© 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.
Interval Kalman Filtering Techniques for Unmanned Surface Vehicle Navigation

by

Amit Motwani

A thesis submitted to Plymouth University in partial fulfilment for the degree of

Doctor of Philosophy

School of Marine Science and Engineering

In collaboration with United Technologies Aerospace Systems

September 2014
To my cat Sparkle
Abstract

Interval Kalman Filtering Techniques for Unmanned Surface Vehicle Navigation
Amit Motwani

This thesis is about a robust filtering method known as the interval Kalman filter (IKF), an extension of the Kalman filter (KF) to the domain of interval mathematics. The key limitation of the KF is that it requires precise knowledge of the system dynamics and associated stochastic processes. In many cases however, system models are at best, only approximately known. To overcome this limitation, the idea is to describe the uncertain model coefficients in terms of bounded intervals, and operate the filter within the framework of interval arithmetic. In trying to do so, practical difficulties arise, such as the large overestimation of the resulting set estimates owing to the over conservatism of interval arithmetic. This thesis proposes and demonstrates a novel and effective way to limit such overestimation for the IKF, making it feasible and practical to implement.

The theory developed is of general application, but is applied in this work to the heading estimation of the Springer unmanned surface vehicle, which up to now relied solely on the estimates from a traditional KF. However, the IKF itself simply provides the range of possible vehicle headings. In practice, the autonomous steering system requires a single, point-valued estimate of the heading. In order to address this requirement, an innovative approach based on the use of machine learning methods to select an adequate point-valued estimate has been developed. In doing so, the so called weighted IKF (wIKF) estimate provides a single heading estimate that is robust to bounded model uncertainty. In addition, in order to exploit low-cost sensor redundancy, a multi-sensor data fusion algorithm compatible with the wIKF estimates and which additionally provides sensor fault tolerance has been developed.

All these techniques have been implemented on the Springer platform and verified experimentally in a series of full-scale trials, presented in the last chapter of the thesis. The outcomes demonstrate that the methods are both feasible and practicable, and that they are far more effective in providing accurate estimates of the vehicle’s heading than the conventional KF when there is uncertainty in the system model and/or sensor failure occurs.
Contents

Abstract ........................................................................................................... i
Contents .......................................................................................................... iii
List of Figures ................................................................................................. vi
List of Tables ................................................................................................... xiii
Nomenclature .................................................................................................. xiv
Acknowledgements .......................................................................................... xv
Declaration ....................................................................................................... xvi

1 Introduction

1.1 Motivation .................................................................................................. 1
  1.1.1 Overall Springer project objectives ...................................................... 1
  1.1.2 Navigation, guidance and control for autonomous operation .............. 3
  1.1.3 Aim and objective of this research ....................................................... 4
1.2 Contributions ............................................................................................. 5
1.3 Publications ............................................................................................... 5
1.4 Outline of the Thesis ................................................................................... 8

2 Literature Review and Background Material

2.1 Unmanned Surface Vehicles ...................................................................... 11
  2.1.1 A brief history of unmanned surface vehicles ..................................... 11
  2.1.2 Current USV panorama ....................................................................... 13
2.2 Kalman Filtering ......................................................................................... 20
  2.2.1 A probabilistic approach to data fusion .............................................. 20
  2.2.2 The Kalman filter ................................................................................ 24
  2.2.3 Kalman filtering in vehicle navigation ................................................. 28
2.3 Robust Kalman Filtering ............................................................................ 30
  2.3.1 Interval Kalman filtering ...................................................................... 33
2.4 Summary ..................................................................................................... 33

3 The Springer Unmanned Surface Vehicle

3.1 Vehicle Characteristics and Hardware Overview ........................................ 35
  3.1.1 The starboard-side Peli-case ............................................................... 39
3.2 User Interface and Algorithm Execution ................................................... 45
3.3 Summary ..................................................................................................... 47
# 4 Navigation, Guidance and Control

4.1 Waypoint Tracking ................................................................. 49  
4.2 Vehicle Motion Model ............................................................. 51  
4.3 Navigation .............................................................................. 53 
   4.3.1 Kalman filtering applied to heading estimation of the Springer USV.... 53 
   4.3.2 Gyroscope and compass Kalman filter .................................... 56  
4.4 Guidance .................................................................................. 59  
4.5 Control .................................................................................... 61 
   4.5.1 PID control ........................................................................ 62 
   4.5.2 State feedback control ......................................................... 63  
4.6 Mission Simulation ................................................................. 63 
4.7 Summary .................................................................................. 68

# 5 Interval Kalman Filtering

5.1 Kalman Filter Limitations ....................................................... 70  
5.2 The Interval Kalman Filter ....................................................... 74  
5.3 Application of the IKF to the Navigation of Springer ................. 77  
5.4 Summary .................................................................................. 83

# 6 The Wrapping Effect

6.1 The Wrapping Effect ............................................................... 84  
6.2 A Hybrid Ellipsoidal-Rectangular IA Enclosure Algorithm for Recursive Interval Affine Transformations ............................................. 89 
6.3 The IKF Equations as Affine Transformations .......................... 95  
6.4 Interval Kalman Filtering using the Hybrid Ellipsoidal-Rectangular IA Enclosure Algorithm ......................................................... 99  
6.5 Summary .................................................................................. 107

# 7 Weighted Interval Kalman Filtering

7.1 The Weighted IKF ................................................................... 108  
7.2 An ANN as the Missing Link .................................................. 113  
7.3 Application to the Navigation of Springer ................................ 121  
7.4 Summary .................................................................................. 132

# 8 Multi-Sensor Data Fusion

8.1 A Fuzzy Data-Fusion Algorithm for Kalman Estimates .............. 134  
8.2 Robustness and Interval Kalman Filtering ................................ 145  
8.3 Robust and Fault Tolerant Heading Estimation ......................... 149  
8.4 Summary .................................................................................. 154
9 Experimental Verification

9.1 Experimental Verification ................................................................. 156
   9.1.1 Trials set-up and objectives ...................................................... 156
9.2 Trial set 1: weighted interval Kalman filtering .................................. 160
9.3 Trial set 2: multi-sensor data fusion .................................................. 174
9.4 Summary .......................................................................................... 187

10 Discussion, Conclusions, and Recommendations for Future Work

10.1 Discussion and Conclusions ............................................................... 188
10.2 Recommendations for Future Work .................................................... 192

References ............................................................................................. 194

Appendices
   A Bayesian Data Fusion ......................................................................... 203
   B ANN Training Procedure ................................................................... 208
   C Selected Publications ......................................................................... 210
List of Figures

Chapter 1

1.1 Elements of a navigation, guidance, and control system for autonomous vehicle steering ........................................ 4

Chapter 2

2.1 Tesla operating world’s first remote-controlled vehicle ................. 11
2.2 (a) Sea Owl, courtesy DRS Defense Solutions ................................ 13
         (b) RoboSki, a jet-ski type remotely controllable drone (courtesy US Navy) ................................................................. 13
         (c) SeaFox Mark I USV by Northwind Marine, designed to be stored and transported in confined space to be quickly deployable (courtesy of Northwind Marine) ........................................ 13
         (d) ASV’s C-Target 13 (courtesy of ASV Ltd) .......................... 13
2.3 (a) SeaDoo Challenger 2000 USV demonstrating autonomous deployment of fibre-optic cable (courtesy of Space and Naval Warfare Center Pacific) ......................................................... 14
         (b) Spartan Scout (courtesy US Navy) ........................................ 14
         (c) Protector USV (Courtesy Rafael Advanced Defense Systems Ltd) .......................................................... 14
         (d) Blackfish jet ski patrolling to protect moored ships from terrorist attacks (Courtesy QinetiQ North America) ............. 14
2.4 (a) ASV 6300 (courtesy ASV Ltd.) ............................................. 16
         (b) Wave Glider undersea view showing the two part architecture..... 16
         (c) Wave Glider with view of the solar panels (courtesy Liquid Robotics) .......................................................... 16
         (d) Harbor Wing’s prototype harnessed the winds off Hawaii, summer 2010 (courtesy Harbor Wing Technologies) ............ 16
2.5 (a) SeaSwarm prototype being tested in the Charles River (courtesy MIT Senseable City Lab) ........................................ 18
         (b) The AEOS-COT USV (courtesy Michael Lindemuth) .......... 18
         (c) VaCAS’s USV (courtesy Virginia Tech) .............................. 18
         (d) IST’s Delfin (courtesy IST) ..................................................... 18
         (e) Pinta is released for its autonomous voyage across the Atlantic, 11 Sept 2010 (courtesy Colin Sauze) .............................. 18
2.6 Author’s impression of navigating by stars ..................................... 21
2.7 Dynamic system model, where $\Delta$ is a one-step delay .................. 25
2.8 Depiction of the recursive KF process ........................................ 27

Chapter 3

3.1 The Springer USV ...................................................................... 36
3.2 (a) Cross-section of the starboard-side hull of Springer ............... 37
         (b) Top view diagram of the Springer ...................................... 37
3.3 Starboard-side Peli-Case:
   (a) detailed view of the connectors on the rear side of the case .......... 40
   (b) layout of components within the Peli-Case ............................... 40
   (c) schematic diagram of the layout within the Peli-Case and
   connections to external devices ........................................... 40
3.4 Compass units:
   (a) TCM2 ........................................................................ 42
   (b) HMR 3000 ................................................................. 42
   (c) KVH C100 .................................................................... 42
3.5 (a) Arduino Mega 2560 standard board ...................................... 43
   (b) TinkerKit gyroscope ..................................................... 43
   (c) Arduino and gyroscope as laid out within the Peli-Case ............. 43
3.6 Schematic drawing of Arduino-gyro connection ........................... 44
3.7 Normal operational mode during autonomous navigation .......... 46
3.8 Software execution ............................................................ 47

Chapter 4
4.1 Waypoint tracking mission ...................................................... 50
4.2 Block diagram representation of a two-input,two-output, USV motion model ................................................................. 51
4.3 Gyroscope measurement model: \( \Omega_t \) is the actual rate of change of
   heading of the vehicle whereas \( \Omega_0 \) is the value output by the
   gyroscope mounted on the vehicle ............................................. 55
4.4 Initial 100 time-step simulation of the way-point tracking mission:
   (a) actual turning rate of the vehicle and (noisy) gyroscope
   measurement ...................................................................... 58
   (b) actual heading of the vehicle, predicted heading based on dead-
   reckoning from gyroscope readings, (noisy) compass measurements,
   and Kalman filter heading estimate .......................................... 58
   (c) gyroscope noise ................................................................ 58
   (d) integrated gyroscope noise (random walk) ......................... 58
4.5 Deviation at time \( k \) .................................................................. 60
4.6 Vehicle trajectory during way point mission simulation .............. 65
4.7 Reference heading and vehicle heading .................................. 65
4.8 KF heading error .................................................................... 65
4.9 Control signal ....................................................................... 66
4.10 Vehicle trajectory during way point mission simulation (\( K_p = 50 \)) .... 67
4.11 Reference heading and vehicle heading (\( K_p = 50 \)) .................. 67
4.12 Control signal (\( K_p = 50 \)) .................................................... 68

Chapter 5
5.1 (a) Applied differential motor speed \( n_d \) (rpm) ......................... 73
   (b) Actual vehicle heading, compass measurements, and KF
   estimate (deg) ..................................................................... 73
5.2 KF heading rate estimate, gyroscope readings and actual vehicle
   heading rate change ................................................................ 73
5.3 IKF estimate depicting its upper and lower boundaries ............... 76
5.4 (a) System input: applied differential motor speed \( n_d \) (rpm) .......... 78
   (b) Heading estimates: nominal KF and IKF (deg) ......................... 78
5.5 (a) Differential propeller speed (rpm) ................................. 81
(b) true heading (simulated), nominal-system KF heading estimate,
IKF boundaries obtained using naive IA, improved IKF boundaries
obtained using IA with reduced dependency effect .......................... 81
5.6 (a) Applied differential motor speed \( n_d \) (rpm) ....................... 82
(b) Heading estimates by a KF based on the nominal model, an
ideal KF, an IKF using naive IA, and an IKF using the
advantageous factorisations to reduce the dependency effect (deg) ...... 82

Chapter 6

6.1 State trajectory and KF state trajectory estimate ....................... 86
6.2 Measured output \( z(k) \), noise-less output \( C x(k) \), and KF output
estimate .................................................................................... 86
6.3 Measured output \( z(k) \) and IKF interval-estimate bounds ............... 87
6.4 Successive rotations of a 2D interval vector using direct
(rectangular) IA and elliptical arithmetic ........................................ 89
6.5 Simulation of the state-vector for the interval system with nominal
poles at \( 0.4 \pm 0.25i \) using:
(a) rectangular IA .......................................................................... 92
(b) ellipsoidal arithmetic ......................................................... 92
6.6 (a) Simulation of the state-vector for the interval system with
nominal poles at 0.5 and 0.1 using rectangular IA ............................ 93
(b) Simulation of the state-vector for the interval system with
nominal poles at 0.5 and 0.1 using ellipsoidal arithmetic ................. 93
6.7 The hybrid ellipsoidal-rectangular IA enclosure algorithm .......... 95
6.8 Simulation of the nominal underdamped system with poles at with
poles at \( 0.4 \pm 0.25i \):
(a) Simulation of the state-vector for the nominal underdamped
system with poles at \( 0.4 \pm 0.25i \). States are depicted as circles for
clarity, although they represent point-values ................................. 101
(b) Rectangular IA IKF estimates of the interval system state. Only
the first 20 iterations shown, as the rectangles keep growing
exponentially .................................................................................. 101
(c) Ellipsoidal arithmetic IKF estimates of the interval system state.. 102
(d) Hybrid IA IKF estimates of the interval system state................. 102
6.9 Output of the nominal underdamped system, and IKF bounds to
the estimates of the output of the corresponding interval system
using rectangular, ellipsoidal, and hybrid IA respectively .......... 103
6.10 Simulation of the state-vector for the nominal overdamped system
with poles at 0.5 and 0.1:
(a) Simulation of the state-vector for the nominal overdamped
system with poles at 0.5 and 0.1 .................................................. 104
(b) Rectangular IA IKF estimates of the interval system state ....... 104
(c) Ellipsoidal arithmetic IKF estimates of the interval system
state .............................................................................................. 105
(d) Hybrid IA IKF estimates of the interval system state.............. 105
6.11 Output of the nominal overdamped system, and IKF bounds to
the estimates of the output of the corresponding interval system
using rectangular, ellipsoidal, and hybrid IA respectively .......... 106
Chapter 7

7.1 KF-ideal, IKF, and KF-1 estimates of the output of the system to the piecewise step input \( u(k) = a \times \text{sign}[\sin(0.02 k T_s)] \)
   (a) Applied system input............................................................... 111
   (b) System output estimates.......................................................... 111

7.2 (a) Sequence of weights calculated so that the weighted average of the IKF boundaries coincides with the KF-ideal estimate.......... 113
   (b) Innovations sequence of KF-1.................................................... 113
   (c) Innovations sequence of KF-ideal............................................... 113

7.3 RMLP used, consisting of 12 input units, 5 hidden units, and single output unit................................................................. 114

7.4 Comparison of desired weighting sequence and output of trained RMLP:
   (a) KF-1 innovations sequence and \( \Delta ikf \) ............................... 116
   (b) Comparison of desired weighting sequence and ANN output ..... 116

7.5 Performance of trained ANN on test set generated from an alternative piecewise constant input signal:
   (a) Alternative piecewise constant signal \( u(k) \) .................................... 117
   (b) KF-1 innovations sequence and \( \Delta ikf \)..................................... 117
   (c) Comparison of desired weighting sequence and ANN output...... 117

7.6 Performance of trained ANN on test set generated from a input of superimposed sinusoids:
   (a) \( u(k) = 5 \sin(0.25k) + 10 \cos(0.15k) + 10 \sin(0.08k) - 8 \sin(0.04k) + 5 \cos(0.05k) \) .............................................................. 118
   (b) KF-1 innovations sequence and \( \Delta ikf \).................................... 118
   (c) Comparison of desired weighting sequence and ANN output...... 118

7.7 Performance of trained ANN on test set generated from modified system (Equations 7.15 to 7.17):
   (a) Input signal \( u(k) \) defined by Equation 7.14............................ 120
   (b) KF-1 innovations sequence and \( \Delta ikf \) ..................................... 120
   (c) Comparison of desired weighting sequence and ANN output ...... 120

7.8 Way-point tracking mission and trajectory followed assuming knowledge of the system state and actual dynamics. The dotted line shows the ideal path, whereas the continuous line the actual path taken by the vehicle in a simulation........................................ 121

7.9 Data corresponding to simulation shown in Figure 7.8:
   (a) true and estimated rate of change of heading............................. 125
   (b) true and estimated heading angle.............................................. 125
   (c) applied differential motor speed............................................ 125

7.10 Simulation of the trajectory followed when the KF estimate is used in the control law ............................................................... 125

