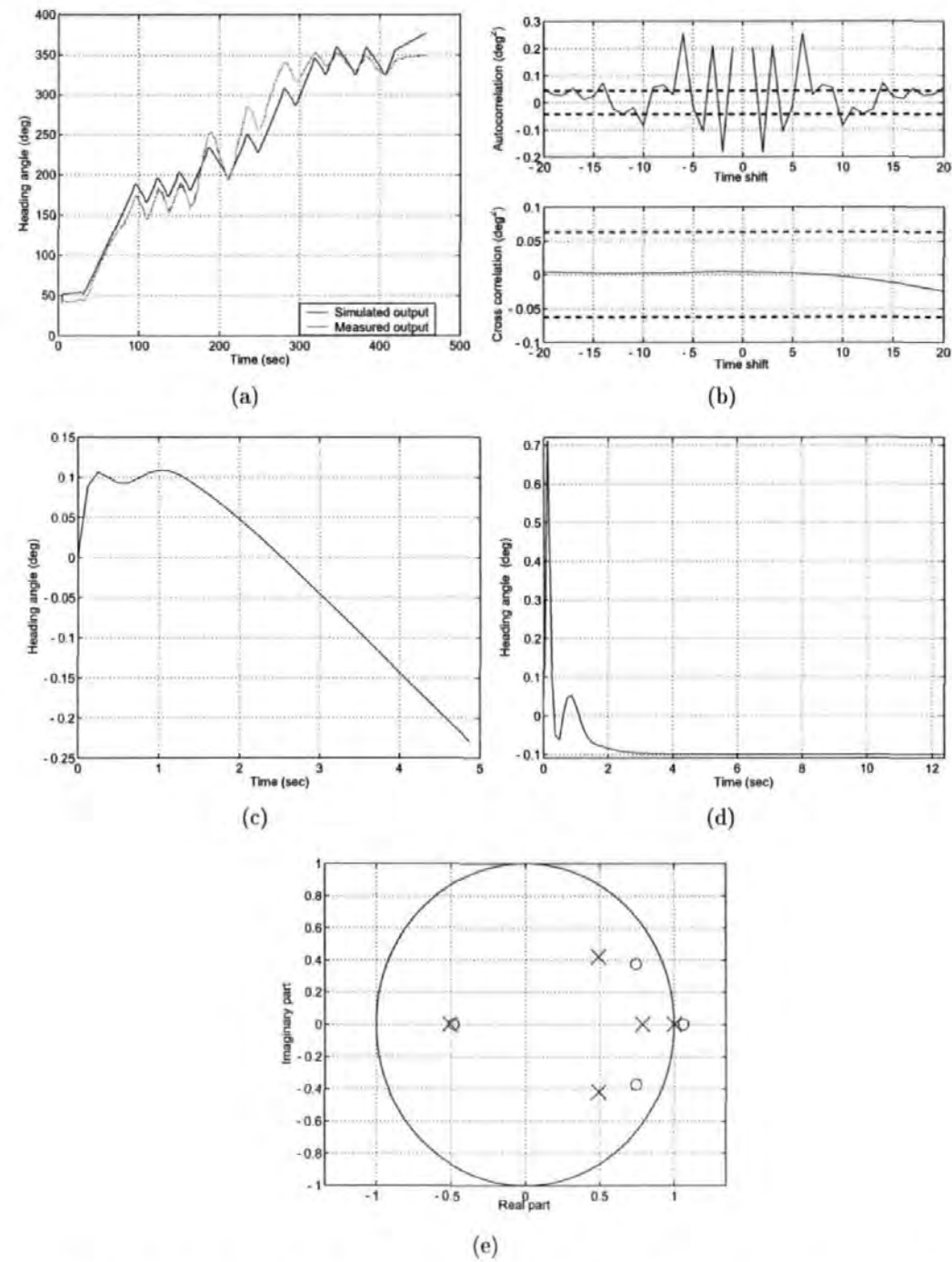


Figure B.4: (a) 82.3896% fit between simulated and measured output, (b) autocorrelation residuals and cross correlation of residuals and the input, (c) step response, (d) impulse response and (e) pole zero plot

B.5 SIXTH ORDER

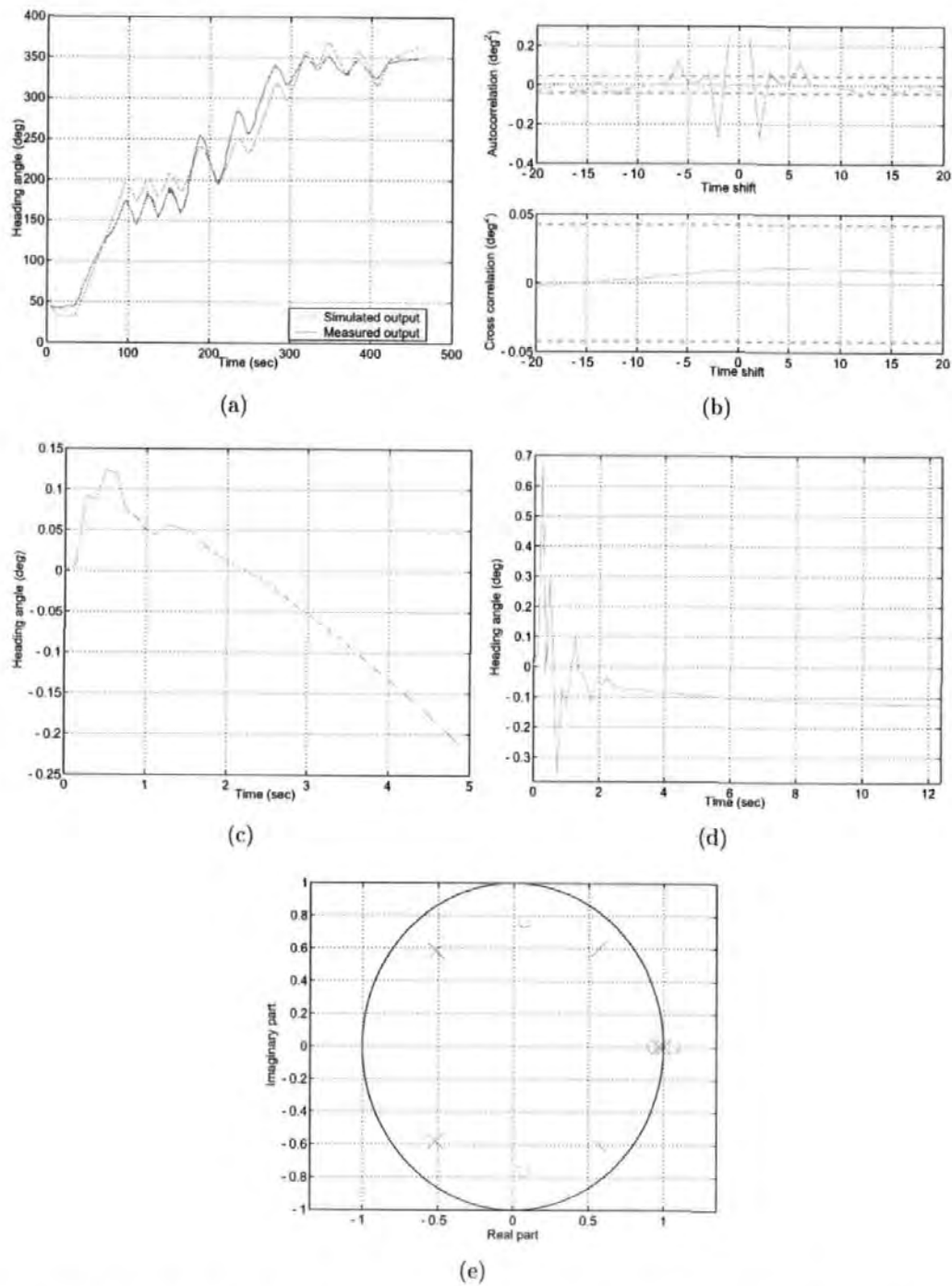


Figure B.5: (a) 83.5926% fit between simulated and measured output, (b) autocorrelation residuals and cross correlation of residuals and the input, (c) step response, (d) impulse response and (e) pole zero plot

B.6 SEVENTH ORDER

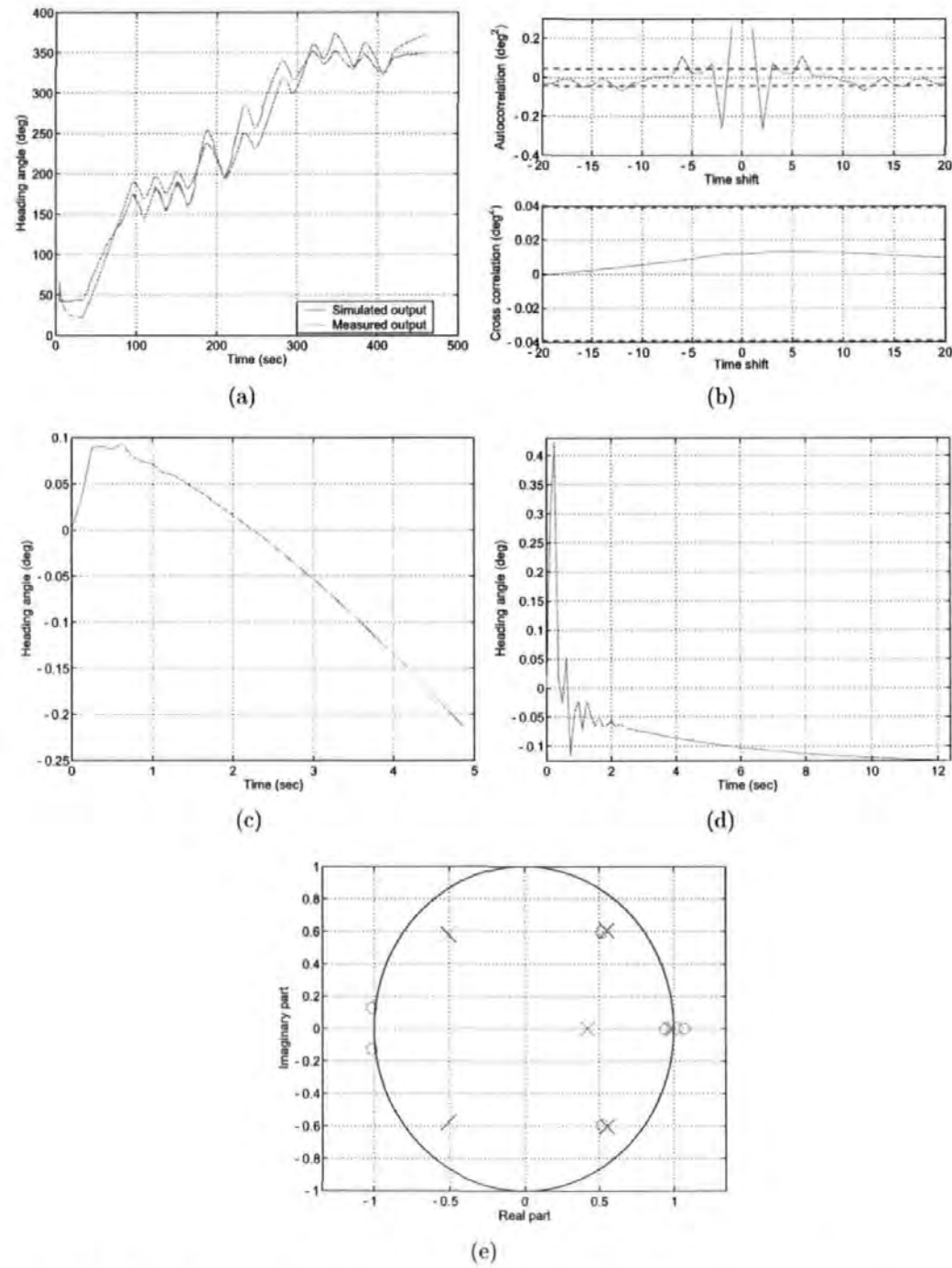


Figure B.6: (a) 82.6636% fit between simulated and measured output, (b) autocorrelation residuals and cross correlation of residuals and the input, (c) step response, (d) impulse response and (e) pole zero plot

B.7 EIGHTH ORDER

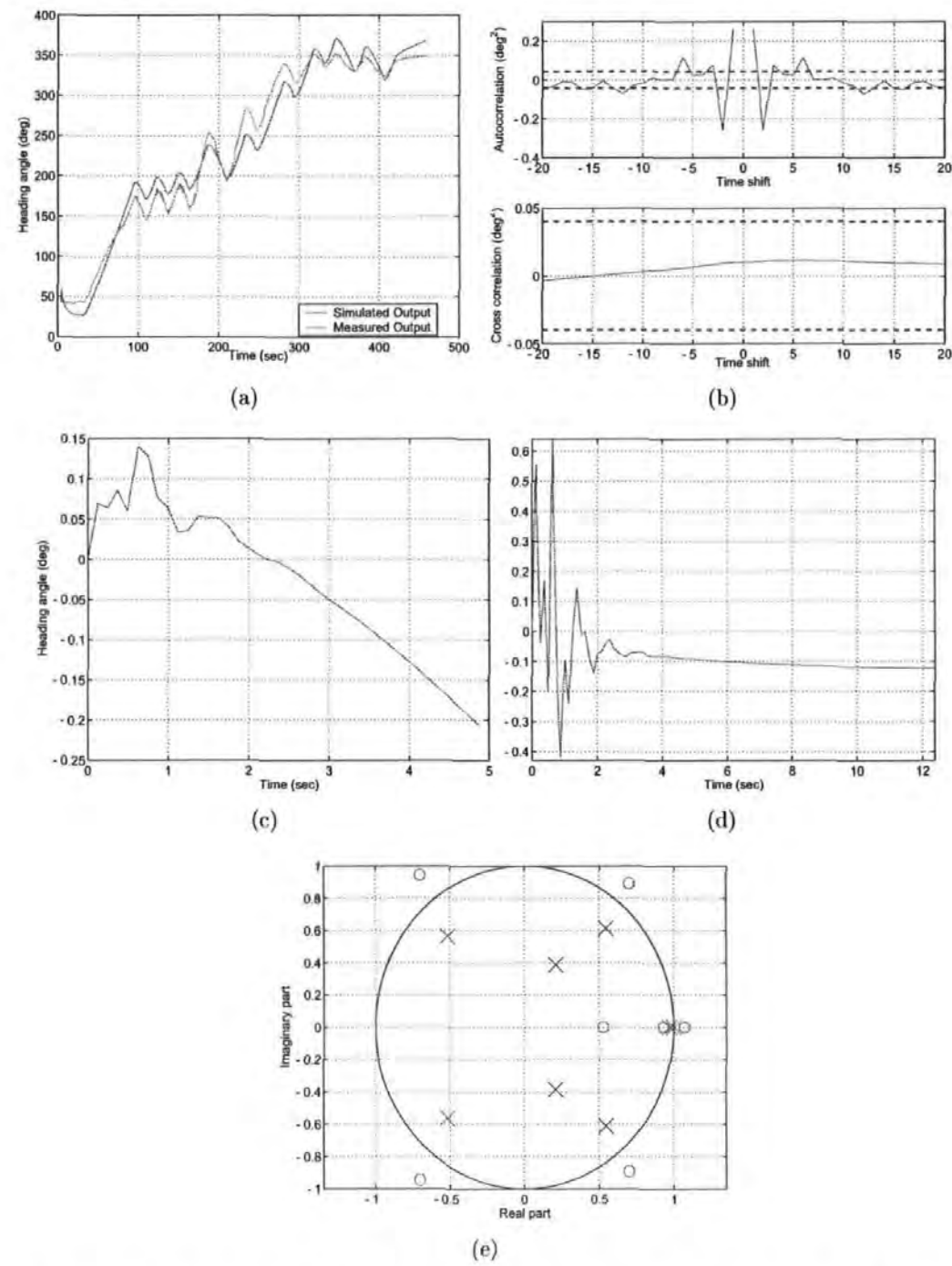


Figure B.7: (a) 88.2953% fit between simulated and measured output, (b) autocorrelation residuals and cross correlation of residuals and the input, (c) step response and (d) pole zero plot

B.8 NINTH ORDER

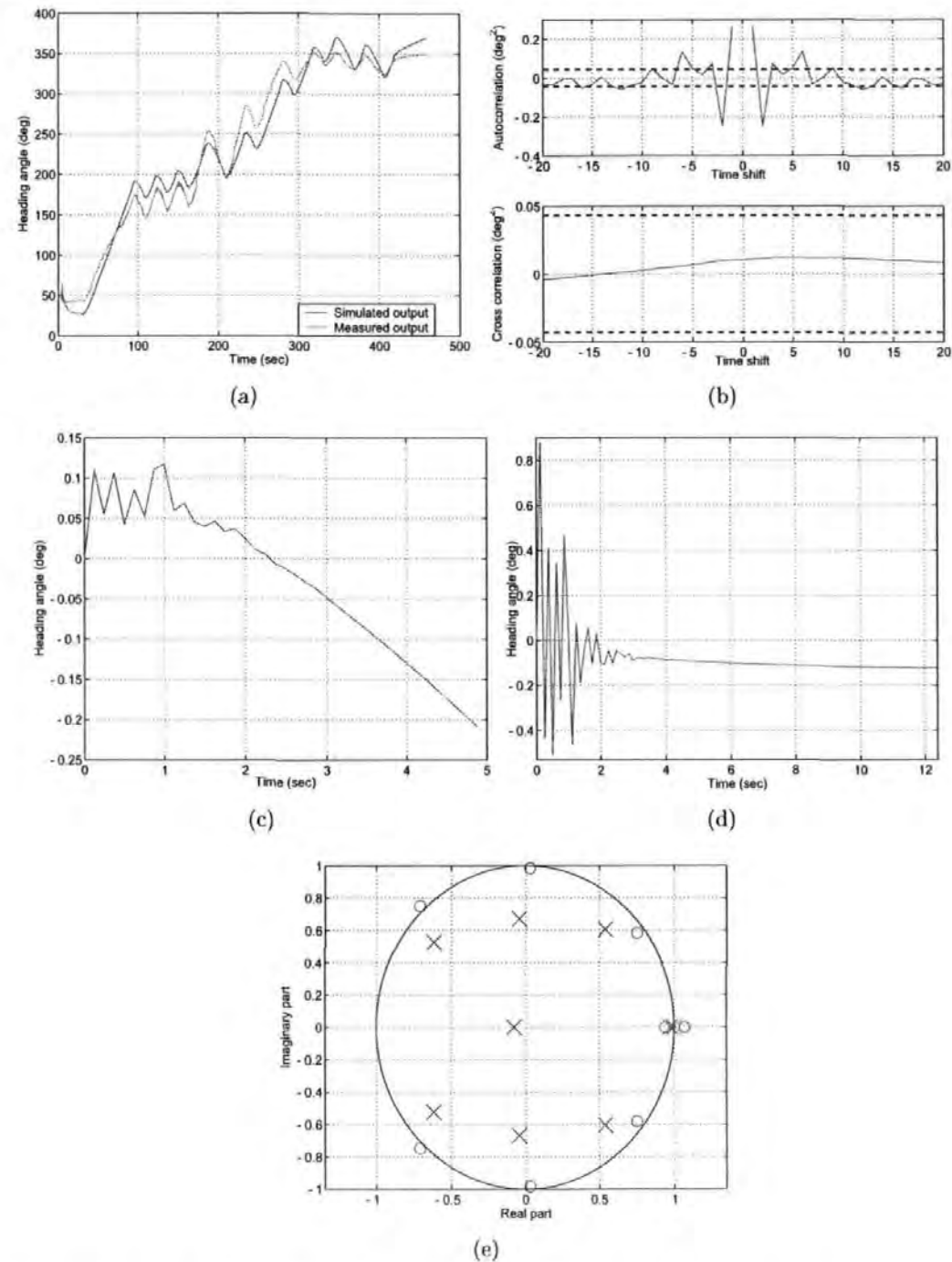


Figure B.8: (a) 83.2608% fit between simulated and measured output, (b) autocorrelation residuals and cross correlation of residuals and the input, (c) step response and (d) pole zero plot

B.9 TENTH ORDER

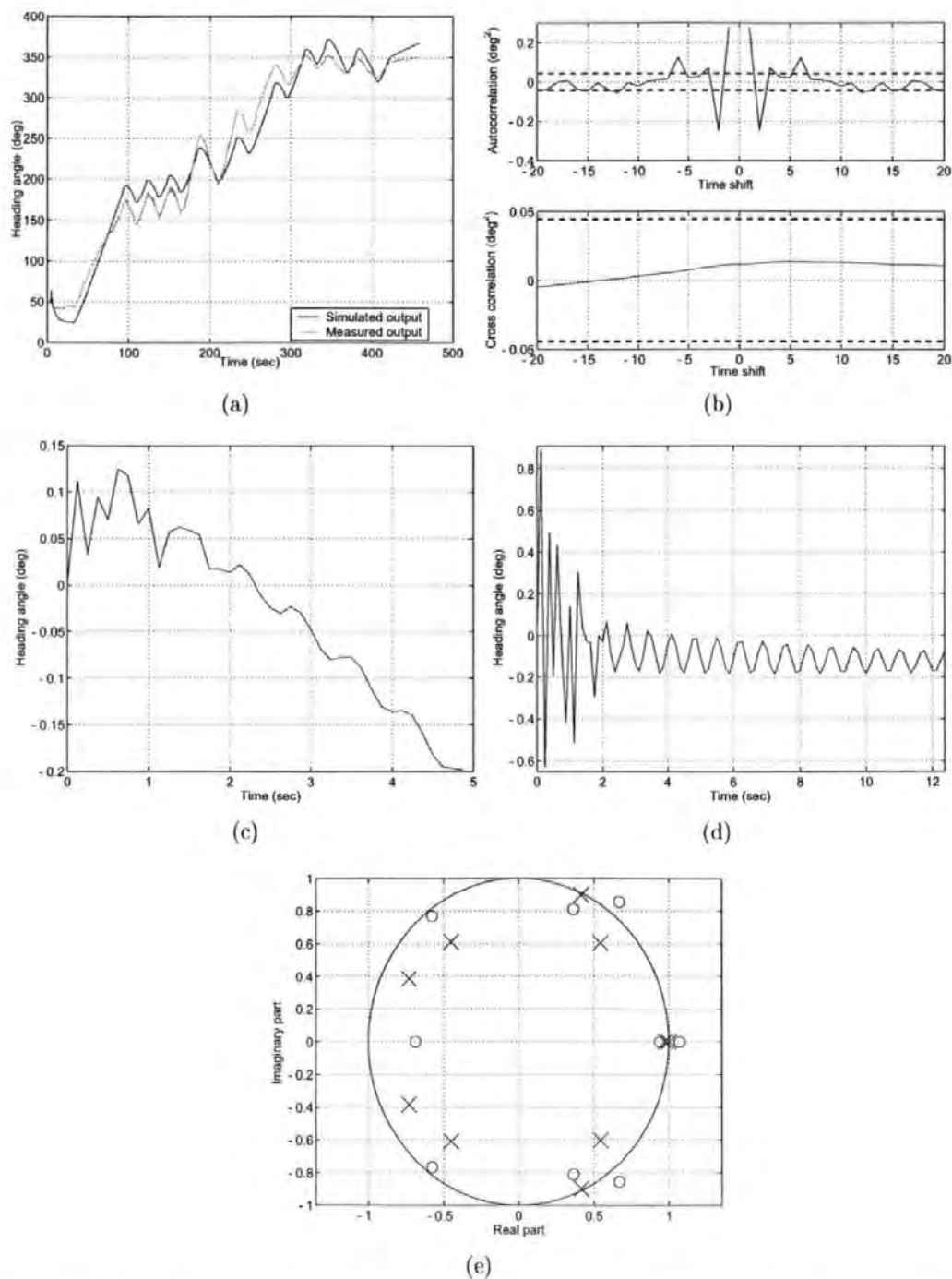


Figure B.9: (a) 83.0287% fit between simulated and measured output, (b) autocorrelation residuals and cross correlation of residuals and the input, (c) step response, (d) impulse response and (e) pole zero plot

The impulse response of the identified *Hammerhead* systems are shown in Figure B.1(d)-B.9(d). A comparison can be made between the response of the second order system and its higher order counterparts. The second order system dies away to a constant heading at approximately 10(sec), whilst the higher order systems, with the exception of the seventh order system, die away at approximately 2-3(sec). However, observation on the response also reveals the nature of oscillation of the systems. Although they die away more rapidly than the second order system, the higher order systems are more oscillatory and consequently may require a complex proper control system design. This is also one of the reason behind the choice on the second order system.

APPENDIX D

LINEAR AND EXTENDED KALMAN FILTER ALGORITHM

D.1 THE LINEAR KALMAN FILTER

State space form of a linear discrete time system observed by a linear measurement where both process and measurement are influenced by additive Gaussian noise can be put into:

$$x_{k+1} = \mathbf{A}_k x_k + \mathbf{B}_k u_k + w_k, w_k \sim N(0, \mathbf{Q}_k) \quad (\text{D-1})$$

$$z_k = \mathbf{H}_k x_k + v_k, v_k \sim N(0, \mathbf{R}_k) \quad (\text{D-2})$$

The process noise (v_k) and the measurement noise (w_k), are generally assumed mutually independent ($E[w_i v_j^T] = 0$, for all i and j).