7.11 Way point following mission for generating training data............ 127

7.12 Data set and training results obtained from training mission:
   (a) applied differential motor speed.............................................. 127
   (b) KF, ideal KF and IKF estimates............................................... 128
   (c) innovations sequence of the nominal KF.................................... 128
   (d) IKF interval widths................................................................. 128
   (e) comparison of desired weighting sequence and trained ANN output ................................................................. 128
Comparison of desired weighting sequence and trained ANN output for:
(a) test mission 1 ............................................................... 130
(b) test mission 2 ............................................................... 130
(c) test mission 3 ............................................................... 130

Trajectory followed by the vehicle for the three test missions when
the wIKF estimates are used in the control law:
(a) Trajectory of the vehicle for test mission 1 using wIKF estimates ............................................................... 132
(b) Trajectory of the vehicle for test mission 2 using wIKF estimates ............................................................... 133
(c) Trajectory of the vehicle for test mission 3 using wIKF estimates ............................................................... 133

Chapter 8

Input and output membership functions:
(a) Input membership functions ........................................... 137
(b) Output membership functions ......................................... 137

Calculation of the output $\Delta w$ for Case 2 ($\text{SMAN} < \text{SMA} \leq 0$) .............. 139

Way point tracking mission ................................................. 141
Simulated vehicle trajectory ................................................. 142

Simulation of way-point tracking mission:
(a) actual heading, KF heading estimates, and fuzzy-logic fused KF estimate ............................................................... 143
(b) innovations sequence of each KF ...................................... 143
(c) SMA of each KF innovations ........................................... 144
(d) fuzzy weights assigned to each KF .................................... 144

Gyro measurement model: $\Omega_i$ is the actual rate of change of heading angle of the vehicle whereas $\Omega_0$ is the value output by the gyroscope mounted on the vehicle ............................................................... 145

Simulation of way-point tracking mission with imposed gyroscope bias:
(a) actual turning rate and gyroscope measurement .................. 148
(b) gyroscope bias ............................................................... 148
(c) actual heading and IKF estimate bounds ............................. 148
(d) actual heading, nominal KF estimate, and wIKF estimate ........ 148
(e) nominal KF innovations .................................................. 149

Simulation of way-point tracking mission with imposed gyroscope bias:
(a) actual heading, KF estimates, and fused KF estimates .......... 150
(b) KF innovations ............................................................. 150
(c) SMA of KF innovations .................................................. 150
(d) fuzzy weights assigned to each KF .................................... 151

Simulation of way-point tracking mission with imposed gyroscope bias:
(a) actual heading, wIKF estimates, and fused wIKF estimates ....... 153
(b) wIKF innovations .......................................................... 153
(c) SMA of wIKF innovations ............................................... 153
(d) fuzzy weights assigned to each wIKF ................................. 154
Chapter 9

9.1 Satellite image of Roadford Lake and its location in South West England ................................................................. 157
9.2 Compass angle conversion ......................................................... 159
9.3 Various photographs of Springer during trials at Roadford Lake:
   (a) Springer launch operation.................................................. 160
   (b) Team aboard a RHIB monitoring the USV mission from a short distance ................................................................. 160
9.4 Establishment of the mission plan waypoints:
   (a) Mission plan drawn on a satellite image of an area of the lake..... 161
   (b) Waypoints drawn on a rectangular grid according to their distances from the initial vehicle location................................. 161
9.5 Experiment 1.1a: actual path taken by the vehicle....................... 162
9.6 Experiment 1.1a data:
   (a) Gyroscope readings................................................................... 163
   (b) Compass measurement, wIKF and KF heading estimates.............. 163
   (c) Autopilot output................................................................. 163
9.7 Experiment 1.1b: actual path taken by the vehicle ..................... 164
9.8 Experiment 1.1b data:
   (a) Gyroscope readings................................................................... 165
   (b) Compass measurement, wIKF and KF heading estimates.............. 165
   (c) Autopilot output................................................................. 165
9.9 Experiment 1.2a: actual path taken by the vehicle....................... 166
9.10 Experiment 1.2a data:
   (a) gyroscope readings .................................................................. 167
   (b) compass measurement and KF and wIKF heading estimates ......... 167
   (c) Autopilot output...................................................................... 167
9.11 Experiment 1.2b: actual path taken by the vehicle ..................... 168
9.12 Experiment 1.2b data:
   (a) gyroscope readings .................................................................. 169
   (b) compass measurement and KF and wIKF heading estimates ......... 169
   (c) Autopilot output...................................................................... 169
9.13 Experiment 1.3a: actual path taken by the vehicle ..................... 170
9.14 Experiment 1.3a data:
   (a) Gyroscope read ................................................................. 171
   (b) Compass measurement, wIKF and KF heading estimates ............ 171
   (c) Autopilot output................................................................. 171
9.15 Experiment 1.3b: actual path taken by the vehicle ..................... 172
9.16 Experiment 1.3b data:
   (a) Gyroscope read ................................................................. 173
   (b) Compass measurement, wIKF and KF heading estimates ............ 173
   (c) Autopilot output................................................................. 173
9.17 Way-point mission established at Roadford Lake for MSDF trials... 175
9.18 Reference trial: actual vehicle trajectory (no compass failure)....... 176
9.19 Trial outcome 1: actual vehicle trajectory ................................ 177
9.20 Trial outcome 1:
   (a) Gyroscope readings ............................................................ 178
   (b) Compass measurements ...................................................... 178
   (c) wIKF and fused wIKF estimates ............................................ 178
   (d) wIKF innovations ............................................................... 179
   (e) wIKF SMAs......................................................................... 179
(f) Fuzzy weights assigned to each wIKF ........................................... 179
9.21 Trial outcome 2: actual vehicle trajectory ........................................ 180
9.22 Trial outcome 2:
(a) Gyroscope readings ........................................................................ 181
(b) Compass measurements .................................................................. 181
(c) wIKF and fused wIKF estimates ...................................................... 182
(d) wIKF innovations ........................................................................ 182
(e) wIKF SMAs ................................................................................ 182
(f) Fuzzy weights assigned to each wIKF ........................................... 182
9.23 Trial outcome 3: actual vehicle trajectory ........................................ 184
9.24 Trial outcome 3:
(a) Gyroscope readings ........................................................................ 184
(b) Compass measurements .................................................................. 185
(c) wIKF and fused wIKF estimates ...................................................... 185
(d) wIKF innovations ........................................................................ 185
(e) wIKF SMAs ................................................................................ 186
(f) Fuzzy weights assigned to each wIKF ........................................... 186
List of Tables

Chapter 3

3.1 Peli-Case rear side connectors .................................................. 41
3.2 Components within the Peli-Case ................................................. 41
3.3 Springer sensor suite for heading determination .............................. 44

Chapter 4

4.1 Waypoint coordinates .................................................................. 51
4.2 Comparison of heading estimate errors ......................................... 59
4.3 Mission performance parameters ................................................. 68

Chapter 7

7.1 Test performances for the ANN trained on data-set shown in Figure...
7.4 ...................................................................................................... 116
7.2 Coordinates of mission way points .............................................. 121
7.3 Comparison of mission performances .......................................... 126
7.4 Simulated system model coefficients ............................................. 127
7.5 Change of dynamics prescribed upon reaching the first way point
   for each of the test missions .......................................................... 129
7.6 Test performances for the trained ANN ......................................... 131
7.7 Comparison of mission performances .......................................... 131

Chapter 8

8.1 KF characteristics ......................................................................... 142
8.2 Parameter values for fusion algorithm .......................................... 144
8.3 KF and fused KF estimate errors ................................................. 145
8.4 Heading RMS errors ................................................................... 154

Chapter 9

9.1 Summary of experiments ................................................................ 162
9.2 Summary of experiment results ................................................... 174
9.3 Experiment 2.1 results ................................................................. 180
9.4 Experiment 2.2 results ................................................................. 183
9.5 Experiment 2.3 results ................................................................. 186
# Nomenclature

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AI</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>ANN</td>
<td>Artificial Neural Network</td>
</tr>
<tr>
<td>ASV</td>
<td>Autonomous Surface Vehicle</td>
</tr>
<tr>
<td>AUV</td>
<td>Autonomous Underwater Vehicle</td>
</tr>
<tr>
<td>BP</td>
<td>Back-Propagation</td>
</tr>
<tr>
<td>COA</td>
<td>Circle of Acceptance</td>
</tr>
<tr>
<td>COT</td>
<td>Centre for Ocean Technology</td>
</tr>
<tr>
<td>CRASAR</td>
<td>Centre for Robot-Assisted Search and Rescue</td>
</tr>
<tr>
<td>DGPS</td>
<td>Differential Global Positioning System</td>
</tr>
<tr>
<td>dps</td>
<td>Degrees per Second</td>
</tr>
<tr>
<td>EKF</td>
<td>Extended Kalman Filter</td>
</tr>
<tr>
<td>GD</td>
<td>Gradient Descent</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>IA</td>
<td>Interval Arithmetic</td>
</tr>
<tr>
<td>IIR</td>
<td>Infinite Impulse Response</td>
</tr>
<tr>
<td>IKF</td>
<td>Interval Kalman Filter</td>
</tr>
<tr>
<td>IMU</td>
<td>Inertial Measurement Unit</td>
</tr>
<tr>
<td>INS</td>
<td>Inertial Navigation System</td>
</tr>
<tr>
<td>INTLAB</td>
<td>Interval Laboratory</td>
</tr>
<tr>
<td>ISR</td>
<td>Intelligence, Surveillance and Reconnaissance</td>
</tr>
<tr>
<td>IST</td>
<td>Instituto Superior Tecnico</td>
</tr>
<tr>
<td>KF</td>
<td>Kalman Filter</td>
</tr>
<tr>
<td>LQG</td>
<td>Linear Quadratic Gaussian</td>
</tr>
<tr>
<td>LOS</td>
<td>Line of Sight</td>
</tr>
<tr>
<td>MAV</td>
<td>Micro-Aerial Vehicle</td>
</tr>
<tr>
<td>MEMS</td>
<td>Micro-Electro-Mechanical System</td>
</tr>
<tr>
<td>MIDAS</td>
<td>Marine and Industrial Dynamic Analysis</td>
</tr>
<tr>
<td>MLP</td>
<td>Multi-Layer Perceptron</td>
</tr>
<tr>
<td>MMSE</td>
<td>Minimum Mean Square Error</td>
</tr>
<tr>
<td>MSE</td>
<td>Mean Square Error</td>
</tr>
<tr>
<td>MPC</td>
<td>Model Predictive Control</td>
</tr>
<tr>
<td>MSDF</td>
<td>Multi-Sensor Data Fusion</td>
</tr>
<tr>
<td>NGC</td>
<td>Navigation, Guidance and Control</td>
</tr>
<tr>
<td>ONR</td>
<td>Office of Naval Research</td>
</tr>
<tr>
<td>PID</td>
<td>Proportional Integral Derivative</td>
</tr>
<tr>
<td>RHIB</td>
<td>Rigid-Hulled Inflatable Boat</td>
</tr>
<tr>
<td>RMLP</td>
<td>Recursive Multi-Layer Perceptron</td>
</tr>
<tr>
<td>RMS</td>
<td>Root Mean Square</td>
</tr>
<tr>
<td>SI</td>
<td>System Identification</td>
</tr>
<tr>
<td>SISO</td>
<td>Single Input, Single Output</td>
</tr>
<tr>
<td>SLAM</td>
<td>Simultaneous Localisation and Mapping</td>
</tr>
<tr>
<td>SMA</td>
<td>Simple Moving Average</td>
</tr>
<tr>
<td>UGV</td>
<td>Unmanned Ground Vehicle</td>
</tr>
<tr>
<td>UKF</td>
<td>Unscented Kalman Filter</td>
</tr>
<tr>
<td>USV</td>
<td>Unmanned Surface Vehicle</td>
</tr>
<tr>
<td>VaCAS</td>
<td>Virginia Centre for Autonomous Systems</td>
</tr>
<tr>
<td>wIKF</td>
<td>Weighted Interval Kalman Filter</td>
</tr>
</tbody>
</table>
Acknowledgements

I have always liked building things, gadgets and so forth, and this PhD certainly has indulged me in that respect: I have had over three years to tinker and play with a life-sized robotic boat to my heart’s content. I feel princely when, in need of some or the other contraption built, all I have to do is ask and it appeareth as if by magic: thank you to the team of dedicated technicians – Mike Sloman, Bill Stephenson, Greg Nash, and Bob Williams, I cannot emphasize enough how much the success of this project depended on you.

Many thanks to all the staff at the university who lent their support: Lucy Cheetham and Carole Watson (who welcomed me at the university even before I got there), Francesca Niedzielski (for keeping a steadfast eye on my successful progress), the staff at the School Office led by the ever resourceful Barbara Fuller who I am convinced can resolve just about any logistic or administrative quandary us students seem to get ourselves into, and the team from the Graduate School, especially Sarah Kearns for her neverending encouragement.

Thank you to all of you whom I have had the pleasure to meet and befriend within different university circles. In particular my colleagues at Buckland House – you have made my day to day so much more lively.

My thanks to my co-researchers on the Springer project: George Terzakis, whose insightful mathematical discussions and coding skills been invaluable, and Andy Annamalai, who is … one of a kind. Thanks also to our colleagues from UCL, Wenwen Liu and Rui Song. I have enjoyed working with all of you.

I would like to thank everyone in the MIDAS team. In particular, a big thank you to my supervisors: Dr Culverhouse, for his continued support, Prof. Sutton, the soul of this project and without whom it would be difficult to imagine a Springer and such a close-knit team, and my Director of Studies, Dr Sharma, who has quietly nudged me back on track whenever I have tended to stray from it (i.e. most of the time) – without your guidance I would have lost sight of my aim long ago.

Finally thank you to my family and friends who have always stood by me even during the times I was too busy to pay them the attention they deserve.
Declaration

At no time during the registration for the degree of Doctor of Philosophy has the author been registered for any other University award without prior agreement of the Graduate Committee.

Work submitted for this research degree at the Plymouth University has not formed part of any other degree either at Plymouth University or at another establishment.

This study was financed with the aid of a studentship from the Engineering and Physical Sciences Research Council, United Kingdom.

Relevant scientific seminars and conferences were regularly attended at which the work carried out was presented. A list of publications and presentations can be found in Section 1.3.

Word count of main body of thesis: 41,200

Amit Motwani

26 September 2014
Chapter 1

Introduction

“If one does not know to which port one is sailing, no wind is favourable.” — Lucius Annaeus Seneca

This chapter outlines the objectives of the research and presents an overview of the concepts that are developed throughout the thesis without delving into technical details. The main contributions of the research and a list of resulting publications are included in this introductory chapter.

1.1 Motivation

Autonomous vehicle operation is a challenging task. It is so because in practice such vehicles must operate in ever changing and unforeseeable circumstances. In a controlled environment, where every parameter is carefully monitored, in which systems behave as the mathematics dictate, sensors reliably convey the truths they see, and where even the uncertainty is certain, of course autonomy is possible. The challenge today is in operating these vehicles within the unpredictability and variability of real world environments. To achieve this, mathematically optimal operational strategies alone are not enough. Vehicles must be intelligent: they must be able to learn and adapt to circumstances they were never programmed to find themselves in in the first place. In this sense, the greater the degree of intelligence, adaptive capabilities, fault tolerance and general robustness built into such vehicles, the lesser the degree of human supervision, and ultimately intervention, necessary in their operation.
1.1.1 Overall *Springer* project objectives

In a previous project started in 2004, which saw the *Springer* unmanned surface vehicle (USV) (Chapter 3) designed and built, autonomous waypoint following was already achieved. That is, under the expected circumstances. In 2011 a follow-on project was initiated with the aim of addressing the question, and what about in unexpected circumstances?

It should be recognised that the problem being addressed is so wide and multifaceted that the aim of the current *Springer* project, of which this thesis is a product, is not to address the complete unknown. It is merely to relax the constraints of perfect knowledge and ideal situation. Although it may not seem much, it is indeed a step toward that ideal of true autonomy, and in any case, allows a certain reduction in the necessary degree of human intervention in the operation of such vehicles.

Concretely, the current *Springer* project focuses on three distinct aspects that expand its operational limitations. The first is motivated by the question, “what if GPS reception for some (any) reason becomes unavailable?”, for it had thus far been assumed that the vehicle would always be in receipt of a GPS fix to allow it to localise itself. In order to address this problem, a visual simultaneous localisation and mapping (SLAM) subsystem is being designed for the vehicle to allow it to self-localise by matching visible features of its environment to those already present in its map. This in itself is an extremely complex task and is the subject of another thesis.