The Kalman filter equations for the system in Equation (D-1) and (D-2) can be summarised as in Table D.1:

Process model	$x_{k+1} = \mathbf{A}_k x_k + \mathbf{B}_k u_k + w_k$
Measurement model	$z_k = \mathbf{H}_k x_k + v_k$
Initial conditions	$\hat{x}_0^-, \mathbf{P}_0^-$
Other assumptions	$E[w_i v_j^T] = 0, w_k \sim N(0, \mathbf{Q}_k), v_k \sim N(0, \mathbf{R}_k)$
Time update	$\hat{x}_{k+1}^- = \mathbf{A}_k \hat{x}_k^- + \mathbf{B}_k u_k$ $\mathbf{P}_{k+1}^- = \mathbf{A}_k \mathbf{P}_k^- \mathbf{A}_k^T + \mathbf{Q}_k$
Measurement update	$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T [\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k]^{-1}$ $\hat{x}_k = \hat{x}_k^- + \mathbf{K}_k [z_k - \mathbf{H}_k \hat{x}_k^-]$ $\mathbf{P}_k = [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k] \mathbf{P}_k^-$

Table D.1: Linear discrete Kalman filter equations

At every sample, the Kalman filter uses the process model to propagate the state estimate:

$$\hat{x}_{k+1}^- = \mathbf{A}_k \hat{x}_k^- + \mathbf{B}_k u_k \quad (\text{D-3})$$

The minus sign (-) on \hat{x}_{k+1}^- indicates that the estimate is an *a priori* estimate for the next step. The state estimate update in Equation (D-3) does not equal the true state update in Equation (D-1). Therefore, the covariance of the estimation error (\mathbf{P}_k) needs also to be propagated:

$$\mathbf{P}_{k+1}^- = \mathbf{A}_k \mathbf{P}_k \mathbf{A}_k^T + \mathbf{Q}_k \quad (\text{D-4})$$

Equation (D-3) and (D-4) are referred to as the *time update/prediction* equations of the Kalman filter. If a measurement available, the estimate is updated by incorporating the incoming data with a gain that takes the covariance of the estimation error and the new data into account:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T [\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k]^{-1} \quad (\text{D-5})$$

$$\hat{x}_k = \hat{x}_k^- + \mathbf{K}_k [z_k - \mathbf{H}_k \hat{x}_k^-] \quad (\text{D-6})$$

The gain, \mathbf{K}_k , is denoted the *Kalman gain* and is the gain that minimises the trace of the resulting covariance matrix, \mathbf{P}_k . The covariance after the incorporation of new measurement is:

$$\mathbf{P}_k = [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k] \mathbf{P}_k^- \quad (\text{D-7})$$

Equation (D-5)-(D-7) are referred to as the *measurement update* of the Kalman filter.

D.2 THE EXTENDED KALMAN FILTER

Various attempts have been made to modify or generalise the Kalman filter to fit circumstances beyond linear system and additive Gaussian noise. One of the most popular representation of these filters is the *extended* Kalman filter, used when the process or output equations are non-linear.

A non-linear system is given by:

$$x_{k+1} = a(x_k, u_k) + w_k, w_k \sim N(0, \mathbf{Q}_k) \quad (\text{D-8})$$

$$z_k = h(x_k) + v_k, v_k \sim N(0, \mathbf{R}_k) \quad (\text{D-9})$$

Process model	$x_{k+1} = a(x_k, u_k) + w_k, w_k \sim N(0, \mathbf{Q}_k)$
Measurement model	$z_k = h(x_k) + v_k, v_k \sim N(0, \mathbf{R}_k)$
Initial conditions	$\hat{x}_0^-, \mathbf{P}_0^-$
Other assumptions	$E[w_i v_j^T] = 0, w_k \sim N(0, \mathbf{Q}_k), v_k \sim N(0, \mathbf{R}_k)$
Time update	$\hat{x}_{k+1}^- = a(\hat{x}_k, u_k)$ $\mathbf{P}_{k+1}^- = \mathbf{A}_k \mathbf{P}_k \mathbf{A}_k^T + \mathbf{Q}_k$
Measurement update	$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T [\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k]^{-1}$ $\hat{x}_k = \hat{x}_k^- + \mathbf{K}_k [z_k - \mathbf{H}_k \hat{x}_k^-]$ $\mathbf{P}_k = [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k] \mathbf{P}_k^-$
Linearisations	$\mathbf{A}_k = \frac{\partial a}{\partial x_k} _{x_k = \hat{x}_{k+1}^-}, \mathbf{H}_k = \frac{\partial h}{\partial x_k} _{x_k = \hat{x}_k}$

Table D.2: Discrete extended Kalman filter equations

Table D.2 provides the summary of the extended Kalman filter equations. The state estimate time update of the filter is now:

$$\hat{x}_{k+1}^- = a(\hat{x}_k, u_k) \quad (\text{D-10})$$

If $a(\hat{x}_k, u_k)$ is linearised with respect to x (the Jacobians $\mathbf{A} = \nabla_x a|_{x=\hat{x}_{k+1}^-}$, an approximated linearised covariance update can be performed:

$$\mathbf{P}_{k+1}^- = \mathbf{A}_k \mathbf{P}_k \mathbf{A}_k^T + \mathbf{Q}_k \quad (\text{D-11})$$

To perform the data update $h(x_k)$ must be linearised as well:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T [\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k]^{-1} \quad (\text{D-12})$$

$$\hat{x}_k = \hat{x}_k^- + \mathbf{K}_k [z_k - \mathbf{H}_k \hat{x}_k^-] \quad (\text{D-13})$$

$$\mathbf{P}_k = [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k] \mathbf{P}_k^- \quad (\text{D-14})$$

It is clear that apart from the linearisations and the state estimate propagations, the equations for the extended Kalman filter are similar to the linear Kalman filter. However, as the equations here have been linearised, the error covariance matrix (\mathbf{P}) is only a first order approximation of the true estimation error covariance, and the filter estimate is no longer optimal. Despite this, the filter works well in many practical applications (Brown and Hwang, 1997).

APPENDIX E

FUZZY LOGIC

Human brains do not reason in the same way as computers. The way computers reason is in clear steps with strings of 0s and 1s. Humans reason with a sense of a gradual degree of truth to attributes like *big*, *fast* and *young*. However, these vagueness way of thinking is usually avoided in classical logic and computing, because it is considered as having a negative influence in their inference processes (Nauck *et al.*, 1997). This conundrum has been noticed by several scholars in the past and subsequent attempts to develop a mathematical structure capable of encapsulating the human way of reasoning have then undertaken. In effect, several methodologies have been proposed and nowadays they are developed under a domain so-called soft computing (SC) technology.

The term SC was invoked by Zadeh (1994), to refer to systems that are capable of providing tolerance to uncertainty and an imprecision in their reasoning. SC constitutes several techniques with fuzzy logic (FL), genetic computing (GP), neural networks (NN) and probabilistic reasoning (PR) as the cardinal members (Tsoukalas and Uhrig, 1997). FL, GP, NN and PR mostly contribute to different research domains and therefore they are synergistic and complementary rather than competitive in nature. The blend between these "substances" leads to the so called "hybrid intelligent systems" (Jang *et al.*, 1997). Nowadays, one of the most noticeable and burgeoning of the hybrid intelligent systems are fuzzy-genetic systems.

Fuzzy-genetic systems integrate synergistically two complementary approaches: fuzzy logic and the genetic algorithm (GA) (discussed in details in Appendix F). On the one hand, fuzzy inference systems can incorporate human knowledge and perform

inferencing and decision making thereof; while a GA can be used to perform systematically a random search in order to find an optimal solution to an engineering problem. What genetic-fuzzy systems together can provide is a single methodology with all the above characteristics and these are desirable in many MSDF applications.

E.1 BRIEF CHRONOLOGICAL RETROSPECTIVE

Zadeh proposed mathematical theory of approximate reasoning capable of emulating human logic in his breakthrough paper on fuzzy sets in 1965 (Zadeh, 1965). The word *fuzzy* is mentioned first in this paper to mean "vague" in the technical sense. In the first decade after this paper was published, most application oriented papers in this field were focused on theoretical studies toward possible applications and sometimes with real applications on a laboratory scale. One of the most noticeable idea was the concept of fuzzy control pioneered by Mamdani and Assilian (1975). In this work, control behaviour was described by a qualitative algorithm. The first application of the proposed method was the design of the automatic control of a steam engine/boiler combination in the laboratory. But not long after that, Holmblad and Østergaard (1982) realised the automatic control of a cement kiln. Since then the concept of fuzzy systems was soon to be associated with a vast number of practical applications. One well known example was the the design of an automatic drive fuzzy control system for subway trains in Sendai city, Japan (Reznik, 1997). Indeed, fuzzy logic eventually received formal recognition by the technological world and lately efforts have increased to define a standard, based on ISO-9000, the general system development guidelines, for the methodology of fuzzy logic systems (Schram *et al.*, 1997).

E.2 FUZZY SYSTEMS

The fuzzy set paradigm of Zadeh's theory of approximate reasoning established a connection between concrete and ambiguous ways of reasoning. Mathematical formulation was developed to allow the concept of fuzziness or ambiguity captured in a language that can be comprehended by computers, to provide capabilities to emulate the human mind as a decision maker. Later, Dubois and Prade (1988) and Klir and

Yuan (1995) devised the theory of possibility developed on the ground of this new methodology.

The fundamental principle of *fuzzy sets* is a variable notion of membership. That is to say that elements can belong to sets to a certain degree. The fuzzy set theory was developed to handle situations that have no sharp boundaries or in which the events are ambiguously defined. Consider for example the set of *HIGH*, assuming that the perception of high is the velocity of an object that moves with no less than 80(km/hour):

$$HIGH = \{x \in P \mid Velocity(x) \geq 80\} \quad (E-1)$$

over some domain P of all moving objects and using a function *Velocity* that returns the velocity of an object $x \in P$ in km/hour. Characteristic function can also be defined for the same problem:

$$\mu_{HIGH}(x) = \begin{cases} 1 & : Velocity(x) \geq 80 \\ 0 & : 80 < Velocity(x) \end{cases} \quad (E-2)$$

which assigns to elements of P a value of 1 whenever this element belongs to the set of *HIGH*, and 0 otherwise. This characteristic function can be seen as a *membership function* for the set *HIGH*, defining the set *HIGH* on P .

Using this perception, an object that moves 79.99(km/hour) is therefore not considered moving with a high-velocity. Hence, defining the set *HIGH* using such a sharp boundary seems not very appropriate. Using the fundamental principle of fuzzy sets, it can be specified that 79.99 (km/hour) still belongs to the set of *HIGH*, but only to a degree less than one. The corresponding membership function would look slightly different:

$$\mu_{HIGH}(x) = \begin{cases} 1 & : Velocity(x) \geq 80 \\ 1 - \frac{80 - (Velocity(x))}{20} & : 60 \leq Velocity(x) < 80 \\ 0 & : Velocity(x) < 60 \end{cases} \quad (E-3)$$

The set *HIGH* contains the velocity of an object between 60 and 80(km/hour) with

a linearly decreasing *degree of membership*, that is to say, the closer the velocity of an object approaches 60, the closer its degree of membership to the set of *HIGH* approaches zero. Hence, the fuzzy set allows an ambiguous proposition to be described in a logical sense using the entire interval $[0,1]$ to create degrees of possibility of truth, in contrast to classical sets, where an element can only either belong to a set (Boolean value 1) or lies completely outside of this set (Boolean value 0).

Essentially, fuzzy sets can be considered as look-up tables that contain a series of truth membership values that encode the imprecision associated with certain events. Consequently, unlike their classical counterparts which are not capable of capturing the ambiguity of many real-life situations, fuzzy sets have more expressive power in this respect. In conclusion, *fuzzy logic* can be defined as the discipline that represents vagueness, imprecision or uncertainty by handling multi-valent membership degrees. It is a *precise discipline* dealing with *imprecise data*.

From a practical point of view, Zadeh's philosophy allows the mechanism of human reasoning to be incorporated into the systems theory and led to the development of a linguistic type of systems called *fuzzy systems*. The use of fuzzy sets to enhance the performance of a system that employs MSDF techniques, permits a generalisation of information and a quantification of imprecision, often required in the design and implementation of such a system. Fundamentally, the representation of information in fuzzy systems imitates the mechanism of approximate reasoning performed in the human mind.

A fuzzy system comprises of four major components:

1. A fuzzification interface, which maps real crisp inputs into fuzzy inputs, by means of fuzzy sets,
2. A rule base containing a number of rules in the form of "**IF** < *antecedent* > **THEN** < *consequent* >", where knowledge about the problem is acquired,
3. A mechanism of inference, which deals with the fuzzy rules in order to generate fuzzy conclusions (consequent) from fuzzy premises (antecedents),

4. A defuzzification interface, which realises the transformation of fuzzy outputs into crisp output.

Further details of the individual elements of a fuzzy system mentioned above are presented in the following Sections.

E.3 FUZZIFICATION

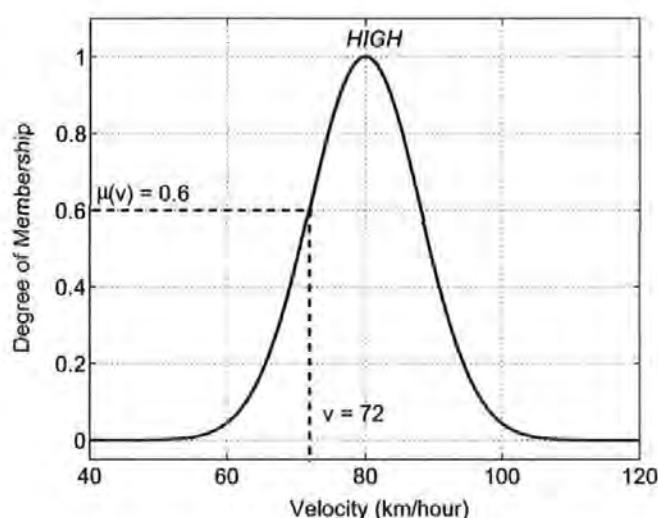


Figure E.1: The fuzzy set 'HIGH'

Essentially, four elements make up a fuzzy set, Figure E.1: a linguistic value (*e.g.*, *HIGH*), a horizontal axis to denote the real-world input variable, represented through crisp values (*e.g.*, *Velocity in km/hour*), a vertical axis indicating the degree of membership to the fuzzy set (a value in the interval $[0,1]$), and a membership function that encodes each element of the horizontal axis v to a degree of membership in the vertical axis $\mu(v)$. More specifically, $\mu(v)$ describes quantitatively how accurate the linguistic value *HIGH* maps the crisp value v . For the representation considered in Figure E.1, where the linguistic value *HIGH* denotes velocities in the range $[50,110](km/hour)$, $\mu(v)$ indicates how well a Velocity of v within this interval is represented by the attribute *HIGH*. *E.g.* if $v = 72(km/hour)$, then its degree membership to the fuzzy set *HIGH*, is $\mu(v) = 0.6$.

For the example presented herein, the membership function has a Gaussian shape, but in practice these functions may take various forms: triangular, trapezoidal, *etc.* Due to simple formula and computational efficiency, triangular membership functions have been used extensively, especially in real-time implementations. However, since these functions are composed of straight line segments, they are not smooth at the corner points specified by the parameters. Gaussian is a type of membership functions defined by smooth, non-linear differentiable functions. Although Gaussian membership function achieve smoothness, they are unable to specify asymmetric functions, which are important in certain applications (Gorzalczabny, 2002).

A fuzzy set A is thus defined through a set of pairs of a membership function μ_A that assign each element v in the *Universe of Discourse* U , a degree of membership $A, \mu_A(v)$:

$$A = \{(v, \mu_A(v)) | v \in U, \mu_A(v) : U \rightarrow [0, 1]\} \quad (\text{E-4})$$

with the partitions corresponding to each fuzzy set in U are known as domains. Considering the example presented in Figure E.1, the U is the interval $[0,10]$ and the domain for the fuzzy set *HIGH* is $[50,110]$, with *Velocity* constitutes a *fuzzy model parameter*. To define completely a problem in terms of fuzzy logic, the allowable range of each model parameter (*Velocity, Acceleration, etc.*) is divided into overlapping regions (*SLOW, MEDIUM, HIGH*), each region describing, semantically, a domain of the associated fuzzy variable, as presented in Figure E.2. The overall process of collecting the current real-world crisp number and to transform the values appropriately into a fuzzy number by means of fuzzy sets is known in the literature as the process of *fuzzification*.

E.4 FUZZY RULE BASE

Fuzzy systems rely on the knowledge possessed by human expert, with which qualitative **IF-THEN** rules are then developed and embedded in the system structure.

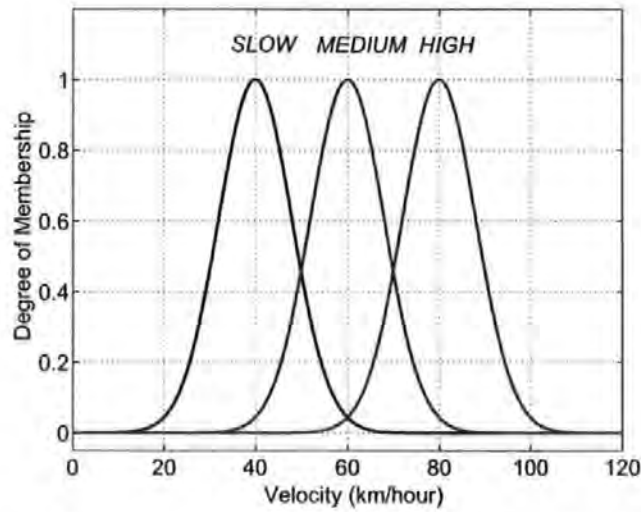


Figure E.2: The partition of the fuzzy variable *Velocity*

Fuzzy rules can be used to characterise imprecise dependencies between different variables. Consider:

$$\text{IF } \textit{Velocity}(x) \geq 80(\textit{km/hour}) \text{ THEN } \textit{Mission}(x) < 1(\textit{hour}) \quad (\text{E-5})$$

describing the mission-time of a particular vehicle moving with a certain velocity. Obviously using linguistic values can make such a rule much more readable:

$$\text{IF } \textit{Velocity}(x) \text{ IS HIGH THEN } \textit{Mission}(x) \text{ IS SHORT} \quad (\text{E-6})$$

Fuzzy rules are therefore of interest whenever a dependency is either imprecise or a high level of precision is not desired in order to maintain a high level of interpretability. A basic type of a fuzzy rule which is widely used has the following forms:

$$\text{IF } x_1 \text{ IS } A_1 \text{ AND } \dots \text{ AND } x_n \text{ IS } A_n \text{ THEN } y \text{ IS } B \quad (\text{E-7})$$

where the A_i in the antecedent and the B in the consequent are linguistic values of the input vector \mathbf{x} and the output variable y , respectively. These type of rules, where the antecedent and the consequent rule are expressed through fuzzy predicates are

known as *Mamdani* (Mamdani and Assilian, 1975) or *linguistic fuzzy systems*. In many applications, rules that assign crisp equations to the output variable are also used. Most commonly there are either linear or quadratic dependencies on one or more input variables (first order or second order *Takagi-Sugeno*) fuzzy system (Takagi and Sugeno, 1985). The later system emerged as an alternative to Mamdani's linguistic formulation, and the idea was to alter the rule structure so that qualitative and quantitative knowledge can be equally incorporated into the knowledge base.

A correct definition of the rule base is ensured if the following criteria are satisfied:

- 1. Completeness - any combination of inputs should result in appropriate output
- 2. Consistency - The rule base does not contains contradictions,
- 3. Continuity - neighbouring rules generate output fuzzy sets with non-empty intersections.

E.5 INFERENCE ENGINE

The inference engine is often regarded as the heart of the fuzzy system, a description that reflects the primary importance of this system component in processing fuzzy data. The inference engine is an interpreter of the rule base, with the task to derive a fuzzy conclusion from a set of fuzzy IF-THEN rules. The basic inference mechanism is the generalised modus ponens (GMP) (Lee, 1990; Jang and Sun, 1995). The GMP is the extension of the classical Boolean logic modus ponens (MP) rule of inference. The MP can be illustrated as follows:

Rule:	IF x IS A THEN y IS B
Fact:	x IS A
<hr/>	
Conclusion:	y IS B

where A and B designate predicates which characterise properties of x and y respectively. According to MP, a truth of a proposition y IS B can be inferred from the truth x IS A and the implication IF x IS A THEN y IS B . However, in much human reasoning MP is employed in an approximate manner, which can be written as:

Rule:	IF x IS A THEN y IS B
Fact:	x IS A'
Conclusion:	y IS B'

This inference procedure is the so-called GMP as it has MP as a special case. When A' and B' are input and output fuzzy sets in U and V respectively, the A' is mapped to B' through the fuzzy inference engine.

In MP, the equivalent of a rule $\langle \text{IF } p \text{ THEN } q \rangle$ is the implication $p \rightarrow q$. Similarly, the interpretation of the fuzzy **IF-THEN** operation is given by a logical function whose arguments are the membership functions of the antecedent and consequent part of the rule. This logical operation results in a membership function μ_Q associated with the evaluated rule; the process of doing so is known as *fuzzy implication*. Driankov *et al.* (1993) and Wang (1997) detailed the various types of implication that have been proposed in the literature. *E.g.*, the implication proposed by Mamdani, calculates the membership function of a rule with minimum operation. For the rule: $\langle \text{IF } x \text{ IS } A \text{ THEN } y \text{ IS } B \rangle$, where $\langle x \text{ IS } A \rangle$ and $\langle y \text{ IS } B \rangle$ are fuzzy propositions defined in U and V , respectively. According to Mamdani implication, the corresponding membership function is

$$\mu_Q(x, y) = \min(\mu_A(x), \mu_B(y)) \quad (\text{E-8})$$

where x defined in U , y defined in V (implying that Q is defined in $U \times V$, where \times denotes the Cartesian product operator). If a fuzzy proposition (FP) in a rule contains connectives **AND**, then logical operations *min* is basically used to calculate its overall membership degree:

$$\begin{aligned} \text{FP: } x_1 \text{ IS } A_1 \text{ AND } x_2 \text{ IS } A_2 \text{ AND...AND } x_n \text{ IS } A_n &\Leftrightarrow \\ \mu_{FP}(x_1, \dots, x_n) &= \min(\mu_{A_1}(x_1), \mu_{A_2}(x_2), \dots, \mu_{A_n}(x_n)) \end{aligned} \quad (\text{E-9})$$

The Mamdani implication can be written as:

$$\mu_Q(x, y) = \min(\mu_{FP_{ant}}(x), \mu_{FP_{cons}}(y)) \quad (E-10)$$

if the general expression for a rule with connectives (**IF** FP_{ant} **THEN** FP_{cons} , with FP_{ant} defined in $U = U_1 \times U_2 \times \dots \times U_n$ and FP_{cons} defined in $V = V_1 \times V_2 \times \dots \times V_m$) is adopted. Logically, the Mamdani implication is equivalent to: $p \rightarrow q \equiv p \wedge q$.

Finally, an output B' , given a set A' in U and having determined μ_Q , is obtained by means of *compositional rule of inference* proposed by Zadeh (1973):

$$B' = A' \circ R = A' \circ (A \rightarrow B) \quad (E-11)$$

with \circ is the sup-min operator, $R = A \rightarrow B$ is a fuzzy relation in the product space $U \times V$, and

$$\mu_{B'}(y) = \sup_x (\min(\mu_{A'}(x), \mu_Q(x, y))) \quad (E-12)$$

This relation is called *sup-min composition*. In its general form, a compositional operator is expressed as a *sup-t* composition, where t represents the logical t -norm operator (*min*, *algebraic product*, *bounded product* and *drastic product*).

The substitution of μ_Q in (E-10) with the expression from (E-12) yields:

$$\begin{aligned} \mu_{B'}(y) &= \sup_x (\min(\mu_{A'}(x), \min(\mu_{FP_{ant}}(x), \mu_{FP_{cons}}(y)))) = \\ &= \sup_x (\min(\mu_{A'}(x), \mu_{FP_{ant}}(x)), \mu_{FP_{cons}}(y)) = \sup_x (\min(\tau_i, \mu_{FP_{cons}}(y))) \end{aligned} \quad (E-13)$$

with τ_i is defined as the degree of matching or fulfillment (DOF). The DOF of each individual rule can be obtained by matching the facts with the rule premises. This degree of matching is given by the actual membership degree of a fuzzy set A' to the input fuzzy set A .

The discussion so far is based on a rule base that consists only one rule. In any practical application, a fuzzy system can be formed by several rules. In such a case, a method to combine all the rules should be utilised in order to generate the appropriate fuzzy output. *composition based* and *individual rule based* are two categories of inference on a set of rules that are identified by Wang (1997).

In composition based inference, a logical operation, normally the union (*max*) are used to combined all the component rules of the rule base into a single rule. The membership function associated to the entire rule is described as:

$$\mu_{Q_{All}}(x, y) = \max(\mu_{Q(1)}(x, y), \mu_{Q(2)}(x, y), \dots, \mu_{Q(N)}(x, y)) \quad (E-14)$$

where $\mu_{Q(i)}$ is the membership function related with the i -th rule. Similar to (E-12), the fuzzy output is then calculated as it would be the outcome of a single, resultant rule :

$$\mu_{B'}(y) = \sup_x (\min(\mu_{A'}(x), \mu_{Q_{All}}(x, y))) \quad (E-15)$$

In individual-rule based inference, a logical operation is used to aggregate *the outputs* of individual rule. The output membership function $\mu_{B'(i)}$ ($i = 1 : N$) of each rule is calculated with (E-12), and by aggregating the value of all individual rule outputs, the final output can be obtained:

$$\mu_{B'}(y) = \max(\mu_{B'(1)}(y), \mu_{B'(2)}(y), \dots, \mu_{B'(N)}(y)) \quad (E-16)$$

According to Driankov *et al.* (1993), this kind of inference has proven to be more efficient in terms of computational time and memory, and is more frequently used.

E.6 DEFUZZIFICATION

The defuzzification process is a mapping from a space of fuzzy sets defined over the output universe of discourse into a space of crisp values. There are several methods to perform the process of defuzzification (Lee, 1990; Jang *et al.*, 1997; Driankov *et al.*, 1993). The most commonly found method in practice, as in the majority cases it leads to quite good results, although being computationally extensive, is the *centre of area* (COA) (Driankov *et al.*, 1993; Nauck *et al.*, 1997). This method obtains the crisp output value by applying the following formula:

$$\bar{y} = \frac{\sum_{i=1}^{\chi} y_i \cdot \mu_i}{\sum_{i=1}^{\chi} \mu_i} \quad (\text{E-17})$$

where χ indicates the number of rules, μ_i the corresponding degree of membership for each linguistic value and y_i is the centre of the i -th output fuzzy set.

E.7 FUZZY SYSTEMS TUNING AND OPTIMIZATION

The simplicity of designing fuzzy systems has been the main drive of their successful implementation. However, there remain a number of drawbacks. A fuzzy system is usually designed by incorporating an expert's implicit knowledge of the underlying process and formulate them into a set of linguistic variables and fuzzy rules. The complexity in developing these parameters increases with the complexity of the process. Fuzzy systems also consist of a number of other parameters that are needed to be selected and configured in prior, such as selection of scaling factors and configuration of the shape of the membership functions.