Autopiloting, as briefly outlined in the next section, makes use of a control algorithm to adequately steer the vehicle. Application of optimal control strategies that offer superior performance compared to conventional controllers, in particular model based predictive control (MPC), to the autonomous piloting of vehicles such as USVs has been a topic of interest for some time. Again however, the mathematics has already been solved and demonstrated. The challenge lies in maintaining its performance even when the initial model assumed to describe the vehicle’s dynamic behaviour varies over time in an unforeseeable manner. To address this, the second area of research of the current *Springer* project delves into making such controllers adaptive to variations in the vehicle’s dynamic characteristics, as would presumably occur, for instance, if its payload suddenly changed. Again, the details of this research topic belong to the pages of another thesis.
It was the invention of the Kalman filter (KF) that first enabled spacecraft to circumnavigate the moon, by providing accurate position estimates. It was also used to estimate the heading of the Springer USV based on information from noisy sensors, as is briefly outlined in the next section. However, the noise processes must be completely known. In other words, the stochastic descriptions of the random processes must be completely accurate, not to mention, of course, that the deterministic counterparts must be known precisely. Although this may be true in some cases, in many others it is an ideal scenario, giving rise to the question of whether the filtering process can be adapted so that it retains its effectiveness even when the hypotheses of certainty are not met. This motivates the third and final research topic of this project into robust filtering techniques based on the use of the so called interval Kalman filter (IKF) for the heading estimation of the Springer USV, and is what this thesis is about.

1.1.2 Navigation, guidance and control for autonomous operation

A basic block diagram of the navigation, guidance and control (NGC) systems for the autonomous steering of the Springer USV is illustrated in Figure 1.1. Details of each of these systems will be given in Chapter 4, and only a basic description to establish the purpose of each and the interactions between them is given herein.

The overall aim being to automatically steer the vehicle, the first subsystem, the navigation system, determines the current heading of the vehicle. It does so by using information from sensors such as magnetic compasses or gyroscopes. Usually sensor readings are noisy, and so some filtering process is used. The KF (Chapter 2) does precisely this, but is dependent upon precise modelling of the sensors. The focus of this thesis is in developing a robust version of the KF known as the IKF (Chapter 5) which accomplishes the same task as the KF, but is, in addition, robust to uncertainties in the models used.

The second of these subsystems, the guidance system, generates the desired or reference heading. It does so by evaluating both the current position or location of the vehicle, and the target position. The first must be updated continuously as the vehicle moves, and is obtained, for example, via a GPS receiver (information which may indeed be also processed by some filtering algorithm). The second is specified in the mission plan.

Finally, once the reference heading and actual heading of the vehicle are known, it is the task of the autopilot, or control system, to generate the necessary
stimulus, or input to the plant, to adequately steer it in the desired direction. In this case, the plant being the USV, the input that is controlled is the difference in revolution rates of its two motors which generates a torque (Chapter 3), provoking the turning of the vehicle. This sort of control is known as feedback control because it is based on measuring the reaction of the vehicle, which may not only be consequence of the control action, but also of environmental disturbances.

Figure 1.1 Elements of a navigation, guidance, and control system for autonomous vehicle steering

1.1.3 Aim and objectives of this research

As mentioned in Section 1.1.1, this work concerns investigating and developing the interval Kalman filtering paradigm for state estimation of uncertain systems. The aim is to evolve an effective algorithm that uses the advantages of the IKF to provide a robust heading estimator for the navigation system of the Springer USV.

The IKF has thus far seen extremely limited usage and the proposed applications have been limited to theoretical or simulated scenarios. Thus in order to achieve the primary aim of this research, the objectives pursued are to investigate its practical applicability, develop the necessary methods to circumvent the difficulties of its implementation, carry out said implementation on the Springer USV in a feasible manner, and demonstrate in a series of real-time trials the advantages of the method, exploiting the vehicle’s available sensors. Dissemination of the acquired knowledge through publications, including the writing of this thesis, also forms part of the objectives.
1.2 Contributions

The investigation carried out has resulted in the following contributions to the current state of knowledge regarding IKF techniques.

- The IKF equations have been reformulated in a way that makes it possible to employ ellipsoidal arithmetic in their computation. Furthermore, a hybrid ellipsoidal-interval arithmetic enclosure algorithm has been proposed to reduce the overestimation of the IKF intervals and obtain stable estimate bounds.

- A technique based on the application of artificial neural networks (ANNs) has been developed to infer an adequate weight with which to average the IKF bounds in order to obtain useful point-valued estimates (weighted IKF (wIKF)).

- A multi-sensor data fusion (MSDF) algorithm that uses fuzzy logic techniques and which is compatible with the wIKF developed has been proposed to construct a robust and fault-tolerant heading estimator by exploiting sensor redundancy.

- To the best of the author’s knowledge, the outcome of this work constitutes the first practical implementation and demonstration of the IKF in a real world system, thus establishing its feasibility and integrability with the other (guidance and control) subsystems.

1.3 Publications

In pursuing the research objectives, the findings of the previous section have and are in the process of being disseminated in a series of publications including journal papers, conference proceedings, and technical reports and notes, as well as through presentations, seminars, and media interviews, and through the maintenance of a dedicated project website. The following comprises a list of papers published or submitted for publication.

Journals papers


Conference Papers and Presentations


**Technical Reports and Notes**


Project Website

http://www.tech.plymouth.ac.uk/sme/springerusv/2011/Springer.html

1.4 Outline of the Thesis

Following on from this introductory chapter, Chapter 2 provides a review of the publicly available literature related to the topics of this thesis. It is divided into three sections: firstly, a review of USV technology in general and a portrayal of the current panorama of the kinds of capabilities that are currently being developed in such vehicles as well as further advances being pursued for new potential applications. The second part provides a background of Kalman filtering with particular attention to its development as a navigational tool. Lastly, the chapter looks at the various developments in robust state estimation techniques. Though the IKF is identified as the most direct and natural extension of the KF to the field of robust filtering, its limited use to date supports the need for detailed investigation and the development of methods to help exploit its full potential.

Chapter 3 provides a necessary background concerning the Springer USV, its hardware and its sensors, and the software architecture developed for this project. Chapter 4 in turn provides details of each of the three major
subsystems that make autonomous operation of the vehicle possible, namely, the 
navigation, guidance and control subsystems, already briefly discussed in the 
present chapter. It also describes the constitution of the waypoint tracking 
mission paradigm used throughout this thesis, as well as the dynamic 
characteristics of the vehicle's motion, essential for carrying out simulation 

studies.

Chapters 5 to 7 form the core chapters of this research. The first of these 
formally introduces the IKF algorithm, and attempts to apply it to the problem 
of obtaining estimate bounds to the USV heading. However, in doing so the first 
fundamental difficulty regarding its implementation, its (typically) excessive 
over-conservatism, is brought to light, and understanding the reason for and 
overcoming this difficulty is the topic of Chapter 6.

Chapter 7 then addresses another practical question, and that is, what to do 
with an interval estimate. Clearly in order to integrate with the control system, 
a single value must be selected from the interval estimate of the heading that 
the IKF provides. This is accomplished with the wIKF, where the initial “w” 
stands for weight, and which provides an adequate point-valued estimate of the 
heading by computing a weighted average of the IKF bounds. The method, 
which can be argued to confer a degree of intelligence to the filter, is based on 
the use of ANNs.

By this stage the wIKF is shown to provide a robust alternative to the KF for 
heading estimation. Chapter 8 can be thought of as an addendum, in which a 

fault tolerant multi-sensor data fusion (MSDF) algorithm to automatically 
reject faulty compasses is devised. This algorithm is initially built to operate 
with ordinary KFs but then adapted to work with wIKFs as well, providing a 
simultaneously robust and fault tolerant solution for the heading estimation of 
Springer, utilising its multiple magnetic compass units.

The penultimate chapter describes the experimental verification of the wIKF 
and MSDF algorithms previously described during trials undertaken with the 
vehicle at a lake in north Devon. The last chapter discusses the research 
outcomes, draws conclusions and provides suggestions for further work.
Chapter 2

Literature Review and Background Material

“You do not see there a wireless torpedo; you see there the first of a race of robots, mechanical men which will do the laborious work of the human race.” — Nikola Tesla

This chapter is dedicated to surveying the available literature closely related to the material and concepts of this thesis, as well as provide an essential background to the KF. It is divided into three sections: the first is an overview of USVs, with a brief historical background followed by a portrait of the current scenario of USV developments worldwide. The second provides an insight into the principles of the KF, and gives an overview of applications involving Kalman filtering with emphasis on surface vehicle navigation. Lastly, the issue of robustness is addressed, and the research being carried out in robust estimation techniques is broadly classified, allowing the reader to situate the interval Kalman filtering approach within this wider field of study, understand its motivation and establish its scope. A brief description of the IKF is given, although mathematical details are deferred to Chapter 5. With respect to the first section, a more exhaustive survey of USVs can be found in Motwani (2012).
2.1 Unmanned Surface Vehicles

2.1.1 A brief history of unmanned surface vehicles

Perhaps the earliest record of an unmanned vehicle dates back to the ancient Greek mathematician Archytas of Tarentum, who around 400 BC is believed to have constructed a wooden steam powered "pigeon". Capable of flying up to 200 m before running out of steam, this robotic bird is often regarded as the first self-propelled machine (Gellius, 1927). It is however not until the 1860’s that one finds the first self-propelled vehicle which incorporated an onboard control system, in the shape of a torpedo developed by the British engineer Robert Whitehead. Whitehead designed a self-regulating mechanism that maintained the torpedo at a constant preset depth using a hydrostatic valve and pendulum balance connected to a horizontal rudder. Later on, he would incorporate Ludwig Obry’s newly invented gyroscope for azimuth control to fix the torpedo's direction (Kirby, 1972).

But probably the most anecdotal invention is the radio-controlled boat built by Tesla at a time in which radio waves were still largely unknown. At a demonstration in New York’s Madison Square Garden in 1989, he astounded crowds by remotely steering his boat in a pool of water, an act so apparently magical that some in the crowd speculated on a trained monkey being hidden inside the vessel! (Figure 2.1) (Soule, 1956)

Though radio-control was further developed during the First and Second World Wars (Soviet teletanks, the British QueenBee target-drone radio-controlled aircraft, German radio-controlled missiles and, later on, FL-Boote radio-controlled motor boats filled with explosives to attack enemy shipping), radio-control technology mostly remained stagnant. That is, up until the latter half of the twentieth century, which saw the advent of solid-state electronics, the start of the Space Age (with the launch of Sputnik in 1957), and the frenzied race that followed to deliver satellites (all of which are radio-controlled) into orbit.

![Figure 2.1 Tesla operating world’s first remote-controlled vehicle.](image)
During the post-war era, development and operation of navy USVs for the most part consisted of simple, radio-controlled drone boats used for battle/bomb damage assessment, target practice for manned vessels, and as tools for dangerous mine clearance operations. Post-war Britain saw the need for fast target craft, and so, many military vessels that were no longer needed for war missions were converted to radio control. For example, the RAF converted four of their 68 ft High Speed Launch vessels to Remote Controlled Target Launches in 1949 (ASR-MSC, 2012). And in the 1960's, the US Navy deployed drone ships for minesweeping operations in waters around Vietnam (US Navy, 2007). Although over the next few decades large drone ships fitted with radio control capability came into service and are still operational in several of the world’s navies (eg. the German Troika mine countermeasure craft known as the Seehund commissioned in the 1980s, or the Danish Hirsholm and Saltholm mine clearance warships deployed since 2007/2008 (Balsved, 2008)), a new concept in USV design began in the 1980s that would reshape the course of USV development.

In 1985, a private enterprise known as Robotics Systems Inc. developed the Owl. Unlike the large ships based on traditional warship design and subsequently fitted with remote control capabilities, the Owl was a small 10ft long ski-jet type craft designed specifically to operate remotely. With a low profile fibreglass hull for increased stealth and payload, it offered high manoeuvrability and mobility in riverine and coastal waters, where larger craft cannot operate effectively, making this a desirable new kind of military asset. With the new possibilities offered by this type of craft, the US Navy’s interests in using USVs for intelligence, surveillance and reconnaissance (ISR) missions started emerging. Since the original Owl, there have been numerous redesigns commissioned by the US Office of Naval Research (ONR), such as Owl MK II built by Universal Secure Applications. Equipped with day time and thermal cameras and with enhanced off-ship launch and recovery systems, it was deployed from ships in the Middle East for force protection missions from 1993 to 2000. It was also used to detect live mines in shipping lanes off Kuwait towing a side-scan sonar (Universal Secure Applications, 2010). Another descendant, the Owl MK VI, or Sea Owl, developed by US Navy defence contractor DRS Technologies, can be equipped with a large range of sensors such as cameras, sonar, radar, microphones and speakers, weapons, and environmental sensors, intended for a variety of applications (Figure 2.2a).
2.1.2 Current USV panorama

The conception of the *Owl* sparked the development of a whole range of *Owl-type* tele-operated USVs, such as the *Roboski* (Figure 2.2b) in the 1990’s and the *Sea Fox* (Figure 2.2c) in the early 2000’s, featuring improvements in speed and agility (Bertram, 2008). In the UK, ASV Ltd created a series of small high speed target drones, the C-Target series (Figure 2.2d) (ASV, 2014). However, whilst focusing on improving hull design for speed, manoeuvrability, or stealth, and sensor or weapon bearing capacity, these vessels remained largely remotely operated drones.

![Image of Sea Owl](image1)

(a) *Sea Owl*

![Image of RoboSki](image2)

(b) *RoboSki* jet-ski type USV

![Image of SeaFox Mark I](image3)

(c) *SeaFox Mark I*

![Image of C-Target 3](image4)

(d) C-Target 3

Figure 2.2 (a) *Sea Owl*, courtesy DRS Defense Solutions; (b) *RoboSki*, a jet-ski type remotely controllable drone (courtesy US Navy); (c) *SeaFox Mark I* USV by Northwind Marine, designed to be stored and transported in confined space to be quickly deployable (courtesy of Northwind Marine); (d) ASV’s C-Target 13 (courtesy of ASV Ltd).

It was with the SS *San Diego* project, an initiative of the Space and Naval Warfare Systems Centre, San Diego, that attention was focused on developing platform-independent USV technology. They chose the *Seadoo Challenger 2000*, a commercially available jet-engine driven recreational sport boat platform, as a test-bed or host on which technology could be developed and tested, and then transitioned to other USVs. One of the initial aims was to leverage technology already developed for unmanned ground vehicles (UGVs) and to adapt and apply these to USVs, due to the similar two dimensional nature of ground and surface navigation. The technologies adapted from UGVs include tele-operated
control, navigational sensor fusion using Kalman filtering, waypoint navigation, and multi-vehicle command and control: in 2004, these were demonstrated via autonomous deployment of fibre-optic cable on the ocean floor at 35 knots using GPS waypoint navigation (Figure 2.3a) (Nguyen and Everett, 2006). Further objectives are to develop robust autonomous capabilities, from simple waypoint navigation to deliberative and reactive obstacle avoidance and path planning using digital nautical charts, marine radar systems, and monocular and stereo vision sensors (Larson et al, 2006).

Other USVs currently being developed for military purposes include the Spartan Scout and Sentinel in the US, the Barracuda and Hammerhead by Meggitt Training Systems of Canada, the Protector by Israel’s Rafael Advanced Defence Systems Ltd, the Inspector by French based company ECA Robotics, and the Blackfish by British defence contractor QinetiQ (Motwani, 2012). All of these are small, mostly RHIB-based platforms, intended to demonstrate military force protection capabilities, hosting an array of sensors and weapon systems.

Aside from the heavy investment by military institutions, USVs are also being developed now for commercial, industrial, and research purposes. For example, ASV Ltd markets several USVs for civilian and research use. Their 6.3 m
ASV6300 (Figure 2.4a), a stable, long endurance work class USV that is actually a semi-submersible vessel, is designed for a wide range of applications. It is especially suited to stable sea-keeping and can be equipped with various sensors such as a multibeam echo sounder, side-scan sonar, an integrated conductivity-temperature-depth sensor, a pan-tilt-zoom camera, and a sub-bottom profiler.

An important market for USVs is the offshore oil and gas industry. In fact, an article in the Norwegian Centre for Ships and Ocean Structures’s 2008 annual report suggested that USV technology would be key in future hydrocarbon exploration and exploitation in that country (Breivik, 2008). ASV Ltd recently launched what they are calling the world’s first unmanned oil and gas workboat. The C-Worker 6 has been designed to be able to conduct precise subsea positioning in rough open seas, and could potentially save the industry millions (ASV, 2014). As in any industry, the replacement of expensive manned surface vehicles for low-cost USVs is an attractive alternative, not only for the reduced personnel costs, but also to reduce the risks posed to on-board personnel in certain conditions, as well as enabling broadening the possible field of exploration in a cost-effective manner.

Such are the objectives of the Wave Glider, launched in 2008 by Liquid Robotics Inc., a unique platform which continuously harvests energy from the environment. It consists of a two-part architecture and wing system which converts wave motion into thrust (Figure 2.4b), whilst solar panels provide electricity for sensor payloads (Figure 2.4c). The Wave Glider can operate for a year without recharging its battery, and so can be deployed for extended monitoring missions, including that of offshore energy projects, from their exploration and production phases to their long-term supervision; for example, BP relies on them to monitor water quality near the defunct Macondo well in the Gulf (Liquid Robotics, 2012).