Due to their learning capability, neural networks are being sought in the development of neuro-fuzzy systems or adaptive fuzzy systems. Typically, the fuzzy system is transferred into a neural network-like architecture, which is then trained by some learning method, such as gradient descent or non-linear regression techniques (Jang, 1993). Neuro-fuzzy approaches are suitable for supervised learning tasks, where the objective is to minimize the error between the output of the fuzzy system and the target

value. Berenji (1992) developed a fuzzy system that is capable of learning as well as tuning of its parameters by using neural networks trained through a reinforcement learning method. Jang (1992) developed a self-learning fuzzy system based on a neural network trained by temporal back-propagation. Lee *et al.* (1995) proposed a self-organizing fuzzified basis function based on the competitive learning scheme. A more recent technique in implementing adaptive or self-tuning fuzzy systems is by using genetic algorithms (GAs), which are discussed in details in Appendix F.

APPENDIX F

GENETIC ALGORITHM

GAs provide both global and robust optimization techniques that mimics the mechanics of natural genetics (Goldberg, 1989). That all natural species can survive by adaptation is the underlying power of GAs. GAs combine a Darwinian *survival of the fittest* strategy to eliminate unfit components and use random information exchange, with an exploitation of knowledge contained in old solutions, to effect a search mechanism with surprising power and speed. GAs employ multiple concurrent search points called "chromosomes" which process through three genetic operations, reproduction, crossover and mutations, to generate new search points called "offspring" for next iterations. Such operations ensure the discovery of an optimal solution to the problem in an appropriate manner. Owing to its generality, it can be applied easily to nonlinear, discontinuous and multi-objective optimization problems that are difficult to solve using classical optimization techniques.

F.1 SINGLE OBJECTIVE GENETIC ALGORITHMS

The GA approach was first proposed by Holland (1975). Goldberg (1989) further elaborated and developed the mechanisms of a GA. With a GA, a population of individuals undergoes a sequence of unary (mutation type) and higher order (crossover type) transformations. "Good" individuals, measured by "fitness", have a higher chance to survive to the next generation. After some number of generations, the algorithm converges to the best individuals. The quality of the individuals (solutions) depends upon many factors, such as termination condition, the coverage of the population, the size of the population, and the evolution mechanisms. The structure of a simple, single objective GA is depicted in Figure F.1. Starting from generation 0, the initial population is generated either by randomly choosing from the feasible domain

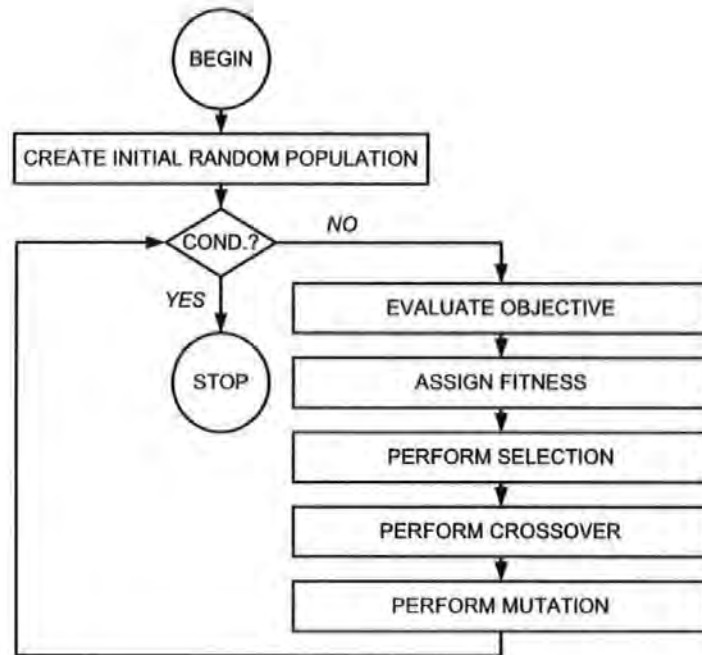


Figure F.1: A general structure of a GA

or by using a specific routine such that the initial population has certain properties. The objective of the optimization problem is evaluated using the population in the current generation. The fitness of each individual in the population is then evaluated. Individuals in the next generation are generated from the current generation based upon the fitness values. An individual with a higher fitness value gets better chance to survive to the next generation. The surviving individuals undergo some alterations by means of mutation and crossover. The fitness of altered individuals need to be evaluated. The procedure is repeated until a termination condition is met, *e.g.* the maximum number of generations or convergence of solutions.

The GA has been successfully applied to several optimization problems which are difficult to solve by conventional methods. The reason for inefficiency in the use of a conventional method include:

1. The problem is too complex or too large in size,
2. The objective functions are non-differentiable,

3. The algorithm tends to converge to local minima,

The GA is able to facilitate the solutions to these problems. It is important to note that there are other heuristic methods, such as simulated annealing (SA) or tabu search (TS) that allow solutions to these problems to be found. However, these methods are more suitable for single objective optimization problems since they deal with just one solution at a time (Fonseca and Fleming, 1995). If SA or TS are used, a multi-objective optimization problem has to be formulated as a single-objective problem prior to optimization so that the quality of a solution represented by a scalar value can be used to justify whether a solution should be accepted as a current solution (Fonseca and Fleming, 1995). In MSDF problems, where objective function of each individual sensor are different from one another, and need to be optimized simultaneously, a GA in multi-objective mode seems to be more suitable and promising since it can manage a set of solutions in the population at each generation and provide a basis for handling a set of non-dominated solutions. Also unlike in the case of SA or TS, the process to choose weights arbitrarily to aggregate multiple objectives into one single objective need not to be done. Therefore, for these reasons a multi-objective genetic algorithm is proposed in this thesis and is the subject of the next section.

F.2 MULTI-OBJECTIVE GENETIC ALGORITHMS

The form of optimization problems that can be tackled by the single objective GAs as described in the previous section is limited to:

$$\min_{\mathbf{x} \in \Omega} f(\mathbf{x}) \quad (\text{F-1})$$

where $\mathbf{x}=[x_1, x_2, \dots, x_k]$ defines the design parameters of the problem, subject to any constraint on those parameters, in the hyperspace Ω . In this case, the objective function to be minimised, f , is a scalar function of the design parameters. In most practical problems, however, several competing objectives need to be satisfied simultaneously. The multi-objective (MO) optimization problem is, the problem of simultaneously minimising the n components f_j , $j = 1, \dots, n$, of a possibly nonlinear

function F of a general design parameters \mathbf{x} in Ω ,

$$\min_{\mathbf{x} \in \Omega} F(\mathbf{x}) = \min_{\mathbf{x} \in \Omega} [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_n(\mathbf{x})] \quad (\text{F-2})$$

The MO problems usually have no unique, perfect solution, but instead a set of non-dominated alternative solutions, known as the *Pareto optimal* set (Ben-Tal, 1980), for which an improvement in one of the objectives will lead to a degradation in one or more of the remaining objectives. These solutions are also known as *non-inferior* or *non-dominated* solutions. The goals of the optimization contain the desired level (or target) of attainment associated with each objective function and are declared in a vector with dimension n . Based on some additional information, a *decision maker* (human or machine) can then choose a preferred solution. This solution is regarded as the final solution to the problem.

The solutions to MO problems can therefore be divided into two stages: an optimization stage and a decision stage. Most existing MO techniques such as ε -constraint, weighted-sum and global attainment methods, require the decision to be undertaken prior to the optimization (Hwang and Masud, 1979). In Pareto approaches (Goldberg, 1989), typically the decision is performed after the exploration of the Pareto optimal surface to present the designer with a varied set of solutions from which an appropriate compromise solution can then be selected with ease. As it is often difficult in MO problems to establish the relative emphasis on each objective *a priori*, this fact may be regarded as the benefit of the Pareto approaches. This is exactly how the multi-objective genetic algorithm (MOGA) (Fonseca and Fleming, 1995) behaves, and due to these facts, it is therefore considered as an ideal vehicle for the optimization of the fuzzy systems used in the MSDF system employed in the AUV navigation in this thesis.

Figure F.2 shows the structure of MOGA which is clearly composed of more operators than the GA shown in Figure F.1. Pareto ranking is considered to be the most important operators and it is closely associated with the graphical user interface (GUI), and these are the topic of the next section.

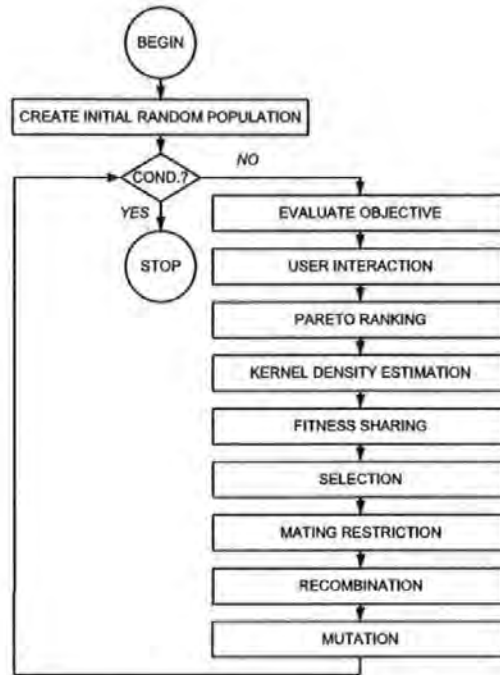


Figure F.2: A general structure of a MOGA

F.2.1 Pareto-Ranking

Pareto ranking is based upon the dominance of an individual solution in the solution hyperspace. The best individuals in the Pareto ranking are those not dominated by other individuals and in this scheme those individuals are 0-ranked (Fonseca and Fleming, 1995), while penalties are inflicted to the dominated ones according to the following equation:

$$Rank(x_i, t) = p_i(t) + 1 \quad (F-3)$$

Here, the number of individuals p_i that dominate a certain individual x_i at t -th generation establishes the rank of that particular individual. Figure F.3 demonstrates the way in which Pareto ranking is achieved in a minimisation problem for two objectives defined with equal priorities. Note that other individuals do not dominate those individuals that are ranked 0, and none of the latter is better than its counterparts.

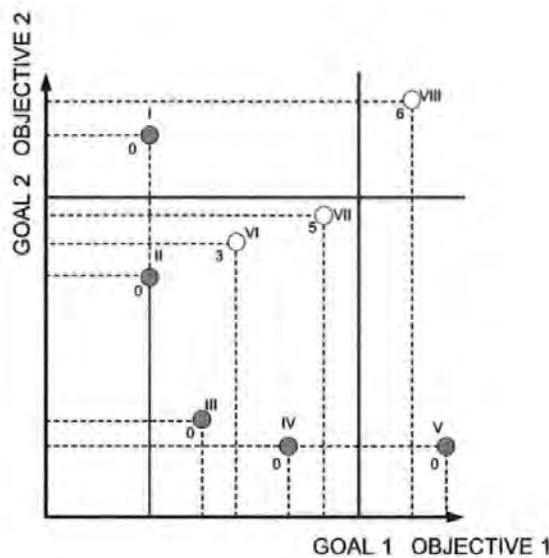


Figure F.3: Pareto ranking with goal values

In this particular example, solution II has a better value than solution III in objective 1. Whilst solution III, on the contrary has less value than solution II in objective 2. This demonstrates the concept of "trade-off", where an improvement in one objective results in a degradation in other objectives.

The non-dominated individuals in Figure F.3, however, are not distinct from each other unless priorities of the two objectives are specified. In different levels of priority the objective with the highest probability should be satisfied before searching individuals to satisfy other objectives. For example, an individual in which all goals are satisfied may be considered preferable to a nondominated one in which some components go beyond the goal boundaries. The three non-dominated individuals (II, III, IV) that are inside the goal boundaries are considered preferable than the other two non-dominated points (I, V).

The ranking of other solutions that satisfy both constraints is performed according to (F-3). To illustrate this, solution VII is ranked 5, since it has 4 other solutions (II, III, IV, VI) in its region (indicated by dashed-lines). The ranking of the solutions that do not satisfy goal 1 and 2, are performed in a similar manner, but the ranking

will start with a number of satisfying solutions incremented by 1. In the example given above, solution VIII is ranked 6. It should be noted that not all rankings will necessary appear. In the particular case considered here, no individuals is ranked 2 or 4.

The graphical representation of trade-off given in Figure F.3 is suitable only to visualise an optimization problem with two objectives. However, when the dimensionality of the solution space increases, this becomes virtually impossible. This subject is discussed in detail in the next section.

F.2.2 Trade-Off Sets Visualisation and Analysis

Fonseca and Fleming (1995) used a form of plot that allows trade-off information to be visualised. As it displays the trade-offs solutions produced by MOGAs on parallel axes, the graphs is known as "parallel co-ordinates". In its x-axis and y-axis, the objectives and the objective value are displayed. An example of this plot is shown in Figure F.4.