Other examples of energy-harvesting USV projects are the Aquarius and HWT X-1 USVs. The Aquarius USV is being developed as part of the expanding research programme of Eco Marine Power Co. Ltd of Fukuoka, Japan. Designed to have a shallow draft and low height, the Aquarius will tap into wind and solar renewable energy sources to reduce fuel consumption and allow it to operate almost silently when required. Typical missions envisaged are monitoring harbour pollution, oceanographic surveys, and marine data collection. (Eco Marine Power, 2014).
The HWT X-1 concept vessel (Figure 2.4d), developed by Harbor Wing Technologies, is a wing-sailed catamaran that uses wind as its main propulsion force. The computer-controlled sail, called WingSail, is oriented independently of the hulls to provide a constant angle of attack relative to the wind and produce a forward thrust. With its innovative design, the vehicle is capable of high manoeuvrability and station keeping, making it suitable for everything from monitoring enemy submarine activity to tracking endangered marine mammals. The vessel uses GPS waypoint navigation with Line of Sight (LOS) guidance and conventional control algorithms to keep it on course (Elkaim and Boyce, 2008; Harbour Wing, 2014).

Building upon the concept of environmentally friendly USVs, such vehicles are also being proposed to perform environmental clean-up operations. A team at the MIT Senseable City Laboratory has developed the Seaswarm, a fleet of autonomous surface vessels (ASVs) each consisting of a photovoltaic powered conveyor belt capable of propelling itself while absorbing oil from the sea surface (Figure 2.5a). These vessels are designed to communicate their location through GPS and WiFi and work together autonomously in an organised fashion to clean oil spills: when the edge of a spill is detected, Seaswarm moves inward until the oil has been cleared from the site before moving on to join other vehicles that are still cleaning. The oil is consumed locally in the cleaning
process, so that the vehicles can operate continuously, making this an extremely efficient system (Senseable, 2014).

At present, the research community is keen to use USVs to deploy instruments at sea for data collection in ways that were previously intractable, as they provide increased accessibility in a cost-effective way. But further to simply using USVs to deploy instrumentation for research purposes, USVs are being designed and reinvented by researchers in academia themselves. Presently there are many academic research projects involved in the development of USVs, striving for better designs, new navigation and control techniques, and better suited vessels for specific applications. For example, the Centre for Ocean Technology (COT) of the University of South Florida (USF) St. Petersburg has been developing USVs since 2001. Their current USV, known as the AEOS-1, is a catamaran design with a central T-shaped chassis that supports the instrumentation, power, communications and the control system, and has been specifically designed for deploying a wide range of environmental and oceanographic instrumentation also under development at USF (Figure 2.5(b)) (Steimle and Hall, 2006).

As another example, the Virginia Centre for Autonomous Systems (VaCAS), a research centre of the Virginia Tech University, has been developing a 15.7 feet long RHIB type USV (Figure 2.5c) since 2008. The USV is intended to be used as a platform to develop methods for autonomous navigation of river environments. The vessel uses a laser-line scanner, optical cameras, and GPS to navigate a river system even in the presence of incomplete or misleading map data. A recent test at Peak Creek, VA, demonstrated the USV’s ability to safely navigate a 4 km stretch of the river in only 25 minutes (VaCAS, 2011).

Other vehicles being developed by research communities include the Delfim (Figure 2.5d) and the Caravela, developed at the Dynamical Systems and Ocean Robotics (DSOR) laboratory of the Instituto Superior Tecnico (IST) in Lisbon. Delfim is a small autonomous catamaran designed as a prototype vehicle to proof test the concept of an autonomous surface vehicle (ASV) capable of working in close cooperation with an AUV. Vehicle navigation relies on a high precision DGPS and an attitude sensor. Currently they are working on the Delfim-x, a successor of the original Delfim, designed for increased autonomy and improved hydrodynamic characteristics. The Caravela, developed together with industrial partners, is a long range autonomous oceanographic vessel which uses RF/satellite communication for sending mission sensor data and receiving mission commands, and an integrated NGC system that allows it to follow predetermined paths with great accuracy (Pascoal et al, 2006).
In recent years the development of autonomous sailboats has been sparked by various robotic sailboat competitions. Two of these are the World Robotic Sailing Championship and the transatlantic race known as the Microtransat Challenge, competitions which push participating teams to improve their entrees year on year, promoting research in the field and contributing to the rapid technical advancement of these types of vessels. The Intelligent Robotics Group at Aberystwyth University in the UK has been developing sailing robots for long-term autonomous oceanographic monitoring over the past few years (Sauze and Neal, 2008). Their Beagle-B has demonstrated the efficacy of its 4 m carbon composite wing-sail design during trials in which it out-sailed its chasers running on traditional sails prone to collapse during light winds. Their Pinta boat (Figure 2.5e), built specifically for racing, took part in the Microtransat Challenge in 2010, attempting to cross the Atlantic and setting a record after sailing for about 350 nautical miles before it is suspected to have capsized (BBC News Wales, 2010), a record as yet unsurpassed.

![Figure 2.5 (a) SeaSwarm prototype being tested in the Charles River (courtesy MIT Senseable City Lab) (b) The AEOS-COT USV (courtesy Michael Lindemuth); (c) VaCAS's USV (courtesy Virginia Tech); (d) IST’s Delfim (courtesy IST); (e) Pinta is released for its autonomous voyage across the Atlantic, 11 Sept 2010 (courtesy Colin Sauze).](image)

Public administrations are also interested in deploying USVs to take over a variety of tasks. For example, a report by the Office of Bridge Technology of the Federal Highway Administration in the US suggested the use of USVs for
deploying portable scour measuring systems for monitoring scour-critical bridges. This came about because it is believed that the number of scour-critical bridges in the US is far greater than can be replaced or repaired, and therefore their monitoring and inspection becomes necessary, especially during high flows. One of the methods for carrying this out is to use portable instruments that need to be deployed on the water surface, and this is conventionally carried out on manned boats. However, this practice can be quite hazardous, especially during flood conditions in which, additionally, there may not be enough clearance under the bridge for manned boats to pass and no nearby launch facilities either, hence the suggestion of using remote controlled USVs as a viable alternative. Several small vessels have been adapted and tested using recreational remote-control radios and accessories, and have been used successfully during major floods, allowing data to be collected more efficiently and in more detail than was previously possible (Lagasse et al, 2009; Mueller and Landers, 1999).

In September of 2008, the island of Galveston, Texas, was struck by Hurricane Ike, severely damaging the Rollover Pass Bridge on the adjacent Bolivar peninsula. The Centre for Robot-Assisted Search and Rescue (CRASAR) at Texas A&M University deployed the custom-built Sea-RAI USV to inspect the bridge footings for scour and to map the debris around the bridge, operations conventionally carried out with divers who must carry out manual inspections in high currents and low visibility, and which they can only do for small amounts of time at the change of tide. Their experience showed that USVs have sufficient utility for immediate use in littoral inspection, thanks to their good navigability in high currents as well as their ability to carry payloads, such as acoustic cameras, and to transmit data in real-time (Murphy et al, 2009).

CRASAR is devoted to developing rescue robotics for their deployment in disaster situations to save lives. They have taken the concept of launching UAVs from USVs and applied it to rescue missions: in 2005, when Hurricane Wilma struck Florida, they deployed their AEOS USV prototype together with a micro aerial vehicle (MAV) to survey the damage caused by the hurricane. It was the first known demonstration of an USV-MAV cooperation in an actual mission (Murphy et al, 2008). Further to that experience, a project was undertaken in which a marsupial USV-UAV team consisted of the Sea-RAI hosting an UAV. This marsupial team was designed to be the most advanced sea-air pairing, with the USV transporting the UAV into a desired location, and the UAV acting as messenger and provider of an external point of view on the USV (Lindemuth et al, 2011).
The list of potential applications for USVs goes on. Furthermore, the relevance of undertaking research in, and the demand for unmanned vehicles, is accentuated by the founding in 2012 of the Unmanned Vehicle University, Phoenix, US: the first university dedicated to higher education in unmanned air, ground and sea systems, offering several MS and PhD programmes in unmanned systems engineering (Unmanned Vehicle University, 2014). Founder Dr Jerry LeMieux is convinced that the conception of unmanned vehicles as an exclusive prerogative of military institutions is a thing of the past, and hopes that students from all walks of life will acquire the skills and knowledge from his university to apply unmanned technology to all sorts of commercial and business ventures (NewsWatch, 2014).

2.2 Kalman Filtering

2.2.1 A probabilistic approach to data fusion

Long before the invention of the magnetic compass, sailors relied on celestial navigation for determining their location and bearing. At night in particular, they would be guided by the stars (Figure 2.6). Assume such a sailor to be at sea one night. He had been maintaining a constant course during the day, given by a heading angle $h$, guided by the position of The Sun. However, as dusk set in and the last rays of light hid beneath the horizon, his uncertainty of his bearing started increasing as the diverting winds seemed to blow first hither, then dither. At a certain point, the sailor reckoned that he could only be confident of having maintained course to within $\pm 2\sigma_X$ with 68% probability, or to within twice that amount with 95% likelihood.

It would be reasonable to describe the sailor’s belief about the current heading of his ship as a random variable $X$ with a normal probability distribution with mean $h$ and standard deviation $\sigma_X$,

$$X \sim N(h, \sigma_X^2)$$  \hspace{1cm} (2.1)

Let $x$ be the true heading of the ship, which is then a realisation of the random variable $X$. With no other information available, it can be shown (Appendix A) that the best estimate of the ship’s heading is given by the mean of the distribution, $\hat{x} \triangleq E(X) = h$. The qualifier “best” is vague; rigorously speaking, an estimate can be optimal with respect to some criterion. It can be shown that the mean of the distribution, which, being normal, is also its mode, is at the
same time the *most likely* estimate, the *weighted least squares* estimate, and the estimate with *minimum variance*.

![Figure 2.6 Author’s impression of navigating by stars.](image)

The sailor, disconcerted by his uncertainty, sighs and looks up at the heavens. It is then that the glimmer of Polaris, the North Star, catches his eye. In a eureka moment, he quickly recalls his knowledge of astronomy and after some gauging concludes that the heading is most likely to be $z_1$, with some uncertainty given the difficulty of accurately judging the relative position of the star in the sky. The uncertain process of inferring an estimate of the ship’s heading based on observing the position of the North Star may be described by the conditional random variable $Z_1|(X = x)$, read as “$Z_1$ given $x$”, with probability distribution

$$Z_1|(X = x) \sim N(x, \sigma_{z_1}^2) \quad (2.2)$$

where the standard deviation $\sigma_{z_1}$ represents the uncertainty of this inference process. The distribution of the random variable $Z_1|(X = x)$ expresses the probability for the inferred value of the ship’s heading from the observation of the North Star to take on any given value, given that the ship’s heading is actually $x$. The sailor’s inference of $z_1$ may be then thought of as a realisation of $Z_1|(X = x)$. 

21
Though the new uncertainty is less than the original (assume $\sigma_{z_1} < \sigma_X$), the sailor is still frustrated that it is not much reduced, and, being a man of prudence, decides that the most reasonable thing to do is to use both predictions. Thus, he deems that the ship's true course is the average of his prior estimate, $h$, and his new estimate, $z_1$, and navigates the ship accordingly.

The sailor was right to think that by combining both estimates, he was reducing the overall uncertainty. This is in fact the underlying principle of data fusion, Bayesian estimation, and Kalman filtering in particular. However, unbeknownst to him, his arithmetic average is not, in general, an optimal estimate given all the available information.

Given the prior belief (Equation 2.1) and the value derived from observation of the North Star, $z_1$, with known uncertainty (Equation 2.2), it can be shown through the application of Bayes's theorem (Appendix A) that the random variable $X \mid (Z_1 = z_1)$ has a normal distribution

$$X \mid (Z_1 = z_1) \sim N\left(\frac{\sigma^2_{z_1}}{\sigma^2_{z_1} + \sigma^2_X} h + \frac{\sigma^2_X}{\sigma^2_{z_1} + \sigma^2_X} z_1, \frac{1}{\frac{1}{\sigma^2_X} + \frac{1}{\sigma^2_{z_1}}}\right)$$

(2.3)

The probability distribution of $X \mid (Z_1 = z_1)$ can be thought of as a modification of the original or prior belief given by the distribution of $X$, taking into account the occurrence of the observation $z_1$, and is called the posterior belief. The optimal estimate, in all of the senses previously stated, is again given by the mean of the distribution,

$$\hat{x}_1 \triangleq E(X \mid (Z_1 = z_1)) = \frac{\sigma^2_{z_1}}{\sigma^2_{z_1} + \sigma^2_X} h + \frac{\sigma^2_X}{\sigma^2_{z_1} + \sigma^2_X} z_1$$

(2.4)

with the mean square error being equal to its variance,

$$\sigma^2_1 \triangleq \text{var}(X \mid (Z_1 = z_1)) = E[X \mid (Z_1 = z_1)] - E(X \mid (Z_1 = z_1))^2 =$$

$$E[X \mid (Z_1 = z_1) - \hat{x}_1]^2 = \text{MSE}(\hat{x}_1) = \frac{1}{\frac{1}{\sigma^2_X} + \frac{1}{\sigma^2_{z_1}}}$$

(2.5)

Note in particular that $\sigma^2_1 < \sigma^2_X$ and $\sigma^2_1 < \sigma^2_{z_1}$, that is, the variance of the estimate based on the fused information is less than that of either of the individual estimates.
Equation 2.4 is the Bayesian estimate of $x$ based on the prior belief and the observation $z_1$, and Equation 2.5 gives the associated mean square error. The uncertainty can be reduced by taking more measurements. In particular, note that Equations 2.4 and 2.5 can be written recursively,

\[
\hat{x}_1 = \hat{x} + K_1 (z_1 - \hat{x}) \tag{2.6}
\]

\[
\sigma_1^2 = \sigma_X^2 (1 - K_1) \tag{2.7}
\]

where $K_1 \equiv \frac{\sigma_X^2}{\sigma_X^2 + \sigma_{z_1}^2}$.

For example, assume that a parting of clouds reveals Sirius, the brightest star of the night sky. The sailor quickly appraises that based on its perceived position, the ship’s heading is most likely to be $z_2$ with a random error quantified by $\sigma_{z_2}$. Then, applying the recursive equations 2.6 and 2.7, the optimal estimate taking into account the prior belief in addition to both observations is given by

\[
\hat{x}_2 = \hat{x}_1 + K_2 (z_2 - \hat{x}_1) \tag{2.8}
\]

and the mean square error of this estimate is

\[
\sigma_2^2 \equiv \text{var}(X | (Z_1, Z_2) = (z_1, z_2)) =
\]

\[
\sigma_1^2 (1 - K_2) = \frac{1}{\frac{1}{\sigma_1^2} + \frac{1}{\sigma_{z_2}^2}} = \frac{1}{\frac{1}{\sigma_X^2} + \frac{1}{\sigma_{z_1}^2} + \frac{1}{\sigma_{z_2}^2}} \tag{2.9}
\]

where $K_2 \equiv \frac{\sigma_X^2}{\sigma_X^2 + \sigma_{z_2}^2}$, and it is clear that $\sigma_2^2 < \sigma_1^2$ and $\sigma_2^2 < \sigma_{z_2}^2$, that is, the estimate $\hat{x}_2$ is the most confident estimate yet.

In general, if $N$ observations, $z_1, ..., z_N$, are realisations of $N$ normally distributed random variables $Z_1|X = x, Z_2|X = x, ..., Z_N|X = x$, then

\[
X | (Z_1, ..., Z_N) = (z_1, ..., z_N) \sim N(\hat{x}_N, \sigma_N^2) \tag{2.10}
\]

with

\[
\hat{x}_N \equiv E(X | (Z_1, ..., Z_N) = (z_1, ..., z_N)) =
\hat{x}_{N-1} + K_N(z_N - \hat{x}_{N-1}) \tag{2.11}
\]
\[ \sigma_N^2 \equiv \text{var}(X \mid (Z_1, \ldots, Z_N) = (z_1, \ldots, z_N)) = \sigma_{N-1}^2 (1 - K_N) \quad (2.12) \]

and \[ K_N = \frac{\sigma_{N-1}^2}{\sigma_{N-1}^2 + \sigma_{Z_N}^2} \], and \( \sigma_N^2 < \sigma_{N-1}^2 \).

Although the example shown here involved normally distributed probability distributions, the basic principle of incorporating new data into a prior probability belief to obtain a posterior probability using Bayes's formula is applicable in general.

### 2.2.2 The Kalman filter

In the preceding section, the Bayesian approach to data fusion was illustrated, in which the belief (or prior probability distribution) for a random variable and the likelihood of a new observation were fused based on Bayes's theorem to obtain the posterior probability. The example depicted was static, in the sense that the probability distribution did not evolve with time, but only upon incorporating new observations. This section describes fundamentally the same approach for fusing probability distributions, but with the additional consideration that the belief model evolves according to a discrete-time dynamic process, and in which an observation of the output of this process is obtained at each time-step, information which is then fused with the current probabilistic belief model.

Consider the generic discrete-time state-space model of a dynamic system or process shown in Figure 2.7, described by the following pair of equations

\[ x_{k+1} = f(x_k, u_k, \omega_k) \quad (2.13) \]
\[ z_k = h(x_k, \nu_k) \quad (2.14) \]

where the shorthand notation \( x_k \) is used to denote \( x(k) \), and likewise for the other variables. The vector \( x_k \) represents the state of the system at time-step \( k \), \( u_k \) is a controllable (deterministic) system input, \( \omega_k \) is a random input disturbance, and \( z_k \) represents a noisy measurement of the system output, \( \nu_k \) being the (random) measurement noise. The values \( \omega_k \) and \( \nu_k \) are thus realisations of random variables, and the sequences \( \{\omega_k\} \) and \( \{\nu_k\} \) are assumed to be realisations of independent white noise sequences. Since they depend on these random processes, both \( x_k \) and \( z_k \) can also be considered to be realisations
of the conditional random variables $X_k | (X_{k-1} = x_{k-1}; u_{k-1})$ and $Z_k | (X_k = x_k)$ respectively, since they are functions of random variables. It should be noted that the states satisfy the Markov property: given the previous state $x_{k-1}$, the current state is independent of all states prior to $k - 1$. The outputs also satisfy the Markov property with respect to the states: given the occurrence of $x_k$, the output is independent of the states and observations at all other time indices.

![Dynamic system model](image)

Figure 2.7 Dynamic system model, where $\Delta$ is a one-step delay.

Equation 2.13 is an evolution or state-transition model, and Equation 2.14 an observation or measurement model. The states themselves are not directly measureable, whereas the outputs, which are dependent on the states, are.

Let $x_k$ be the true system state at time-step $k$. Assume that the belief at time-step $k$, taking into account all the observed outputs up to and including time-step $k$, is given by

$$X_k | (Z_{1:k} = z_{1:k}) \sim N(\hat{x}_{k|k}, P_{k|k})$$

(2.15)

where the notation $z_{1:k} \equiv \{z_1, \ldots, z_k\}$ is used for convenience. The optimal estimate of $x_k$ based on this belief is, as noted in the previous section, $\hat{x}_{k|k} \equiv E(X_k | (Z_{1:k} = z_{1:k}))$, and the mean square error of the estimate is given by $P_{k|k} \equiv P_k \equiv var(X_k | (Z_{1:k} = z_{1:k}))$. Then the estimate of the state at the next time-step is carried out in two stages:

1. Propagate the belief (Equation 2.15) in time according to the transition model (Equation 2.13). If $f$ is linear, then the evolved belief will remain normally distributed,
\[ X_{k+1} | (Z_{1:k} = z_{1:k}) \sim N(\hat{x}_{k+1|k}, P_{k+1|k}) \]  \hspace{1cm} (2.16)

This is the prior belief at time \( k + 1 \). This step is known as the \textit{prediction} step.

2. Incorporate the observation \( z_{k+1} \) of the random variable \( Z_{k+1} | (X_{k+1} = x_{k+1}) \) into the prior. The posterior belief at time-step \( k + 1 \) can then be calculated using Bayes’s formula, and as was seen in the last section, if both the prior and observation beliefs are normally distributed, then so is the posterior, with lower variance than those of the prior and the observation,

\[ X_{k+1} | (Z_{1:k+1} = z_{1:k+1}) \sim N(\hat{x}_{k+1|k+1}, P_{k+1|k+1}) \]  \hspace{1cm} (2.17)

This step is known as the \textit{correction} step, with the optimal estimate being given by \( \hat{x}_{k+1|k+1} \equiv E[X_{k+1} | (Z_{1:k+1} = z_{1:k+1})] \) and its mean square error by \( P_{k+1|k+1} \equiv \text{var}[X_{k+1} | (Z_{1:k+1} = z_{1:k+1})] \).

The KF estimation process is the application of the heretofore described process for the linear and Gaussian case. Concretely, if the system given by Equations 2.13 and 2.14 can be described as

\[ x(k + 1) = A x(k) + B u(k) + \omega(k) \]  \hspace{1cm} (2.18)

\[ z(k) = C x(k) + \nu(k) \]  \hspace{1cm} (2.19)

and the uncertain variables \( \omega(k) \) and \( \nu(k) \) are realisations of independent normally distributed random variables with zero mean and variances \( Q \) and \( R \) respectively, then the application of the two stage estimation process of the state vector is described by the KF equations 2.22 to 2.26 (Maybeck, 1979), in which the following notation is used:

\[ \hat{x}(k|k-1) \equiv E(X_k | (Z_{1:k-1} = z_{1:k-1})); \quad P(k|k-1) \equiv \text{var}(X_k | (Z_{1:k-1} = z_{1:k-1})) \]  \hspace{1cm} (2.20)

\[ \hat{x}(k) \equiv \hat{x}(k) \equiv E(X_k | (Z_{1:k} = z_{1:k})); \quad P(k) \equiv P(k) \equiv \text{var}(X_k | (Z_{1:k} = z_{1:k})) \]  \hspace{1cm} (2.21)
KF equations

Prediction:

\[ \hat{x}(k|k-1) = A \hat{x}(k-1|k-1) + B u(k-1) \]  

(2.22)

\[ P(k|k-1) = A P(k-1|k-1) A^T + Q \]  

(2.23)

Kalman gain:

\[ K(k) = P(k|k-1) C^T \{ C P(k|k-1) C^T + R \}^{-1} \]  

(2.24)

Correction:

\[ \hat{x}(k|k) = \hat{x}(k|k-1) + K(k) \{ z(k) - C \hat{x}(k|k-1) \} \]  

(2.25)

\[ P(k|k) = \{ I - K(k) C \} P(k|k-1) \]  

(2.26)

Hence the KF propagates the probabilistic description of the state vector, alternately through time (where it is diluted by the added uncertainty of the evolution model) and by fusion with the likelihood of the current observation (which narrows the variance) (Figure 2.8). An initial probability assumption (mean and variance) is thus required to initiate this recursive process. The KF estimate, \( \hat{x}(k) \), is defined as the mean of the posterior at each time step (Equation 2.25), and is the minimum mean square error (MMSE) estimate for said probability function (belief model). The MSE of the estimate is denoted

Figure 2.8 Depiction of the recursive KF process
P(k) and is equal to the variance of the posterior, since the estimate is unbiased
\[ E[(X_k | (Z_{1:k} = z_{1:k}) - \hat{x}(k)] = 0. \]

2.2.3 Kalman filtering in vehicle navigation

In autonomous navigation, the state of the system incorporates information about the location and attitude of the vehicle with respect to some established reference frame, knowledge of which is required by the control system, or autopilot, to determine the actuator adjustments, e.g. in terms of motor torque or rudder angle, necessary to maintain the vehicle on the desired course. Navigational sensors for surface vessels typically include compasses, inertial sensors, GPS receivers, and speed logs.

Components of the system state may or may not be measurable, but they are predictable with a model that describes the dynamical behaviour of the system. Since both measurements and model-based predictions always contain a certain degree of uncertainty, for example, in the form of sensor noise or unmeasurable system input disturbances, respectively, the different sources of information are typically fused using a KF in order to reduce the estimation uncertainty of the system state.

Rudolf E. Kalman originally developed the filter in 1960 (Kalman, 1960) for trajectory estimation for the Apollo circumlunar programme. The recursive implementation of the KF (Equations (2.20) to (2.24) makes it well suited to computer implementation and is one of the reasons it has been so successful since its invention. Although initially used in spacecraft navigation (Smith et al, 1961; Schmidt, 1981), it has since been applied in numerous other fields. Examples of KF applications are: forecasting economic time series (Morrison and Pike, 1977) and constructing financial econometric models (Bouye, 2009), computer vision applications such as pattern recognition (Ondel et al, 2007), tracking objects such as ballistic missiles (Siouris et al, 1997), industrial electronics applications such as parameter estimation of electric machines (Auger et al, 2013), filtering of measurements in manufacturing processes (Oakes et al, 2009; Sorenson, 1985), signal processing for biomedical applications (Oikonomou et al, 2009) and monitoring systems based on change detection algorithms (Severo and Gama, 2006), and soft computing systems, e.g. fuzzy membership function estimation (Ghanai and Chafaa, 2009) and neural network design (Haykin, 2001). Also, knowledge of the state-vector is an important aspect of system control (Crain, 2002), and the KF is a key part of the optimal linear-quadratic-Gaussian (LQG) control problem (Makila, 2004).
As detailed in the previous sections, the KF algorithm’s inherent structure allows it to naturally combine measurements from various sensors, taking into account the accuracy of each one. The data-fusion from various sensors provides a more reliable estimate than can be obtained from each individual sensor alone, prompting the use of the KF as a tool for combining low-cost sensors to synergistically create highly-reliable estimates that would otherwise require more precise sensors. Particularly for UGVs, and since the availability of GPS and low-cost and low-power solid-state inertial navigation systems (INS), the KF has been used to perform INS-GPS integration (e.g., Wolf et al. 1997; Tan et al. 2007).

The advantage of the INS-GPS symbiosis is explained as follows. On the one hand, standard GPS receivers are unable to provide the rate or precision required when used on a small vessel such as an USV. On the other, implementation of inertial measurement units (IMUs) in surface/ground systems is more difficult than in airborne systems due to the high noise to signal ratio introduced by interaction of the vehicle with the surface and its relatively slow and vibration-clad movement (Lamon, 2008). The main limitation of low-cost INS systems, in that they can only provide accurate estimates of position and attitude for a short time span before integration drift becomes significant (Section 4.3.3), can be overcome by incorporating a GPS fix at a lower GPS sampling rate; thus, this strategy combines the short-term accuracy of INS with the long-term stability of GPS, providing a relatively low-cost and reliable solution for surface navigation. Integrated GPS-INS sensor packages with onboard microcontrollers that implement KF fusion are now available commercially (VectorNav Technologies, 2014).

Although they have been used extensively in UGVs, INS-GPS navigation systems are being successfully applied in ASVs as well. An example can be found in the USV being developed at Virginia Tech University (VaCAS, 2011), which uses differential GPS (DGPS) together with a puck-sized micro electro-mechanical system (MEMS) based technology inertial sensor offered by MicroStrain Inc that is popularly used in mobile robotic applications (MicroStrain 2012). Technological advances of INS-GPS systems are reviewed in Schmidt (2003), while a comparison of low-cost IMUs for autonomous navigation can be found in Chao et al. (2010). A review of MEMS systems may be perused in Barbour et al. (2010), and a comparison of several MEMS-based IMUs is presented in De Agostino (2010).

The KF is also used to combine GPS and IMU measurements with magnetic compass sensor readings. For example, Zhang et al. (2005) describe the use of an
unscented KF (UKF), a non-linear version of the KF, to combine low-cost IMU, GPS and digital compass using a sophisticated dynamical model of the vehicle. Others have successfully implemented KF-based ASV navigation without IMUs altogether, for example, the ASV Charlie, equipped solely with GPS and magnetic compass units, which uses an extended KF (EKF), a KF for non-linear systems which linearises the model around the estimated state (Caccia et al, 2008). The reader wishing to know more about the workings of the UKF and EKF is referred to Aich & Madhumita (2010).

2.3 Robust Kalman Filtering

The basic KF scheme yields an optimal estimate only for linear processes with stochastic process input disturbance and measurement noise sequences that are white and Gaussian, and when the process dynamics are known precisely, along with knowledge of the initial state estimate and estimate error covariance. In practice, process dynamics models are always an approximation of the true dynamics, initial estimates might be incorrect, and the assumed process disturbance and measurement noise covariances inaccurate, affecting the effectiveness of the KF. There is no general theory that guarantees a statistically optimal estimate when knowledge of system dynamics and noise statistics are incomplete. In these situations, it is usual in practice to assume some completely specified linear model for the system as well as process and measurement noise covariance matrices. However, this can incur in divergence of the predicted mean square state estimation error (which typically remains bounded) from the actual mean square error (which may actually diverge) (Price, 1968). The degradation of the KF estimates under incorrect modelling is illustrated in Chapter 5.

Not only are accurate noise statistics often crucial, but moreover, the sensor accuracy may be changing in time - for example, GPS accuracy is affected by the positions of the satellites, interference of the radio signal, physical barriers to the signal like mountains or those due to atmospheric conditions. In this scenario, the measurement noise covariance, $R$, must be continuously adapted so as to accommodate for changes in GPS accuracy in order to achieve good performance from the KF. Likewise, in USV navigation, the sea conditions are continuously varying, and should ideally be reflected in a varying process input disturbance covariance, $Q$, that most accurately reflects the conditions at each moment.
A certain degree of robustness of the filter can thus be achieved by implementing adaptive capabilities. Strategies for adapting measurement and process noise covariance matrices using innovation-based estimation have been applied for integrating IMU measurements with GPS data (Mohamed and Schwarz, 1999; Loebis et al, 2004a). Methods based on covariance scaling and multiple model adaptive estimation have also been explored (Hide et al, 2003).

To provide adaptive capabilities, KFs are often used in combination with artificial intelligence (AI) techniques, in particular those based on fuzzy logic. For example, fuzzy logic has been used to discriminate, through selective weighting, KFs working in parallel (Hsiao, 1999), and in decentralised, cascaded, and federated architectures (Lendek et al, 2008; Escamilla-Ambrosio and Mort, 2002; Xu, 2007). Fuzzy logic has also been used to tune the parameters of a KF. For example, through analysis of the innovations sequence (Chapter 5), the divergence of the filter can be monitored and corrective action taken through the tuning of the process disturbance and measurement noise covariances using fuzzy rules. Subramanian et al (2009) applied this idea to improve the performance of a KF used to fuse information from machine vision, laser radar, IMU and speed sensors. Similar approaches have been used in INS-GPS navigation systems; for instance, KF adaptation via fuzzy rules based on covariance matching of the actual and theoretical measurement noise covariance (Xu et al, 2006; Loebis et al, 2004b). Other fuzzy logic tuning criteria have been implemented as well to adapt the KF according to the measurement or information available (Kobayashi et al, 1998).

However, one of the problems with using fuzzy logic techniques has been on how to determine adequate membership functions of the fuzzy sets. The a priori approach is a heuristic one, based on experience and observation, but is rarely optimal. To this end, genetic algorithms have been used to optimise the fuzzy membership functions, as Loebis et al (2004a) have done in an autonomous underwater vehicle (AUV) navigation application. Also, neuro-fuzzy techniques have been used extensively in mobile robotic applications (Godjevac and Steele, 1999), and for navigation systems in particular: for instance, an adaptive neuro-fuzzy inference system has been developed for tuning the fuzzy membership functions which endow a INS-GPS navigation system with learning capabilities (Tiano et al, 2001).

As mentioned, the KF's optimal performance is conditioned not only upon accurate knowledge of stochastic model statistics (Q and R), but also upon accurate modelling of the system dynamics. When there is insufficient
knowledge of process and measurement noise statistics, the aforementioned techniques based on AI enhancement effectively try to infer such information from the filter's performance, often through learning mechanisms. However, it is to be noted that in the face of deficient system modelling, these strategies basically search for an adequate disturbance and noise statistics model not just to reflect the true input disturbance and measurement noise, but also to compensate for the incorrectly modelled parameters of the process dynamics. This is clearly a compromise, where all uncertainty is lumped into stochastic variables, so that they neither reflect the true stochastic processes that affect the system, nor the modelling uncertainty of the otherwise deterministic process dynamics, but somehow manage to compensate the lack of modelling of one by incorporating it into the other. While this approach has been used somewhat successfully, it does so at the cost of using highly sophisticated AI methods that must arbitrate this procedure otherwise guided only by heuristics, and it raises the question of finding alternative means to describe the uncertainty in the modelling of the system dynamics.

One of the methods suggested in the literature is to use the KF itself to perform on-line system identification when $Q$ and $R$ are known but one or more of the dynamic process model parameters is unknown or changing. The unknown parameters are modelled as having normal probability density functions, and the process equations are then rearranged to include these parameters as states of the system, whence the KF can be applied to estimate this augmented state vector (which contains the uncertain process parameters) (Chui and Chen, 2008). The problem of course is that every new unknown parameter makes the system more underdetermined.

The aforementioned techniques can be thought of as artificial props or aids devised to make useable a filter that is in itself inherently not robust to model uncertainty. However, in order to surmount the practical limitation of accurately modelled dynamics imposed by the KF hypotheses, research into filters which are themselves intrinsically robust to uncertain system dynamics from a mathematical perspective has become a wide field of study. These so-called robust filters aim to obtain estimates of unmeasurable state variables when the system model is only known with some degree of precision.