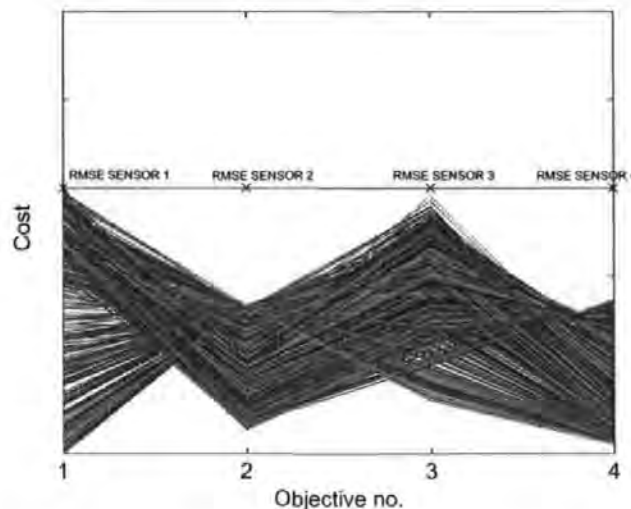


Figure F.4: The parallel co-ordinate visualisation plot

The design goal values for each objective used in this example, *i.e.*, the root mean square error (RMSE) of each sensor, are shown by the crosses and potential solutions to this optimization problem are shown by the lines displayed below the crosses. In this particular case, all solutions have achieved the goals. The concept of trade-off, in which an improvement of an objective leads to a degradation in the other, is realised in this plot by the crossing lines. In this particular example, conflicts are exhibited by objectives (3) and (4). This means that reducing the RMSE of sensor 3, objective (3), will usually result in deterioration in RMSE of sensor 4 performance, objective (4). As contradictory objectives cannot be satisfied simultaneously, an intervention by a decision maker (human or machine) to find a compromising situation is therefore needed. Fonseca and Fleming (1995) developed a GUI that facilitates the decision maker to direct the search in a desired spaces by altering the goals and priorities of the objectives and to make a final solution thereof.

APPENDIX G

CO-ORDINATE TRANSFORMATION

The application of an integrated INS/GPS techniques to provide an accurate position and orientation of AUVs requires the comprehension of three co-ordinate frames: Earth-centered Earth-fixed (ECEF), geographical/North-East-Down (NED) and body co-ordinate frame. These are briefly discussed in the following.

1. *Earth-centered Earth-fixed co-ordinate frame.* Latitude, longitude and height provided by a GPS receiver are defined in the ECEF co-ordinate frame. As shown in Figure G.1, the origin of the frame is at the mass centre of the Earth. The $X_{ECEF}Y_{ECEF}$ -plane coincident with the Earth's equatorial plane. The $+X_{ECEF}$ -axis points in the direction of 0° longitude, and the $+Y_{ECEF}$ -axis points in the direction of 90° East longitude. The X_{ECEF} - and Y_{ECEF} -axes rotate with the Earth. The Z_{ECEF} -axis is chosen to be normal to the equatorial plane in the direction of the geographical North pole.

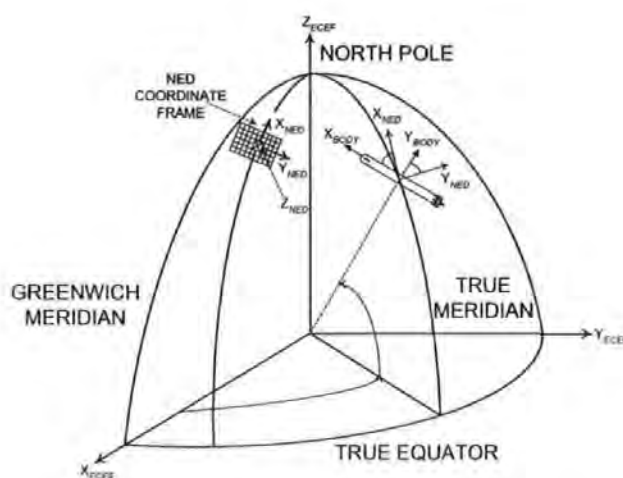


Figure G.1: AUVs navigation co-ordinate frames

2. *Geographical/North-East-Down co-ordinate frame.* This frame is defined by three orthogonal axes originating at an arbitrary local point on the ocean surface. North corresponds to X_{NED} -axis, East corresponds to Y_{NED} -axis and increasing depth corresponds to Z_{NED} -axis as shown in Figure G.1
3. *Body co-ordinate frame.* It is defined with respect to the body of the AUVs. The three axes of the AUVs are X_{BODY} -axis/longitudinal pointing in the nominal forward direction of the vehicle, Y_{BODY} -axis/lateral pointing through the right hand side of the level vehicle, Z_{BODY} -axis/downward through the nominal bottom of the vehicle. The origin of body co-ordinate frame for a submerged vehicle is at the half point along the symmetric longitudinal axis.

In an integrated INS/GPS, sensor outputs defined in the body co-ordinate frame need to be transformed into the NED coordinate frame. The same case also applies to the GPS latitude and longitude. The NED co-ordinate frame can therefore be considered as a meeting/integration point between the ECEF and the body co-ordinate frame.

G.1 TRANSFORMATION FROM BODY TO NORTH-EAST-DOWN CO-ORDINATE FRAME

Like most AUVs, the *Hammerhead* AUV uses an IMU to provide 3D body angular velocities (Figure G.2(a)) and linear acceleration, which after a single integration produces linear velocities (Figure G.2(b)). In addition, an electronic compass with a bi-axial inclinometer is used to provide 3D angular displacements defined in NED coordinate frame (Figure G.2(c)). These angular displacements are referred to as Euler angle and are used to transform the 3D body linear and angular velocities to the local NED co-ordinate frame using the following rotation order: roll (ϕ) about the X_{NED} -axis, then pitch (θ) about the first intermediate Y_{NED} -axis, followed by yaw (ψ) about the second intermediate Z_{NED} -axis. Matrix multiplication results for this transformation are included in Equation G-1

$$[\mathbf{R}_{EULER}] = [\mathbf{R}_{Z_{NED},\psi}] \cdot [\mathbf{R}_{Y_{NED},\theta}] \cdot [\mathbf{R}_{X_{NED},\phi}]$$

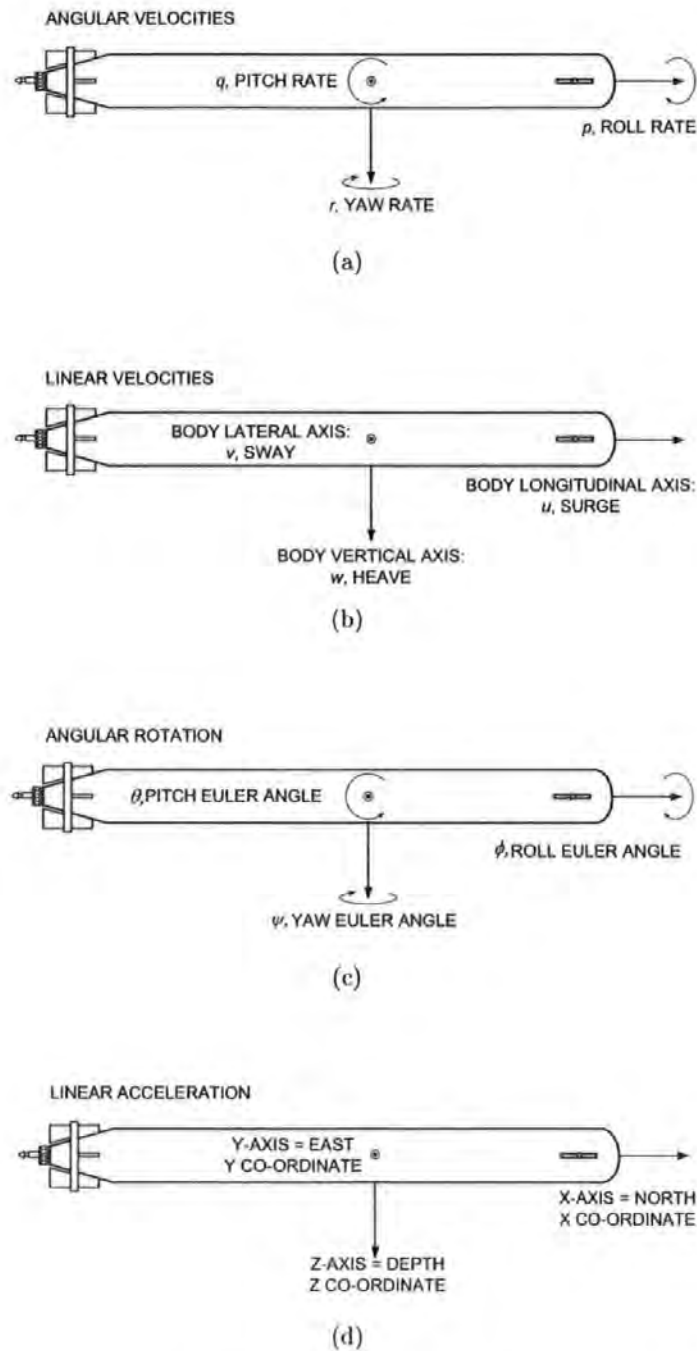


Figure G.2: (a) Angular velocities in body co-ordinate frame, (b) linear velocities in body co-ordinate frame, (c) angular rotation in NED co-ordinate frame and (d) linear translation in NED co-ordinate frame

$$= \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (G-1)$$

$$= \begin{bmatrix} \cos \theta \cos \psi & \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ \cos \theta \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix} \quad (G-2)$$

Normally Euler angle must be restricted from representing a vertical orientation or else mathematical singularities may result. Several techniques for avoiding Euler angle singularities in the vicinity of $\theta = \pm \frac{\pi}{2}(\text{rad})$ are discussed in Cooke *et al.* (1992).

The 3D NED linear velocities can be obtained from the body co-ordinate frame linear velocities by the following matrix equation:

$$\begin{bmatrix} \dot{X}_{NED} \\ \dot{Y}_{NED} \\ \dot{Z}_{NED} \end{bmatrix} = [\mathbf{R}_{EULER}] \cdot \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (G-3)$$

The 3D NED angular velocities are obtained from the body angular velocities by the following non-orthogonal linear transformations (Cooke *et al.*, 1992):

$$\dot{\phi} = p + q \sin \phi \tan \theta + r \cos \phi \tan \theta \quad (G-4)$$

$$\dot{\theta} = q \cos \phi - r \sin \phi \quad (G-5)$$

$$\dot{\psi} = \frac{q \sin \phi + r \cos \phi}{\cos \theta} \quad (G-6)$$

Equation G-4, G-5 and G-6 can be combined into matrix notation:

$$\begin{bmatrix} \dot{\phi}_{NED} \\ \dot{\theta}_{NED} \\ \dot{\psi}_{NED} \end{bmatrix} = [\mathbf{T}_{EULER}] \cdot \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (G-7)$$

where

$$[\mathbf{T}_{EULER}] = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \quad (\text{G-8})$$

The preceding equations provide a complete set of component transformations from body to NED co-ordinate frame linear and angular velocities. All component of velocities can be further grouped together and form the following matrix transformation:

$$[\mathbf{V}]_{NED} = \left[\begin{array}{c|c} [\mathbf{R}_{EULER}] & 0 \\ \hline 0 & [\mathbf{T}_{EULER}] \end{array} \right] \cdot [\mathbf{V}]_{BODY} \quad (\text{G-9})$$

where

$$[\mathbf{V}]_{BODY} = \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix} \quad (\text{G-10})$$

$$[\mathbf{V}]_{NED} = \begin{bmatrix} \dot{X}_{NED} \\ \dot{Y}_{NED} \\ \dot{Z}_{NED} \\ \dot{\phi}_{NED} \\ \dot{\theta}_{NED} \\ \dot{\psi}_{NED} \end{bmatrix} \quad (\text{G-11})$$

These velocity relationship are the so called kinematics equations of motion (Greenwood, 1988).

G.2 TRANSFORMATION FROM EARTH-CENTERED-EARTH-FIXED TO NORTH-EAST-DOWN CO-ORDINATE FRAME

In the *Hammerhead* AUV, a GARMIN GPS25-LVS is installed to acquire absolute position data in world geodetic system 1984 (WGS84). In this datum, one degree of latitude corresponds to approximately 111(*km*); therefore one minute of latitude corresponds to that number divided by 60, or approximately 1845(*m*). The length of a minute of longitude, measured along a parallel, depends upon the latitude of that parallel. The length varies from approximately 1855(*m*) at the equator to 0(*m*) at the pole. One minute of longitude corresponds to approximately 1855(*m*) multiplied by the cosine of that latitude. The conversion of latitude and longitude from deg-min-sec to meters is therefore transforming the absolute position defined in the ECEF to the NED co-ordinate frame. Readers interested in the details are referred to Kennedy (2002).

During a particular surface mission, the position of an AUV in NED co-ordinate frame derived from the latitude and longitude GPS data can therefore be easily obtained. Finally, by subtracting the initial value of the NED position from all subsequent values, integration with the NED data derived using the techniques discussed in the previous section can be performed.

REFERENCES

- Ahmad, S. M. and R. Sutton (2003). Dynamic Modelling of a Remotely Operated Vehicle. In: *Proceedings 1st IFAC Workshop on Guidance and Control of Underwater Vehicles*. Newport, South Wales, UK. pp. 47–52.
- Alspach, D. L. (1973). A Parallel Solution to the Adaptive Kalman Filtering Problem with Vector Measurements. *Computer and Electrical Engineering* 1(1), 83–94.
- Atwood, D. K., J. J. Leonard, J. G. Bellingham and B. A. Moran (1995). An Acoustic Navigation system for Multi-Vehicle Operations. In: *Proceedings of 9th International Symposium on Unmanned Unthetered Submersible Technology*. Durham, NH, USA. pp. 202–208.
- Balasuriya, B. A. A. P. and T. Ura (1999a). Multisensor Fusion for Autonomous Underwater Cable Tracking. In: *Oceans'99 MTS/IEEE*. Seattle, WA, USA. pp. 209–215.
- Balasuriya, B. A. A. P. and T. Ura (1999b). Sensor Fusion Technique for Cable Following by Autonomous Underwater Vehicles. In: *Proceedings of the 1999 IEEE International Conference on Control Applications*. Kohala Coast-Island of Hawaii, Hawaii, USA. pp. 1779–1784.
- Bellingham, J. G. (1992). Capabilities of Autonomous Underwater Vehicles. In: *Scientific and Environmental Collection with Autonomous Underwater Vehicles* (J. Moore, Ed.). pp. 7–14. MIT Sea Grant.
- Bellingham, J. G., C. A. Goudey, T. R. Consi, J. W. Bales, D. K. Atwood, J. J. Leonard and C. Chrysostomidis (1994). A Second Generation Survey AUV. In: *Proceedings of the 1994 Symposium on Autonomous Underwater Vehicle Technology*. Cambridge, MA, USA. pp. 148–155.
- Bellingham, J. G., T. R. Consi, U. Tedrow and D. Di Massa (1992). Hyperbolic Acoustic Navigation for Underwater Vehicles: Implementation and Demonstration. In:

- IEEE Symposium on Autonomous Underwater Vehicle Technology*. Washington, DC, USA. pp. 304–309.
- Ben-Tal, A. (1980). Characterization of Pareto and Lexicographic Optimal Solutions. In: *Multiple Criteria Decision Making Theory and Application* (G. Fandel and T. Gal, Eds.). Vol. 177 of *Lecture Notes in Economics and Mathematical Systems*. pp. 1–11. Springer Verlag. Berlin, Germany.
- Berenji, H. R. (1992). Learning and Tuning Fuzzy Logic Controllers through Reinforcements. *IEEE Transactions on Neural Networks* **3**, 724–740.
- Boozer, D. and W. McDaniel-Jr. (1972). On Innovation Sequence Testing of the Kalman Filter. *IEEE Transactions on Automatic Control* **AC-17**(1), 158–160.
- Bossley, K. M., M. Brown and C. J. Harris (1999). Neurofuzzy Identification of an Autonomous Underwater Vehicle. *International Journal of System Science* **30**(9), 901–913.
- Bracio, B. R., W. Horn and D. P. F. Moller (1997). Sensor Fusion in Biomedical Systems. In: *Proceedings of the 19th Annual International Conference of IEEE Engineering in Medicine and Biology and Society*. Chicago, IL, USA. pp. 1387–1390.
- Brooks, R. R. and S. S. Iyengar (1998). *Multisensor Fusion: Fundamentals and Applications with Software*. Prentice Hall. New Jersey, USA.
- Brown, R. G. and P. Hwang (1997). *Introduction to Random Signals and Applied Kalman Filtering*. John Wiley and Sons, Inc.. New York, USA.
- Carpenter, G. A. and S. Grossberg (1996). Fuzzy ART. In: *Fuzzy Engineering* (B. Kosko, Ed.). pp. 467–497. Prentice Hall.
- Carpenter, G. A. and S. Grossberg (2003). Adaptive Resonance Theory. In: *The Handbook of Brain Theory and Neural Networks* (M.A. Arbib, Ed.). pp. 79–81. MIT Press.
- Chaer, W. S., R. H. Bishop and J. Ghosh (1997). A Mixture-of-Experts Framework for Adaptive Kalman Filtering. *IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics* **27**(3), 452–464.

- Chaer, W. S., R. H. Bishop and J. Ghosh (1998). Hierarchical Adaptive Kalman Filtering for Interplanetary Orbit Determination. *IEEE Transactions on Aerospace and Electronics Systems* **34**(3), 883–896.
- Chen, G. and K. C. Chui (1990). A Modified Adaptive Kalman Filter for Real-Time Applications. *IEEE Transactions on Aerospace and Electronic Systems* **26**(5), 691–699.
- Cooke, J. C., M. J. Zyda, D. R. Pratt and R. B. McGhee (1992). NPSNET: Flight Simulation Dynamic Modeling Using Quaternions. *PRESENCE: Teleoperations and Virtual Environments* **1**(4), 404–420.
- Córdon, O., F. Herrera and M. Lozano (1998). On the Combination of Fuzzy Logic and Evolutionary Computation: A Short Review and Bibliography. In: *Fuzzy Evolutionary Computation* (W. Pedrycz, Ed.). pp. 55–77. Kluwer Academic.
- Cox, R. and S. Wei (1995). Advances in the State of the Art for AUV Inertial Sensors and Navigation Systems. *IEEE Journal of Oceanic Engineering* **20**(4), 361–366.
- Dagleish, F. R. (2004). Applications of Laser-Assisted Vision to Autonomous Underwater Vehicle Navigation. PhD Thesis. Cranfield University. Cranfield, UK.
- Dagleish, F. R., S. Tetlow and R. L. Allwood (2003). A Preliminary Experiments in the Development of a Laser Based-Imaging Sensor for AUV Navigation. In: *Proceedings 1st IFAC Workshop on Guidance and Control of Underwater Vehicles*. Newport, South Wales, UK. pp. 239–244.
- Dana, P. H. (2000). Global Positioning System Overview. World Wide Web, <http://www.colorado.edu/geography/gcraft/notes/gps/gps.html>.
- Doyle, R. S. and C. J. Harris (1996). Multisensor Data Fusion for Helicopter Guidance Using Neurofuzzy Estimation Algorithms. *The Royal Aeronautical Society Journal* pp. 241–251.
- Driankov, D., M. Hellendoorn and M. Rainfrank (1993). *An Introduction to Fuzzy Control*. Springer Verlag. Heidelberg, Germany.
- Dubois, D. and H. Prade (1988). *Possibility Theory*. Plenum Press. New York, USA.

- Eide, P. (1996). MMAE Failure Detection System for the F-16. *IEEE Transactions on Aerospace and Electronic Systems* **32**(3), 1125–1136.
- Ellowitz, H. I. (1992). The Global Positioning System. *Microwave Journal* **35**(4), 24–33.
- Escamilla-Ambrosio, P. J. and N. Mort (2001). A Hybrid Kalman Filter-Fuzzy Logic Multisensor Data Fusion Architecture with Fault Tolerant Characteristics. In: *Proceedings of the 2001 International Conference on Artificial Intelligence*. Las Vegas, NV, USA. pp. 361–367.
- Fitzgerald, R. J. (1971). Divergence of the Kalman Filter. *IEEE Transactions on Automatic Control* **AC-16**(6), 736–747.
- Fonseca, C. M. and P. J. Fleming (1995). An Overview of Evolutionary Algorithms in Multiobjective Optimization. *Evolutionary Computation* **3**(1), 1–16.
- Gelb, A., Ed.) (1989). *Applied Optimal Estimation*. The MIT Press.
- Geyer, E. M., P. M. Creamer, J. A. D'Appolito and R. G. Gains (1987). Characteristics and Capabilities of Navigation Systems for Unmanned Unthetered Submersibles. In: *Proceedings International on Unmanned Unthetered Submersible Technology*. Durham, NH, USA. pp. 320–347.
- Goheen, K. R. and E. R. Jefferys (1990). The Application of Alternative Modelling Techniques to ROV Dynamics. In: *Proceedings IEEE International Conference on Robotics and Automation*. Cincinnati, OH, USA. pp. 1302–1309.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley. Massachusetts, USA.
- Golden, J. P. (1980). Terrain Contour Matching (TERCOM): A Cruise Missile Guidance Aid. *SPIE: Image Processing for Missile Guidance* **238**, 10–18.
- Gorzalczabny, M. A. (2002). *Computational Intelligence Systems and Applications: Neuro-Fuzzy and Fuzzy Neural Synergism*. Physica-Verlag. Heidelberg, Germany.
- Greenwood, T. D., Ed.) (1988). *Principles of Dynamics*. Prentice Hall.

- Grenon, G., P. E. An, S. M. Smith and A. J. Healey (2001). Enhancement of the Inertial Navigation System for the Morpheus Autonomous Underwater Vehicle. *IEEE Journal of Oceanic Engineering* **26**(4), 548–560.
- Grewal, M. S. and A. P. Andrews (2001). *Kalman Filtering: Theory and Practice Using MATLAB*. John Wiley and Sons.
- Grewal, M. S., L. R. Weill and A. P. Andrews (2001). *Global Positioning Systems, Inertial Navigation, and Integration*. John Wiley and Sons.
- Hall, D. (1992). *Mathematical Techniques in Multisensor Data Fusion*. Artech House. Massachusetts, USA.
- Hall, D. L. and J. Llinas (1997). An Introduction to Multisensor Data Fusion. *Proceedings of the IEEE* **85**(1), 6–23.
- Hallset, J. O. (1992). A Vision System for an Autonomous Underwater Vehicle. In: *Proceedings of the 11th IAPR International Conference on Pattern Recognition*. Los Alamitos, CA, USA. pp. 320–323.
- Hanlon, P. D. and P. S. Maybeck (1998). Interrelationship of Single-Filter and Multiple-Model Adaptive Algorithms. *IEEE Transaction on Aerospace and Electronic Systems* **34**(3), 934–946.
- Hanlon, P. D. and P. S. Maybeck (2000a). Characterization of Kalman Filter Residuals in the Presence of Mismodeling. *IEEE Transactions on Aerospace and Electronic Systems* **36**(1), 114–131.
- Hanlon, P. D. and P. S. Maybeck (2000b). Multiple-Model Adaptive Estimation Using a Residual Correlation Kalman Filter Bank. *IEEE Transactions on Aerospace and Electronic Systems* **36**(2), 393–406.
- Harris, C. J., A. Bailey and T. J. Dodd (1998). Multisensor Data Fusion in Defence and Aerospace. *The Aeronautical Journal* **102**(1015), 229–244.
- Healey, A. J and D. B. Marco (2001). Command, Control and Navigation: Experimental Results with the NPS ARIES AUV. *IEEE Journal of Oceanic Engineering* **26**(4), 466–477.

- Healey, A. J and D. Lienard (1993). Multivariable Sliding Mode Control for Autonomous Diving and Steering of Unmanned Underwater Vehicle. *IEEE Journal of Oceanic Engineering* **18**(3), 1–13.
- Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. The University of Michigan Press. Ann Arbor, MI, USA.
- Holmblad, L.P. and J.J. Østergaard (1982). Control of a Cement Kiln by Fuzzy Logic. In: *Fuzzy Information and Decision Processes* (M. M. Gupta, Ed.). North Holland Publication Company. Amsterdam, Netherland.
- Huster, A., S. D. Fleischer and S. M. Rock (1998). Demonstration of a Vision-Based Dead-Reckoning System for Navigation of an Underwater Vehicle. In: *Proceedings of the Oceans '98 MTS/IEEE*. Nice, France. pp. 326–330.
- Hwang, C.-L and A. S. M. Masud (1979). *Multiple Objective Decision Making - Methods and Applications*. Vol. 164 of *Lecture Notes in Economics and Mathematical Systems*. Springer Verlag. Berlin, Germany.
- Jang, J. S. R. (1992). Self-Learning Fuzzy Controllers Based on Temporal Back Propagation. *IEEE Transactions on Neural Networks* **3**, 714–721.
- Jang, J. S. R. (1993). ANFIS: Adaptive-Network-Based Fuzzy Inference Systems. *IEEE Transactions on Systems, Man and Cybernetics* **23**, 665–685.
- Jang, J. S. R. and C. T. Sun (1995). Neuro-Fuzzy Modelling and Control. *Proceedings of the IEEE* **83**(3), 378–406.
- Jang, J. S. R., C. T. Sun and E. Mizutani (1997). *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Prentice Hall. New Jersey, USA.
- Jazwinski, A. H. (1969). Adaptive Filtering. *Automatica* **5**(4), 475–485.
- Jetto, L., S. Longhi and D. Vitali (1999). Localisation of a Wheeled Mobile Robot by Sensor Data Fusion Based on a Fuzzy Logic Adapted Kalman Filter. *Control Engineering Practice* **7**, 763–771.