Most of the research in the field of robust estimators for uncertain systems has centred on three approaches: the guaranteed cost state paradigm, the set-valued estimation approach, and robust $H_\infty$ filtering. The first of these is sometimes regarded as an extension of the KF to uncertain systems, and called robust
Kalman filtering. The approach, developed in principle for quadratically stable time-invariant systems under steady-state operation, shows that the solution of certain algebraic Riccati equations can provide a cost matrix that defines an upper bound on the state estimation error covariance matrix (Jain, 1975; Petersen and McFarlane, 1996; Petersen and Savkin, 1999). This then evolves into the problem of minimising said bound.

The set-valued estimation approach is based on a deterministic interpretation of Kalman filtering obtained by describing the noise processes as norm bounded (Bertsekas and Rhodes, 1971), and then finding the set in the state space that is consistent with the observed measurements via set inversion, usually requiring some optimisation algorithm (Jaulin, 2009; Savkin and Petersen, 1998; Zhu, 2012).

$H_\infty$ filtering centres on minimising the $H_\infty$ norm of the transfer function from the disturbance inputs to the filter estimation error. Unlike the KF, the disturbance and noise processes need not be Gaussian, although the optimisation problem is usually dependent on assuming some kind of structured uncertainty inherent in these processes. In the case of uncertainties in the system model, the robust version of $H_\infty$ filtering further seeks to minimise the worst case $H_\infty$ norm of the transfer function between the noise inputs and the estimation error (Gao and Chen 2007; Sayed 2001).

### 2.3.1 Interval Kalman filtering

All the approaches to robust estimation hitherto described involve either a modification or extension of the KF algorithm, or formulate the problem based on different objectives. However, another approach to filtering for systems with bounded model uncertainty is to directly use the KF on these uncertain models. The method proposed by Chen et al (1997), the so called IKF, is actually not a modification of the KF at all: as will be shown in Chapter 5, its equations exactly mirror those of the traditional KF. In order for the filter to have the same equations, the model it operates on must have exactly the same form as required for a KF. The difference, in this case, is in the type of element set it is constructed upon: rather than the set of real numbers $\mathbb{R}$, the IKF assumes elements to belong to the set of nonempty, closed and bounded real intervals, $I\mathbb{R}$. This allows for a natural description of bounded uncertainty to be incorporated into any model without necessitating any additional morphological description. The IKF retains all the same optimal properties of the standard
KF, naturally providing guaranteed bounds to the optimal state estimate by giving these as interval-valued elements.

Despite the seeming simplicity and optimality of this approach, the IKF has seen very limited acceptance since it was first proposed, with only a few authors suggesting its use for practical applications (He and Vik, 1999; Siouris et al, 1997; Tiano et al, 2001; Tiano et al, 2005), but with no evidence of it being implemented in practice. One of the main reasons for this limited use is that the IKF requires operating with interval arithmetic (IA), which can be difficult to implement successfully in practice owing to its overly conservative nature. As will be illustrated in Chapter 5, this results in the actual set of states of the interval system being excessively over-estimated, and are of little use in practice. Therefore one of the objectives of this research is to develop a method that enables efficient computation of the IKF states so that these may be used as was intended by Chen and co-authors.

2.4 Summary

A review of the current state of affairs in USV development has shown the potential applications for such vehicles in all walks of life, and as these grow, so does the need for effective and robust autonomous navigation systems. An intuitive depiction of the data fusion process has provided the background and insight to the principles of Kalman filtering, and the ensuing review has also established it as a widely used algorithm for state-estimation by data fusion, particularly for vehicle navigation.

An overview of robust filtering techniques springing from the shortcomings of the KF has also established that there is a wide research base in this area, and that the IKF as a proposed robust filter has not yet been exploited in any practical way. Understanding the reasons behind this and consequently proposed solutions will be the main topic of discussion of the rest of this thesis.
Chapter 3

The Springer Unmanned Surface Vehicle

“The chance for mistakes is about equal to the number of crew squared.” — Ted Turner

In 2004, the Marine and Industrial Dynamic Analysis (MIDAS) research group at Plymouth University began designing and developing a USV named Springer. Conceived as a cost-effective, environmentally friendly USV for undertaking environmental surveys in shallow waters such as rivers and inland waterways, it serves too as a test-bed for developing intelligent NGC systems. The work described in this thesis is part of that tenet, and this chapter is devoted to describing the current state of hardware and software architecture employed to carry out the same.

3.1 Vehicle Characteristics and Hardware Overview

Springer (Figure 3.1) is a medium waterplane twin hull vessel measuring 4.2 m long and 2.3 m wide, with a displacement of 0.6 tonnes. The catamaran-type configuration provides sufficient geometric stability to be able ignore roll and pitch effects under most conditions, hence for all practical purposes the vehicle can be treated as possessing three degrees of freedom.

The vehicle is propelled by two 24 V, 74 lbs electric Minn Kota Riptide® transom mount saltwater trolling motors, powered by eight Sonnenschein SL135
12 V 135 Ah batteries, four placed within each hull as two pairs of parallel-connected batteries wired in series to produce 24 V. The batteries are connected to a RoboteQ AX2850 digital motor controller in the starboard-side hull, and the battery terminals are accessible via cables that can be reached through portholes in the side of the hull for charging. The motor housings contain leak detectors and optical speed encoders, and are connected to the motor controller via custom made circuitry in a separate box next to the controller. The RoboteQ can accept either R/C radio commands or serial data (RS 232) from a PC, and this box also houses circuitry for commuting between radio control and serial control of the motors. Schematics of these circuits can be found in the Springer manual available through the web site (MIDAS, 2014).

![Springer USV](image)

The RoboteQ AX2850 is a configurable, microcomputer-based, dual-channel digital motor controller with built-in high power drivers. An instruction set of character strings is provided to send individual motor speed commands as well as configuration commands via RS-232 from the PC to the controller. Using feedback from the optical encoders enclosed in the motor housings, the RoboteQ regulates the speed of the motors to match the commanded set-point values. This regulation is based on an internal PID controller with configurable parameters. (Roboteq Inc, 2007).

Navigational sensors and computers are housed in watertight Pelican™ Cases on each hull. These are placed in a bay area between two aluminium crossbeams that connect the two hulls. The computer on the starboard-side Peli-Case links directly to the motor controller in the hull through a serial-data cable which accesses both the watertight case and the interior of the hull via waterproof Bulgin connectors. Panels on both hulls host several other connectors, for
example, to connect cables between the two hulls for transferring leak detector and encoder signals from the port-side hull to the motor-controller box in the starboard-side hull, and power lines from batteries on the port side to motor-controller on the other and then back to the motor on the port side. These cables cross over from one hull to the other through the hollow crossbeams (Figures 3.2a and 3.2b).

(a) Cross-section of the starboard-side hull of Springer. The port side is similar but without the motor controller and electronics box. The mast is shown on the same section for convenience.

(b) Top view of the Springer.

Figure 3.2. (a) Cross-section of the starboard-side hull of Springer; (b) top view.
The computers and sensors within each Peli-Case receive power from two additional smaller batteries (Sonnenschein GF 12 022 Y F 12V 24AH gel type) contained in each hull in the space directly beneath these, supplying 12V independently to each of the Peli-Cases via cables again wired through connectors on the hull panel. One of these batteries also supplies power to the router on the mast, which will be described next.

The mast is a vertical shaft with a polycarbonate watertight enclosure affixed to the top that houses a Linksys WRT54GX wireless router. The router is connected via Ethernet cables to the computers on both Peli-Cases, allowing these to transfer data between them through a LAN. The router also serves to communicate wirelessly with the on-board computers from external devices such as laptops. Its antennae have been fixated to the lid of the enclosure and wired to the router within through small holes practiced on the lid. The mast also carries a GlobalSat BU-353 waterproof USB GPS receiver unit that can be connected to the computer in either Peli-Case.

As mentioned, a computer is housed within each Peli-Case. The computers need to run all the NGC algorithms, receiving and processing data from sensors and sending control signals to the motor-controller. The Intense PC pro, a general purpose PC running Windows 7 on an Intel Core i7-3517UE 1.7GHz 64 bit dual core processor, was chosen because of its low-power consumption (10 W-26 W), small size and ruggedised die-cast aluminium shell, fanless operation and extended environment temperature range tolerance (0 to 70 C with SSD). An RS-232 serial port comes as standard with these units, required for interfacing with the RoboteQ motor controller. An optional connectivity module with up to six additional serial ports is available, and was chosen for the unit installed in the starboard-side Peli-Case to connect additional serial devices (Section 3.1.1). Moreover, these units are in the price range of ordinary desktop PCs with similar computing specifications.¹

With no airflow within the Peli-Case, heat dissipation from the interior to the exterior relies on conduction through heat-sinks installed at its base (Figure 3.3a). Hence a fanless PC with no moving parts is advantageous to keep heat generation and power consumption to a minimum. Nevertheless, in order to ensure that the PC would not overheat when operating within the sealed Peli-Case, several experiments were carried out beforehand, in which heat was generated in a sealed Peli-Case and the temperature within monitored over time. Results showed that even with 80 W of heat generated within the case,

¹ accurate as of date of purchase, October 2012.
the temperature remained sufficiently low for the PCs to be able to operate (Motwani 2012b).

It should also be noted that a general purpose PC was chosen in lieu of an embedded system since the research conducted for this thesis pursues the proof of concept of the ideas and methods developed, whereas the computational efficiency of the particular implementation adopted is secondary. To this end, the algorithms that this research has produced were implemented in MATLAB® for its high-level functionality which makes it well suited for rapid prototyping. A detailed description of the software architecture is deferred to Section 3.2.

### 3.1.1 The starboard-side Peli-case

This section details the hardware contained within the Peli-Case situated on the starboard side of the vehicle, configured to the requirements of the research work described in this thesis. The Peli-Case on the port side is currently set up for other research objectives and is the subject of another thesis.

Apart from the PC, the starboard Peli-Case accommodates three digital compass units and an Arduino-Mega board used for interfacing a MEMS-based gyroscope. The layout and connections with the exterior are shown in Figure 3.3. The connectors numbered in Figure 3.3a and Figure 3.3c are summarised in Table 3.1, whilst the components within the case numbered in Figure 3.3b and Figure 3.3c are labelled in Table 3.2.

The three compasses consist of a TCM2, an HMR 3000 and a KVH C100. All of these are microprocessor controlled units which can interface to a PC via RS-232 serial protocol. The main characteristics of each are described next.

The TCM2 by PNI Corporation (Figure 3.4a) incorporates a three-axis magnetometer and a two-axis tilt sensor. The magnetometers are magneto-inductive: each axis has a sensing coil or solenoid which serves as an inductive element in a low-power LR oscillator. The relative permeability of the coil core material varies as a function of the external magnetic field, which is detected through the circuit’s response. The tilt sensor allows the microprocessor to mathematically correct the compass reading for tilt (known as electronic gimballing) (PNI, Co. 2014).
Figure 3.3 Starboard-side Peli-Case. (a) detailed view of the connectors on the rear side of the case; (b) layout of components within the Peli-Case; (c) schematic diagram of the layout within the Peli-Case and connections to external devices.
Table 3.1. Peli-Case rear side connectors

<table>
<thead>
<tr>
<th>Connector</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2 way, for connection to 12V battery in hull</td>
</tr>
<tr>
<td>2</td>
<td>4 way for USB link between PC and GPS receiver</td>
</tr>
<tr>
<td>3</td>
<td>same as 2 (for connection additional GPS receiver)</td>
</tr>
<tr>
<td>4</td>
<td>3 way for serial link between PC and motor-controller</td>
</tr>
<tr>
<td>5</td>
<td>9 way for Ethernet link between PC and router</td>
</tr>
<tr>
<td>6</td>
<td>2 way for supplying power to router</td>
</tr>
</tbody>
</table>

Table 3.2. Components within the Peli-Case

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Intense PC pro</td>
</tr>
<tr>
<td>2</td>
<td>TCM2</td>
</tr>
<tr>
<td>3</td>
<td>HMR 3000</td>
</tr>
<tr>
<td>4</td>
<td>KVH C100</td>
</tr>
<tr>
<td>5</td>
<td>gyroscope and Arduino</td>
</tr>
<tr>
<td>6</td>
<td>terminal block connector</td>
</tr>
</tbody>
</table>

The HMR 3000 by Honeywell (Figure 3.4b) is based on anisotropic magnetoresistance (ARM). Magnetoresistance is the property of a material to change the value of its electrical resistance when an external magnetic field is applied to it. Anisotropic means that the property is additionally directionally dependent, that is, the electrical resistance depends on the angle between the direction of electric current and orientation of the magnetic field. Again, it is electronically gimballed, giving accurate heading even when the compass is tilted up to 40 degrees (Honeywell International, Inc., 2014).

The C100 by KVH Industries (Figure 3.4c) is a fluxgate compass, consisting of a high permeability saturable ring core suspended in an inert fluid that maintains it horizontal. Windings are used to drive the core in and out of saturation, and two secondary perpendicular coils are used to sense the horizontal components of the magnetic field induced in the core by the driving coils. When there is no external magnetic field, the net magnetic field generated by the driving coils is zero, but in the presence of an external magnetic field, a temporary net field is generated as one part of the core comes into and out of saturation before the other, and this is picked up by the secondary coils, in which a voltage is induced due to the changing field. The non-mechanically-gimballed C100 provides...
accurate heading with the unit tilted up to 16 degrees (KVH Industries, Inc. 2014).

![TCM2](image1)

![HMR 3000](image2)

![KVH C100](image3)

Figure 3.4 Compass units: (a) TCM 2; (b) HMR 3000; (c) KVH C100

All of the compasses are provided with specific string-based instruction sets that allow the user to run in-built calibration procedures and query the instruments for heading data.

The gyroscope (Figure 3.5b) is the TinkerKit Gyroscope 2 Axis sensitivity 4X, a breakout board based on the LPR5150AL from ST Microelectronics, a MEMS dual axis analogue gyroscope popular with robotics hobbyists. The TinkerKit gyroscope outputs 0V to 5V on one of its two signal pins when its angle is changed, with a value of approximately 2.5 V when there is no angle change in either axis (TinkerKit, 2014). In order to digitise this value so that it can be sent to the PC, an Arduino Mega microcontroller board was used.

The Arduino Mega 2560 (Figure 3.5a) is a microcontroller board based on the Atmega2560. It has 54 digital input/output pins (14 of them can be used as PWM outputs), 16 analogue inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a
reset button. The board connects to a PC using serial communication via the USB connection, and data sent to the PC can be read by any serial terminal program such as HyperTerminal, PuTTY, Tera Term, etc. (Arduino, 2014).

The Arduino incorporates a 10 bit ADC (analogue-to-digital converter), converting the input voltage range, 0 to 5 V, to a digital value between 0 and 1023 ($2^{10}-1$). The turning rate in degrees per second (dps) is obtained by dividing the difference between the value read and the reference value (when the gyroscope is still) by the sensitivity of the device (found on the datasheet). Figure 3.6 is a schematic drawing of how the devices are connected.
A summary of the principal characteristics of the compasses and the gyroscope is given in Table 3.3.

Table 3.3. Springer sensor suite for heading determination.

<table>
<thead>
<tr>
<th>Unit</th>
<th>Technology</th>
<th>Measurement noise*</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCM2 compass</td>
<td>Magneto-inductive effect 3-axis magnetometer with two-axis inclinometer for tilt compensation (PNI, Co. 2014)</td>
<td>1° RMS</td>
</tr>
<tr>
<td>HMR3000 compass</td>
<td>Magneto-resistive sensors in three orthogonal directions, with fluidic tilt sensor for tilt-compensated heading (Honeywell International, Inc. 2014)</td>
<td>0.5° RMS</td>
</tr>
<tr>
<td>KVH C100 compass</td>
<td>Flux-gate compass with iron magnetic compensation (KVH Industries, Inc. 2014)</td>
<td>0.5° RMS</td>
</tr>
<tr>
<td>TinkerKit gyroscope</td>
<td>2-axis MEMS technology gyro, based on the LPR5150AL from ST Microelectronics (Tinkerkit, 2014)</td>
<td>0.05(deg/s) RMS</td>
</tr>
</tbody>
</table>

*approximate values obtained from testing the devices in the trials environment. The gyro RMS noise is the result of smoothing the acquired signal over 1 second (overall system sampling time), as the gyro data was sampled at a faster rate.
3.2 User Interface and Algorithm Execution

The Microsoft® Visual Studio IDE was used to build an application with a customised user control panel programmed using C#. The control panel is used to configure mission parameters (e.g., specify way points), select and establish serial port communication with the required sensors and with the motor-controller, select the desired NGC system, and start running the mission. During the mission, a routine programmed within this application takes charge of executing the mission by running in a continuous loop.