- Kailath, T. (1968a). An Innovations Approach to Least-Squares Estimation, Part. I: Linear Filtering in Additive Noise. *IEEE Transactions on Automatic Control* **AC-13**(6), 646–655.
- Kailath, T. (1968b). An Innovations Approach to Least-Squares Estimation, Part. II: Linear Smoothing in Additive White Noise. *IEEE Transactions on Automatic Control* **AC-13**(6), 655–660.
- Kailath, T. (1970). The Innovations Approach to Detection and Estimation Theory. *IEEE Proceedings* **58**(5), 680–695.
- Kayton, M. (1997). *Avionics Navigation Systems*. John Wiley and Sons, Inc. New York, USA.
- Kennedy, M. (2002). *The Global Positioning System and GIS*. John Wiley and Sons, Inc.. London, UK.
- King, P. J., K. J. Burnham and D. J. G James (1994). Adaptive Kalman Filter for On-Line Parameter Estimation. *System Science* **20**(1), 61–69.
- Klir, G. J. and B. Yuan (1995). *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall. New Jersey, USA.
- Kobayashi, K., K. C. Cheok, K. Watanabe and F. Munekata (1998). Accurate Differential Global Positioning via Fuzzy Logic Kalman Filter Sensor Fusion Technique. *IEEE Transaction on Industrial Electronics* **45**(3), 510–518.
- Kohonen, T. (1988). *Self-Organization and Assoicative Memory*. Springer-Verlag. New York, USA.
- Kumar, K., D. Yadav and B.V. Srinivas (1991). Adaptive Noise Models for Extended Kalman Filter. *Journal of Guidance, Control and Dynamics* **14**(2), 475–477.
- Kwak, S. H., C. D. Stevens, J. R. Clynych, R. B. McGhee and R. H. Whalen (1993). An Experimental Investigation of GPS/INS Integration for Small AUV Navigation. In: *Proceedings of 8th International Symposium on Unmanned Unthetered Submersible Technology*. Durham, NH, USA. pp. 239–251.

- Kwak, S. H., J. B. McKeon, J. R. Clynych and R. B. McGhee (1992). Incorporation of Global Positioning System into Autonomous Underwater Vehicle Navigation. In: *Proceedings of the 1992 Symposium on Autonomous Underwater Vehicle Technology*. Washington, DC, USA. pp. 291–297.
- Lea, R. K. (1998). Control of a Tethered Underwater Flight Vehicle. PhD thesis. University of Southampton.
- Lee, C. C. (1990). Fuzzy logic in Control Systems: Fuzzy Logic Controllers, Part I and II. *IEEE Transactions on Systems, Man and Cybernetics* 20(2), 404–435.
- Lee, T. H., J. H. Nie and W. K. Tan (1995). A Self-Organizing Fuzzified Basis Function Network Control System Applicable to Nonlinear Servo Mechanisms. *Mechatronics* 5(6), 695–713.
- Liang, Y., D. X. An, D. H. Zhou and Q. Pan (2004). A Finite-Horizon Adaptive Kalman Filter for Linear Systems with Unknown Disturbances. *Signal Processing* 84, 2175–2194.
- Ljung, L. (1999). *System Identification, Theory for the User*. 2 ed.. PTR Prentice Hall.
- Ljung, L. (2001). *System Identification Toolbox User's Guide, Version 5*. The Mathworks Inc.
- Llinas, J. and E. Waltz (1990). *Multisensor Data Fusion*. Artech House. Massachusetts, USA.
- Loebis, D., F. R. Dalglish, R. Sutton, S. Tetlow, J. Chudley and R. Allwood (2003a). An Integrated Approach in the Design of Navigation System for an AUV. In: *Proceedings of MCMC 2003 Conference*. Girona, Spain. pp. 329–334.
- Loebis, D., J. Chudley and R. Sutton (2003b). A Fuzzy Kalman Filter Optimized Using a Genetic Algorithm for Accurate Navigation of an Autonomous Underwater Vehicle. In: *Proc. 6th IFAC Conference on Manoeuvring and Control of Marine Crafts*. Girona, Spain. pp. 19–24.

- Loebis, D., R. Sutton and J. Chudley (2002). Review of Multisensor Data Fusion Techniques and Their Application to Autonomous Underwater Vehicle Navigation. *Journal of Marine Engineering and Technology* **AC-15**(2), 175–184.
- Loebis, D., R. Sutton and J. Chudley (2003c). A Fuzzy Kalman Filter for Accurate Navigation of an Autonomous Underwater Vehicle. In: *Proceedings 1st IFAC Workshop on Guidance and Control of Underwater Vehicles*. Newport, South Wales, UK. pp. 161–166.
- Loebis, D., R. Sutton and J. Chudley (2004a). A Fuzzy Kalman Filter Optimized Using a Multi-objective Algorithm for Enhanced Autonomous Underwater Vehicle Navigation. *Proceedings of the Institution of Mechanical Engineers Part M* **218**(M1), 53–69.
- Loebis, D., R. Sutton, J. Chudley and W. Naeem (2004b). Adaptive Tuning of a Kalman Filter via Fuzzy Logic for an Intelligent AUV Navigation System. *Control Engineering Practice* **12**(12), 1531–1539.
- Luo, R. C. and M. G. Kay (1990). A Tutorial on Multisensor Integration and Fusion. In: *Proceedings of the 16th Annual Conference of IEEE on Industrial Electronics*. Pacific Grove, CA, USA. pp. 707–722.
- Luo, R. C., Y. Chih-Chen and L. S. Kuo (2002). Multisensor Fusion and Integration: Approaches, Application and Future Research Directions. *IEEE Sensors Journal* **2**(2), 107–119.
- Magill, D. T. (1965). Optimal Adaptive Estimation of Sampled Stochastic Processes. *IEEE Transactions on Automatic Control* **AC-10**(4), 434–449.
- Majumder, S., S. Scheduling and H. F. Durrant-Whyte (2000a). Sensor Fusion and Map Building for Underwater Navigation. In: *Proceedings of the Australian Conference on Robotics and Automation*. Melbourne, Australia. pp. 25–30.
- Majumder, S., S. Scheduling and H. F. Durrant-Whyte (2000b). Sensor Fusion and Map Building Localisation for Underwater Navigation. In: *Proceedings of the 7th International Symposium on Experimental Robotics*. Honolulu, Hawaii, USA.

- Majumder, S., S. Scheduling and H. F. Durrant-Whyte (2001). Multisensor Data Fusion for Underwater Navigation. *Robotics and Autonomous Systems* **35**, 97–108.
- Mamdani, E. H. and S. Assilian (1975). An Experiment in Linguistic Synthesis of Fuzzy Controllers. *International Journal of Man-Machine Studies* **7**, 1–13.
- Marks, R. L., S. M. Rock and M. J. Lee (1994). Real-Time Video Mosaicking of the Ocean Floor. *IEEE Journal of Oceanic Engineering* **20**(3), 229–241.
- Maybeck, P. S. and P. D. Hanlon (1995). Performance Enhancement of a Multiple Model Adaptive Estimator. *IEEE Transaction on Aerospace and Electronic Systems* **31**(4), 1240–1254.
- McGhee, R. B., J. R. Clynch, A. J. Healey, S. H. Kwak, D. P. Brutzman, X. P. Yun, N. A. Norton, R. H. Whalen, E. R. Bachmann, D. L. Gay and W. R. Schubert (1995). An Experimental Study of an Integrated GPS/INS System for Shallow-Water AUV Navigation. In: *Proceedings of the 9th International Symposium on Unmanned Unthetered Submersible Technology*. Durham, NH, USA. pp. 153–167.
- McGinnity, S. and G. Irwin (1997). Nonlinear State Estimation Using Fuzzy Local Linear Models. *International Journal of Systems Science* **28**(7), 643–656.
- Mehra, R. K. (1970). On the Identification of Variances and Adaptive Kalman Filtering. *IEEE Transactions on Automatic Control* **AC-15**(2), 175–184.
- Mehra, R. K. (1971). On-line Identification of Linear Dynamic systems with Applications to Kalman Filtering. *IEEE Transactions on Automatic Control* **AC-16**(1), 12–21.
- Mehra, R. K. (1972). Approaches to Adaptive Filtering. *IEEE Transactions on Automatic Control* **AC-17**(5), 693–698.
- Mohamed, A. H. and K. P. Schwarz (1999). Adaptive Kalman Filtering for INS/GPS. *Journal of Geodesy* **73**, 193–203.
- Naeem, W. (2004). Guidance and Control of an Autonomous Underwater Vehicle. PhD Thesis. The University of Plymouth. Plymouth, UK.

- Nauck, D., F. Klawonn and R. Kruse (1997). *Foundations of Neuro-Fuzzy Systems*. John Wiley and Sons, Inc.. Wiesbaden, Germany.
- Naylies, I. (2000). The Sensory Requirement of a PC Controlled AUV. Master's thesis. Offshore Technology Centre, Cranfield University.
- Ni, L. (2001). Fault-Tolerant Control of Unmanned Underwater Vehicles. PhD thesis. Virginia Polytechnic Institute and State University.
- Norton, N. A. (1994). Evaluation of Hardware and Software for a Small Autonomous Underwater Vehicle Navigation System (*SANS*). Master's thesis. Naval Post-graduate School. Monterey, CA, USA.
- Pham, D. T. and D. Karaboga (1991). Optimum Design of Fuzzy Controllers Using Genetic Algorithms. *Journal of Systems Engineering* **7**, 114–118.
- Rendas, M. J. and I. M. G. Lourtie (1994). Hybrid Navigation System for Long Range Navigation. In: *Proceedings of the 1994 Symposium on Autonomous Underwater Vehicle Technology*. Cambridge, MA, USA. pp. 353–359.
- Reznik, L. (1997). *Fuzzy Controllers*. Newness.
- Sangsuk-Iam, S. and T. E. Bullock (1990). Analysis of Discrete Kalman Filtering under Incorrect Noise Covariances. *IEEE Transactions on Automatic Control* **35**(12), 1304–1308.
- Scheizer, P. F. and E. J. Petlevitch (1989). Automatic Target Detection and Cueing System for an Autonomous Underwater Vehicle. In: *Proceedings of the 7th International Symposium Unmanned Unthetered Submersible Technology*. Durham, NH, USA. pp. 359–371.
- Schiller, G. J. and P. S. Maybeck (1997). Control of a Large Structure Using MMAE/MMAC Techniques. *IEEE Transactions on Aerospace and Electronic Systems* **33**(4), 1122–1131.
- Schram, G., U. Kaymak and H. B. Verbruggen (1997). Fuzzy logic Control. In: *Proceedings GARTEUR Conference on Robust Flight Control*. pp. 135–147.

- Sutton, R. and G. D. Marsden (1997). A Fuzzy Autopilot Optimized Using a Genetic Algorithm. *Journal of Navigation* **50**, 120–131.
- Sutton, R., S. D. H. Taylor and G. N. Roberts (1996). Neurofuzzy Techniques Applied to a Ship Autopilot Design. *Journal of Navigation* **49**, 410–429.
- Sutton, R., S. D. H. Taylor and G. N. Roberts (1997). Tuning Fuzzy Ship Autopilots Using Artificial Neural Networks. *Transaction of Institute of Measurement and Control* **19**(2), 94–106.
- Takagi, T. and M. Sugeno (1985). Fuzzy Identification of Systems and Its Application to Modelling and Control. *IEEE Transactions on Systems, Man and Cybernetics* **SMC-15**(1), 116–132.
- Tetlow, S. and R. L. Allwood (1995). Development and Applications of a Novel Underwater Laser Illumination System. *Underwater Technology* **21**(2), 13–20.
- Titterton, D. H. (1997). *Strapdown Inertial Navigation Technology*. Peter Peregrinus.
- Tsai, C. and L. Kurz (1983). An Adaptive Robustizing Approach to Kalman Filtering. *Automatica* **19**(3), 217–228.
- Tsonkalas, L. H. and R. E. Uhrig (1997). *Fuzzy and Neural Approaches in Engineering*. John Wiley and Sons, Inc., New York, USA.
- Varshney, P. K. (1997). Multisensor Data Fusion. *Electronics and Communication Engineering Journal* **9**(6), 245–253.
- Vazquez, J. R. and P. S. Maybeck (2004). Enhanced Motion and Sizing of Bank in Moving-Bank MMAE. *IEEE Transactions on Aerospace and Electronic Systems* **40**(3), 770–779.
- Vickery, K. (1998). Acoustic Positioning Systems: A Practical Overview of Current Systems. In: *Proceedings of the 1998 Workshop on Autonomous Underwater Vehicle*. Cambridge, MA, USA. pp. 5–17.
- Victor, J. S. and J. Sentiéro (1994). The Role of Vision for Underwater Vehicles. In: *Proceedings of the 1994 Symposium on Autonomous Underwater Technology*. Cambridge, MA, USA. pp. 28–35.

- Wang, L.-X (1997). *A Course in Fuzzy Control*. Prentice Hall. New Jersey, USA.
- Wheaton, B. J. and P. S. Maybeck (1995). Second Order Acceleration Models for an MMAE Target Tracker. *IEEE Transactions on Aerospace and Electronic Systems* **31**(1), 151–167.
- Xia, Q., M. Rao, Y. Ying and X. Shen (1994). Adaptive Fading Kalman Filter with Application. *Automatica* **30**(8), 1333–1338.
- Yeazel, Jack (2003). WAAS and its Relation to Enabled Hand-Held GPS Receivers. World Wide Web, <http://gpsinformation.net/exe/waas.html>.
- Zadeh, L. A. (1965). Fuzzy Sets. *Information and Control* **8**, 338–353.
- Zadeh, L. A. (1973). Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. *IEEE Transactions on Systems, Man and Cybernetics* **SMC-3**, 28–44.
- Zadeh, L. A. (1994). Fuzzy Logic, Neural Networks, and Soft Computing. *Communications of ACM* **37**(3), 77–84.