The NGC algorithms however are coded in MATLAB. In particular, all the filtering algorithms presented in this thesis were coded and simulated in MATLAB because of its ease for rapid prototyping and its computational efficiency for carrying out matrix operations. Generally, the algorithms were coded either as static functions or as class-based methods. This means that the end-user can run these algorithms by simply calling the appropriate methods, without needing to expose the underlying code.

In order to run the necessary NGC algorithms in MATLAB from the C#-controlled operation of the USV, MATLAB is invoked as an Automation server from the C# environment. Automation is a COM protocol that allows one application (the controller or client, in this case the C# program) to control objects exported by another application (the server, in this case MATLAB). MATLAB’s Automation server implements a number of properties and methods that the client can use to pass data to and read data from it, as well as execute MATLAB statements within it. In this way, the NGC algorithms coded in MATLAB can be triggered from the C# program at any point in its execution and the relevant data exchanged.

The general ideal of this architecture by which the C# application interfaces with the user and hardware, invoking MATLAB for executing the relatively computationally intensive NGC algorithms, is depicted in Figure 3.7.

A more detailed description of the software execution process is shown in Figure 3.8. The user first launches the C# application (user control panel) on the PC connected to the motor-controller and sensors (the Intense PC within the starboard-side Peli-Case. This PC can be accessed from any Wi-Fi enabled laptop through a pre-configured LAN via the router on the mast, using a remote desktop software such as Windows Remote Desktop Connection). Automatically during this process, the application also launches a MATLAB Automation server.

45
The user can then select the serial ports to connect each device to, configure mission parameters, and select particular NGC algorithms to be used for the mission. Once it is verified that communication is established with the selected devices and the mission is configured as desired, a button on the control panel starts execution of the mission, and if an autopilot has been selected\(^2\), motor commands are sent to the motor-controller and the vehicle is set in motion.

The main loop depicted in Figure 3.8, which is executed at a frequency of 1Hz, represents the stages within the autonomous execution of the mission. Firstly, data from the relevant sensors is read by the C# program and displayed on the user control panel. This data is also sent to the MATLAB Automation server workspace. According to the selected NGC system, the C# program requests the MATLAB server to execute the necessary NGC algorithms to calculate the estimated navigational heading of the USV, the desired reference heading, and the autopilot steering action. These values are passed back to the C# program which displays them on the user control panel, and sends the necessary commands to the motor-controller to regulate the speed of each motor according to prescribed autopilot output (the NGC algorithms are explained in more detail in Chapter 4). This process is then repeated until the mission is either completed or is aborted by the user. All data acquired or generated at each loop iteration (sensor readings, reference heading, autopilot command) is automatically saved to a file upon end of mission.

\(^2\) Navigation-only missions are possible as well, in which case the vehicle may be manually piloted.
Figure 3.8 Software execution.

### 3.3 Summary

This chapter has provided an overview of the *Springer* USV’s characteristics and its hardware, as well as the software architecture used. The current configuration of the *Springer*’s hardware is the result of trying to utilise existing components from previous research projects whilst updating others according to the specific needs of the current research. Of particular importance to the work described in this thesis are the navigational sensors within the starboard-side Peli-Case, whose main characteristics have been described in detail.
The use of a general purpose PC platform for implementing the NGC algorithms makes this a highly flexible and reconfigurable system, e.g. allowing it to be easily operated using alternative code written in other programming languages, for connecting supplementary sensors, or for running additional data-logging software e.g. to deploy scientific instrumentation, etc. Also, all the algorithms developed in MATLAB can easily be run and tested via customised simulations on any PC, which helps speed up development and reduces trial costs.

Having described the underlying hardware and operational architecture of the vehicle, the next chapter will give a brief description of the actual NGC algorithms which are used to calculate the variables needed to operate the vehicle autonomously.
Chapter 4
Navigation, Guidance and Control

“Nothing comes sailing by itself.” — Alexander Dale Oen

A general overview of NGC was presented in Chapter 1. This chapter details the NGC strategies used for autonomous way point tracking of the Springer USV in this study. A particular waypoint mission is used as an example throughout the thesis, and is presented in Section 4.1. Before describing the NGC algorithms, the vehicle steering dynamics and modelling thereof are discussed in Section 4.2. NGC techniques are presented in Sections 4.3 to 4.5, and finally some mission simulations are shown in Section 4.6.

4.1 Waypoint Tracking

Historically, waypoints have been associated with physical landmarks, geographical formations that stand out from their near environment such as rock formations and mountains, as well as easily distinguishable man-made structures, which allow travellers to determine their absolute location on a map. Waypoints for marine navigation include landmarks along the coastline such as lighthouses, chimneys, churches, etc. Lightships anchored at sea, light-towers, buoys and lanbys are common seamarks used to mark offshore positions important to the mariner.

With the advent of GPS systems, waypoints increasingly consist of abstract locations on a map used to define invisible routing paths for navigation. They are specified as coordinates on a grid, such as degrees of latitude and longitude.
The waypoint mission described herein is specified on a two-dimensional rectangular grid whose axes measure distances rather than degrees, with the abscissa pointing due true east and the ordinate due true north. The origin of the grid is placed at a known reference latitude and longitude, and motion is considered to be flat and contained within the plane tangent to the Earth’s surface at said reference point. Distances between points calculated using this planar approximation for the surface of the Earth will be valid as long as the distances travelled by the vehicle are small and the location is not too near the poles. For latitudes circa 50°, computing the distance between two points separated 20 km using the Pythagorean Theorem incurs in an error of about 20 m, that is, about 0.1% (Chamberlain, 1996).

Consider the mission plan shown in Figure 4.1, represented on a 2D Cartesian grid, consisting of a series of predefined waypoints specified by the coordinates given in Table 4.1. These coordinates are, as discussed earlier, relative to the origin of the grid: the absolute geographic coordinates are easily determined given the geographical location of the origin (e.g. latitude and longitude). The conversion of relative coordinates to geographic latitude and longitude will be detailed in Chapter 9. Angles (such as vehicle heading, etc) are considered with respect to the x axis and are positive in an anti-clockwise direction.

The vehicle must subsequenly target each of these waypoints in order. The ideal path for the vehicle to follow consists of the straight line segments connecting each pair of consecutive waypoints, as shown in Figure 4.1.

Figure 4.1 Waypoint tracking mission
Table 4.1 Waypoint coordinates

<table>
<thead>
<tr>
<th>Waypoint No.</th>
<th>x coordinate (m)</th>
<th>y coordinate (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (start)</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>180</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>335</td>
<td>90</td>
</tr>
<tr>
<td>3</td>
<td>379</td>
<td>264</td>
</tr>
<tr>
<td>4</td>
<td>202</td>
<td>295</td>
</tr>
<tr>
<td>5</td>
<td>46</td>
<td>205</td>
</tr>
<tr>
<td>6</td>
<td>122</td>
<td>42</td>
</tr>
<tr>
<td>7 (finish)</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

In subsequent simulation studies, the effect of a surface current is simulated by adding a translational component to the vehicle's movement in the direction of the current.

4.2 Vehicle Motion Model

A vehicle motion model allows predicting the vehicle's future position and heading angle based on its current position and heading, and the current rpm of each motor. As described in Chapter 3, the vehicle is propelled by two individually controlled motors, one on each hull. The turning rates of each motor determine both the forward speed of the vehicle as well as changes in its heading, as depicted in Figure 4.2, where $n_1$ and $n_2$ are the two propeller speeds in rpm.

![Figure 4.2 Block diagram representation of a two-input, two-output, USV motion model.](image)

In order to decouple forward motion from steering dynamics, it is assumed that as long as the average rpm of the two motors remains constant, the vehicle maintains its velocity regardless of turning manoeuvres. Let $n_c$ and $n_d$ represent the common mode and differential mode propeller turning rates defined as
The previous assumption states that if $n_c$ is maintained constant, then the vehicle will move at a constant velocity. Steering is achieved by applying different revolution rates to the two motors, i.e., through the control of $n_d$. For a constant value of $n_c$, that is, with the vehicle moving at constant speed, the steering dynamics can thus be modelled as a relationship between $n_d$ and the rate of change of heading of the vehicle.

Such single-input, single-output (SISO) models have been derived for the Springer USV using system identification (SI) techniques. Trials have been conducted during which $n_c$ was maintained constant and various turning manoeuvres were carried out by applying step-changes in $n_d$. The applied $n_d$ and vehicle heading angle were recorded at each sampling instant (1 Hz). SI techniques (Ljung, 1999) were applied to correlate the collated data and obtain linear steering models of the form

\begin{align}
    x(k+1) &= A \ x(k) + B \ u(k) \quad (4.2) \\
    y(k) &= C \ x(k) \quad (4.3)
\end{align}

where the input $u(k)$ is the differential propeller speed $n_d(k)$, and the output $y(k)$ the rate change in heading angle $\dot{\theta}(k)$. Steering models of this sort have been obtained for various different values of $n_c$ (Annamalai and Motwani, 2013).

The output of Equation (4.3) can be integrated according to

\begin{align}
    \theta(k+1) &\approx \theta(k) + T_s \ y(k) = \theta(k) + T_s \ C \ x(k) \quad (4.4)
\end{align}

where $T_s = 1 \ s$ is the sampling time. Hence, for the vehicle operating at constant speed $v$ (corresponding to some value of $n_c$), (4.2) and (4.3) allow one to obtain the next heading $\theta(k + 1)$ given the current heading $\theta(k)$ and previous state $x(k - 1)$ and applied differential motor speed $n_d(k - 1)$. 

\[ n_c = \left( \frac{n_1 + n_2}{2} \right) \] 
\[ n_d = \left( \frac{n_1 - n_2}{2} \right) \]
The position of the vehicle can then be updated according to

\[
\begin{aligned}
(x_{USV}(k+1), y_{USV}(k+1)) &= \\
(x_{USV}(k), y_{USV}(k)) + T_s v(\cos(y_k), \sin(y_k)), \quad y_k \triangleq \frac{\pi}{180} \frac{\theta(k+1)+\theta(k)}{2}
\end{aligned}
\]

(4.5)

In the case of a surface current, the position of the vehicle is calculated as in Equation (4.5) but with an added translation in the direction of the current proportional to the current speed.

The linear steering models of the form given by Equations (4.2) and (4.3) obtained for various different vehicle speeds can then be interpolated to obtain steering models for any intermediate vehicle speeds. This multi-model approach is an alternative to nonlinear modelling. Although nonlinear models of the vehicle dynamics have also been obtained (Sharma and Sutton, 2012), they are not discussed in the present work.

4.3 Navigation

For surface vehicles, navigation entails knowing one’s position and heading at each instant, and is particularly important for autonomous USV operation because this information is fed to both the guidance system and the autopilot (Figure 1.1).

Kalman filtering is presented in this section as an efficient navigational tool for heading estimation. In the present work, vehicle localisation is directly assimilated through GPS fixes, and the techniques developed will be applied to heading estimation. Nevertheless, the KF-based techniques proposed are of general application.

4.3.1 Kalman filtering applied to heading estimation of the Springer USV

The principles behind Kalman filtering and its recursive formulation were detailed in Section 2.2.2. This section applies the KF to the problem of estimating the heading of the Springer USV.

The steering model of the vehicle (Equations 4.2 and 4.3) can be used in various ways with both gyroscope and compass readings in a KF. Assume that in addition to said model, a characterisation of input disturbances representative of random environmental forces due to waves and wind can be obtained and
described as a stochastic, normally distributed white noise sequence. The steering model is then given by

\[
x(k + 1) = A x(k) + B u(k) + \omega(k) \tag{4.6}
\]

\[
y(k) = C x(k) \tag{4.7}
\]

where \( u(k) = n_d(k) \), \( y(k) = \dot{\theta}(k) \), and \( \omega(k) \) is the input disturbance with variance \( Q \). The measurement of the output is then given by the gyroscope reading, \( z_{\text{gyro}}(k) \), which can be assumed to be unbiased and provide the turning rate with an RMS error of 0.05 deg/s (Table 3.3), that is,

\[
z_{\text{gyro}}(k) = y(k) + v(k) \tag{4.8}
\]

with \( v(k) \sim N(0, R) \), \( R = 0.05^2(\text{deg/s})^2 \).

Alternatively, the rate change in heading given by Equation 4.7 can be integrated according to Equation 4.4, and an augmented state-space description of the yaw dynamics with the heading angle as output can be written as

\[
\begin{bmatrix}
  x(k + 1) \\
  \theta(k + 1)
\end{bmatrix} =
\begin{bmatrix}
  A & 0 \\
  T_s C & 1
\end{bmatrix}
\begin{bmatrix}
  x(k) \\
  \theta(k)
\end{bmatrix} +
\begin{bmatrix}
  B \\
  0
\end{bmatrix}
\begin{bmatrix}
  u(k) \\
  \omega(k)
\end{bmatrix} \tag{4.9}
\]

\[
y(k) =
\begin{bmatrix}
  0 \\
  1
\end{bmatrix}
\begin{bmatrix}
  x(k) \\
  \theta(k)
\end{bmatrix} \tag{4.10}
\]

with \( u(k) = n_d(k) \) and \( \omega(k) \sim N(0, Q) \) the environmental input disturbance as before, and \( y(k) = \theta(k) \) the heading angle of the vehicle. The measurement is then given by the compass reading, \( z_{\text{compass}}(k) \), assumed to provide an unbiased measurement of the actual heading with an RMS error of 0.5° or 1°, depending on the selected compass (Table 3.3).

\[
z_{\text{compass}}(k) = y(k) + v(k) \tag{4.11}
\]

with \( v(k) \sim N(0, R) \), \( R = (0.5 \text{ deg})^2 \) or \( (1 \text{ deg})^2 \).

Another option is to modify Equation 4.10 to include both turning rate and heading as outputs; the model equations in this case are
\[
\begin{bmatrix}
x(k+1) \\
\theta(k+1)
\end{bmatrix} =
\begin{bmatrix}
A & 0 \\
T_s C & 1
\end{bmatrix}
\begin{bmatrix}
x(k) \\
\theta(k)
\end{bmatrix} +
\begin{bmatrix}
B \\
0
\end{bmatrix} u(k) +
\begin{bmatrix}
\omega(k) \\
0
\end{bmatrix}
\]  
(4.12)

\[
\begin{bmatrix}
y_1(k) \\
y_2(k)
\end{bmatrix} =
\begin{bmatrix}
C & 0 \\
0 & 1
\end{bmatrix}
\begin{bmatrix}
x(k) \\
\theta(k)
\end{bmatrix}
\]  
(4.13)

with \( u(k) = n_d(k) \) and \( \omega(k) \sim N(0, Q) \) the environment disturbance, \( y_1(k) = \dot{\theta}(k) \), and \( y_2(k) = \theta(k) \). The measurements of the outputs consist of the gyroscope and compass readings,

\[
\begin{bmatrix}
z_{gyro}(k) \\
z_{compass}(k)
\end{bmatrix} =
\begin{bmatrix}
y_1(k) \\
y_2(k)
\end{bmatrix} +
\begin{bmatrix}
v_1(k) \\
v_2(k)
\end{bmatrix}
\]  
(4.14)

with \( v_1(k) \sim N(0, R_1) \), \( R_1 = 0.05^2 (deg/s)^2 \) and \( v_2(k) \sim N(0, R_2) \), \( R_2 = (0.5 \, deg)^2 \) or \( (1 \, deg)^2 \).

Yet another option would be to use a predictive model of the heading based only on the gyroscope reading, and use the compass reading in the corrective stage of the filter. Denote the actual turning rate of the vehicle, \( \dot{\theta} \), as \( \Omega_i \), and the gyroscope reading as \( \Omega_0 \) (Figure 4.3), given by

\[\Omega_0 = \Omega_i + \omega\]  
(4.15)

with \( \omega \) the gyroscope measurement noise, which, when sampled at \( T_s = 1s \), can be described as \( \omega(k) \sim N(0, Q) \), \( Q = 0.05^2 (deg/s)^2 \) (Table 3.3). Discrete integration of the gyro-rate (Equation 4.4) provides the heading angle at each sampling time, which in terms of the gyroscope reading can be written as

\[
\theta(k+1) = \theta(k) + T_s \times [\Omega_0(k) - \omega(k)]
\]  
(4.16)

Fig. 4.3 Gyro measurement model: \( \Omega_i \) is the actual rate of change of heading of the vehicle whereas \( \Omega_0 \) is the value output by the gyroscope mounted on the vehicle.
Letting the state $x(k)$ represent the heading angle at each time step, (Equation 4.16) can be viewed as the state equation for predicting the next state given the current state and known input,

$$x(k + 1) = x(k) + T_s u(k) - T_s \omega(k)$$

(4.17)

where the input $u(k)$ is the gyroscope reading, $\Omega_0(k)$, whilst the measurement equation is given by the compass measurement model,

$$z^{\text{compass}}(k) = x(k) + v(k), \, v(k) \sim N(0, R), \, R = (0.5 \text{ deg})^2 \, \text{or} \, (1 \text{ deg})^2$$

(4.18)

### 4.3.2 Gyroscope and compass Kalman filter

This section motivates the use of a KF data-fused heading estimate.

Given an initial heading estimate $\hat{\theta}(0)$, one can, of course, estimate the subsequent headings from the successive gyroscope measurements alone,

$$\hat{\theta}(k) = \hat{\theta}(0) + T_s \sum_{i=0}^{k-1} \Omega_0(i)$$

(4.19)

If it is assumed that the time-sampled gyroscope noise sequence $\{\omega(k)\}$ is white and normally distributed with variance $Q$, and $E\hat{\theta}(0) = \theta(0)$, then although the estimate is unbiased in the sense that

$$E\hat{\theta}(k) = E\hat{\theta}(0) + T_s \sum_{i=0}^{k-1} E\Omega_0(i) = \theta(0) + T_s \sum_{i=0}^{k-1} \Omega_0(i) = \theta(k)$$

(4.20)

its variance grows linearly with time,

$$var\hat{\theta}(k) = var\hat{\theta}(0) + T_s^2 var \sum_{i=0}^{k-1} \Omega_0(i) = var\hat{\theta}(0) + T_s^2 \sum_{i=0}^{k-1} var\omega(i) = var\hat{\theta}(0) + k T_s^2 Q$$

(4.21)

that is, with roughly 68% probability the RMS error of the estimate will lie between $\pm \sqrt{kT_sQ^{1/2}}$ after $k$ time-steps, or with 50% probability between $\pm 0.6745\sqrt{kT_sQ^{1/2}}$.

Not only does the variance of the estimation error increase with time, but also, the resulting error sequence will no longer be white, but rather, what is known as a random walk, causing the estimate to drift from the true value, a
phenomenon which in this case is commonly referred to as **gyro integration drift**.

Consider the waypoint tracking mission described in Section 4.1. Figure 4.4 depicts, for an initial 100 time-step simulation, the actual turning rate of the vehicle, $\Omega_i(k)$, as compared to simulated gyroscope readings of the same, $\Omega_0(k)$, (Figure 4.4a), for which the noise $\omega(k)$ is generated pseudo randomly according to a normal distribution with zero mean and variance $Q = 0.05^2(\text{deg/s})^2$ (Figure 4.4c). The actual vehicle heading ($\theta(k) = \theta(0) + T_s \sum_{i=0}^{k-1} \Omega_i(i); \theta(0) = 0$) is shown in Figure 4.4b along with the heading obtained from integration of the gyroscope readings (Equation 4.19, with $\dot{\theta}(0) = 0$), and the difference between these two, $\hat{\theta}(k) - \theta(k) = T_s \sum_{i=0}^{k-1} \omega(i)$, is plotted in Figure 4.4d. The plot shown in Figure 4.4d is characteristic of a random walk, and the accumulated heading prediction error is around $0.8^\circ$, which is within normal expectation, given that $\sqrt{100 \times 1 \times 0.05} = 0.5^\circ$. (On a side note, it should be mentioned that in this simulation, the water surface was assumed to be moving with a current of $0.1 \text{ m/s}^{-1}$ in a northerly direction. This explains the negative heading rate of the vehicle, which, moving eastwards en route to waypoint 1, also finds itself drifting northwards, and hence the autopilot commands the vehicle to turn so that it is always pointing towards the target waypoint).

Conversely, Figure 4.4b also shows simulated compass readings, $z_{\text{compass}}(k)$, generated as pseudo random values drawn from the normal distribution with mean equal to the actual vehicle heading at each time-step, $\theta(k)$, and variance $R = 1 \text{ deg}^2$,

$$z_{\text{compass}}(k) = \theta(k) + \nu(k), \{\nu(k)\} \text{ white noise } \sim N(0, R = 1) \tag{4.22}$$

Unlike the gyroscope estimate, the compass readings do not suffer from integration drift; however, the measurement noise is typically larger than that of an inertial sensor (as shown in Table 3.3), and smoothing of the signal due to integration is not present. Hence direct measurement of the heading produces a relatively noisy signal.
Figure 4.4 Initial 100 time-step simulation of the way-point tracking mission: (a) actual turning rate of the vehicle and (noisy) gyroscope measurement; (b) actual heading of the vehicle, predicted heading based on dead-reckoning from gyroscope readings, (noisy) compass measurements, and Kalman filter heading estimate; (c) gyroscope noise; (d) integrated gyroscope noise (random walk).

In order to exploit the precision of the gyroscope for short-term estimation together with the long-term stability of compass measurements, the KF based on the model described by Equations 4.17 and 4.18 fuses the heading based on gyroscope prediction with compass measurement at each time step.

The sequence of KF estimates of the heading for the aforesaid vehicle simulation, \( \hat{\theta}_{KF}(k) \), given an initial estimate of the state \( \hat{x}(0) = \hat{\theta}_{KF}(0) = 0 \) and of the state estimate error covariance \( P(0) = 0 \), is shown in Figure 4.4b.
quantitative assessment of each of the estimation methods is provided in Table 4.2. It can be seen that whilst the gyroscope predictions are more accurate than the compass measurements for short term prediction, in the long term, the gyroscope-based prediction RMS error increases indefinitely with time. On the other hand, the KF-based prediction, which fuses the gyroscope prediction with compass measurement, provides a closer estimate than the other two individually (note this is true at least statistically, even though what is shown here is a particular realisation of said statistics).

Table 4.2. comparison of heading estimate errors

<table>
<thead>
<tr>
<th>method for heading estimation</th>
<th>heading RMS error from k=1 to N (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>N = 100</td>
</tr>
<tr>
<td>gyroscope readings integration (dead reckoning)</td>
<td>0.312</td>
</tr>
<tr>
<td>$$e_{gyro} = \left{N^{-1} \sum_{k=1}^{N} [\theta(k) - \hat{\theta}(k)]^2 \right}^{1/2}$$</td>
<td>0.312</td>
</tr>
<tr>
<td>compass measurements</td>
<td>0.927</td>
</tr>
<tr>
<td>$$e_z = \left{N^{-1} \sum_{k=1}^{N} [\theta(k) - z_\theta(k)]^2 \right}^{1/2}$$</td>
<td>0.927</td>
</tr>
<tr>
<td>Kalman filter estimate</td>
<td>0.013</td>
</tr>
<tr>
<td>$$e_{KF} = \left{N^{-1} \sum_{k=1}^{N} [\theta(k) - \hat{\theta}_{KF}(k)]^2 \right}^{1/2}$$</td>
<td>0.013</td>
</tr>
</tbody>
</table>

4.4 Guidance

Different guidance strategies used in marine environments to guide the vehicles are illustrated in Annamalai (2012). The simplest guidance strategy is waypoint LOS and is utilised herein. It is briefly illustrated as follows.

Based on the current estimated position of the USV (e.g. from a GPS receiver) and the coordinates of the next way-point to be reached, the desired or reference heading angle based on LOS is calculated as follows:

$$r(k) = \arctan\left(\frac{y_d(k)-y(k)}{x_d(k)-x(k)}\right)$$

(4.23)

where \((x, y)\) is the current location of the vessel and \((x_d, y_d)\) the target coordinates. In practice, because the inverse of the tangent is restricted to (-90°,
90°), the four quadrant inverse tangent, \( \arctan2(y_d(k) - y(k), x_d(k) - x(k)) \), which takes into account the signs of both arguments, is used instead. Also, as the reference (or desired) heading angle changes, particularly when a waypoint is reached and the reference changes abruptly targeting the next waypoint, care is taken to ensure that the vehicle is directed to turn toward the reference angle in the direction that requires the least change in heading.

In addition to updating the reference heading, the guidance system implemented for Springer keeps track of the mission status, which includes a log of the waypoints reached or missed and the current target waypoint, as well as the total distance travelled, deviation from the ideal trajectory, and controller energy consumed. These are updated every sampling instant based on the current position of the USV. All of these concepts are described next.

In order to decide whether a waypoint has been reached or not, the guidance system considers a circle of acceptance (COA) around each of these (Figure 4.5). At each sampling instant, the guidance system calculates the distance left to the next way-point according to

\[
\rho = \sqrt{[x_d(k) - x(k)]^2 + [y_d(k) - y(k)]^2}
\]  

(4.24)

When this distance is less than the radius of the COA, it is considered that the way-point is reached, and the guidance system directs the vessel to the next way-point.

Figure 4.5. Deviation at time \( k \)
However, the vessel might pass by the vicinity of a way-point without entering the COA. This condition is determined by checking the derivative \( d\rho/dt \), which when switches from negative to positive, indicates that the vessel has missed the way-point. In this case, the guidance system also directs the vessel toward the next way-point.

The vessel normally follows a path different from the ideal one. Several performance indices are used to assess the trajectories followed, which the guidance system computes at each time step and keeps track of. The deviation from the ideal trajectory can be measured as

\[
rd(k) = \sqrt{PB^2 + PB_1^2 - 2PB \cdot PB_1 \cos(\alpha)}
\]  

(4.25)

where \( PB \) is the distance, at time \( k \), to the next waypoint from the position of the vehicle were it on the ideal path, and \( PB_1 \) the distance to the next waypoint from its the actual position at time \( k \), \( \alpha \) being the angle between the two vectors, as shown in Figure 4.5.

Finally, the average controller energy \( \overline{CE_u} \), measured in \( (\text{rpm})^2 \), is defined as

\[
\overline{CE_u} = \frac{1}{N} \sum_{k=1}^{N} \left[ \frac{u(k)}{60} \right]^2
\]  

(4.26)

where \( N \) is the total number of time steps and \( u \) the controller effort or applied signal at time \( k \) in rpm.

### 4.5 Control

Feedback control of the Springer USV heading intends to generate the adequate control signal (differential propeller revolution rates) in order to steer the vehicle so that its heading matches that of the reference or desired heading \( r(k) \).

Several control strategies have been implemented for the autopilot of Springer. Proportional-Integral-Derivative (PID) and state-feedback strategies are
described in this section. More sophisticated controllers for the vehicle, such as model predictive based controllers, are topics of another research package.

**4.5.1 PID control**

One of the most common used control algorithms in industry, the PID controller can be formulated as the sum of three terms,

\[
 u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{d}{dt}e(t)
\]  

(4.27)

where \( u(t) \) is the controller output, and the error term \( e(t) \) is the difference between the desired plant output or reference and the actual plant output. The parameters \( K_p, K_i \) and \( K_d \) are, respectively, the proportional, integral, and derivative gains. The proportional term tries to correct the present error, whilst the integral term responds to the accumulated error and guarantees zero error in steady-state. The derivative term acts upon a prediction of the future error based on the current rate of change of error (see, e.g. Astrom and Hagglund, 1995).

In discrete-time, the “velocity algorithm” for a PID controller, derived by approximating the first-order derivatives via backward finite differences, is given by

\[
 u(k + 1) = u(k) + K_p \left[ \left( 1 + \frac{T_d}{T_i} + \frac{T_d}{T_s} \right) e(k) + \left( -1 - 2 \frac{T_d}{T_s} \right) e(k - 1) + \right. \\
\left. \frac{T_d}{T_s} e(k - 2) \right]
\]  

(4.28)

where \( T_i = K_p/K_i \) and \( T_d = K_d/K_p \).

The dynamics of the closed-loop system depends upon the values of the controller gains, and several tuning methods are available in the literature (Astrom and Hagglund, 1995).

In the case of the *Springer*, the error term at each time step is calculated as the difference between the reference heading (Equation 4.23) computed by the guidance system, and the KF estimate of the actual heading of the vehicle.
4.5.2 State feedback control

Consider the augmented state equation 4.9 which describes the steering dynamics of the vehicle, and is abbreviated as

\[
\tilde{x}(k+1) = \tilde{A}\tilde{x}(k) + \tilde{B}u(k) + \tilde{\omega}(k)
\]  

(4.29)

A state feedback control law is formulated by

\[
u(k) = -K\tilde{x}(k) + K_s r(k)
\]  

(4.30)

where \(K\) is the state feedback gain, chosen to specify some adequate closed-loop dynamics, for example, via Ackermann’s pole placement method (see e.g. Shinners, 1998), and \(K_s\) is a scaling gain selected a posteriori to regulate the steady-state gain of the closed-loop system. The closed-loop dynamics is then given by

\[
\tilde{x}(k+1) = (\tilde{A} - \tilde{B}K)\tilde{x}(k) + \tilde{B}K_s r(k) + \tilde{\omega}(k)
\]  

(4.31)

In the case of Springer, the state vector of the dynamic steering model is not measurable, and the KF estimate of the (augmented) state vector is used in Equation 4.30 instead of the actual state vector.

4.6 Mission Simulation

The way point tracking mission described in Section 4.1 is simulated in this section utilising the navigation, guidance and control systems described throughout this chapter. This serves to detail further implementation considerations.

The position of the vehicle is simulated at each time step according to Equation 4.5, with \(v = 1.5 \text{ ms}^{-1}\). The vehicle’s yaw rate is simulated according to the state space model give by Equations 4.6 and 4.7, with \(etg(A) = 0.2 \pm i0.25\), and an input disturbance \(\omega(k) \sim N(0, 25 \text{ rpm}^2)\). The heading is obtained by Equation 4.4, with an initial vehicle heading of 0°. In addition, a current of \(0.15 \text{ ms}^{-1}\) is simulated that transports the vehicle 0.15 m northwards each time step (\(T_s = 1 \text{ s}\)), without affecting its heading.
The actual vehicle position is assumed to be known in order to calculate the reference angle at each time step according to Equation 4.23. The heading of the vehicle is estimated by the gyroscope and compass KF described in Section 4.3.3, with the gyroscope readings generated as the actual simulated yaw rate of the vehicle with an added normally distributed pseudo random noise with variance \(0.05^2(\text{deg}/\text{s})^2\). The compass readings are generated by superimposing a normal pseudo random noise with variance \(1^2(\text{deg})^2\) to the simulated vehicle heading. The KF is initialised with zero state vector and zero error covariance.

It should be noted that the simulated vehicle heading and hence the KF heading estimate is not limited to between 0 and 360°. Thus, the estimate must first be wrapped to the interval \([0, 360^\circ]\) before comparing it with the guidance reference \(r(k)\), which is restricted to this interval.

A proportional feedback controller is used with \(K_p = 25\), with the output error calculated as \(e(k) = r(k) - \hat{\theta}(k)\), where \(r(k)\) is the reference heading (Equation 4.23) and \(\hat{\theta}(k)\) the KF estimate of the vehicle’s heading. To ensure that the vehicle always turns towards the target in the direction of the shorter angle, the following corrections are then applied:

\[
e(k) := e(k) - 360^\circ \quad \text{if} \quad e(k) > 180^\circ \tag{4.32}
\]
\[
e(k) := e(k) + 360^\circ \quad \text{if} \quad e(k) > -180^\circ \tag{4.33}
\]

The trajectory of the vehicle is shown in Figure 4.6, in which the effect of the surface current is apparent. Figure 4.7 plots the reference heading against the actual heading of the vehicle. The sudden changes in the references correspond to the time instants at which a waypoint has been reached and the following one is targeted. If the vehicle followed the ideal path, then the reference heading would be composed purely of a piecewise step function, but is only approximately so because of the deviation from the ideal path, which requires the reference to be corrected at each time step. The deviation is caused by the current and stochastic input disturbance, as well as the limited turning rate of the vehicle. In order to simulate this limitation, the autopilot output is not allowed to surpass \(\pm 900 \text{rpm}\) or change by more than \(900 \text{rpm}\) from one time step to the next (in simulation, these saturation limits are simply imposed programmatically on the controller calculated output).
The difference between the simulated vehicle’s heading and the KF estimate is shown in Figure 4.8. The RMS error of the filter is $0.21^\circ$, substantially less than that of the compass, which for this simulation is $0.97^\circ$ (in line with the expected RMS error of $1^\circ$).

Finally, the controller output is shown in Figure 4.9. Notice that the spikes in the control action occur during the instants of brusque changes in the reference heading (upon arrival at a way point).

Figure 4.6 Vehicle trajectory during way point mission simulation.

Figure 4.7 Reference heading and vehicle heading
Although the choice of controller is not a topic of discussion in the present thesis, in order to provide the reader with a better intuition of its role, and distinguish its effect on the overall mission from those of the navigation and guidance subsystems, a simulation with an increased controller gain of $K_p = 50$ is shown in Figure 4.10. The corresponding time history of the reference and vehicle heading angles are plotted in Figure 4.11, and that of the control output in Figure 4.12. It should be noted that the closed-loop dynamics, being more underdamped than in the previous case because of the more aggressive controller, leads to larger oscillations around the reference, both in the variable being controlled (vehicle heading) and in the trajectory of the vehicle. Table 4.3 summarises some of the mission performance indices, in which the stark difference in average controller energy is evidenced.
Figure 4.10 Vehicle trajectory during way point mission simulation ($K_p = 50$)

Figure 4.11 Reference heading and vehicle heading ($K_p = 50$)
Table 4.3 Mission performance parameters

<table>
<thead>
<tr>
<th></th>
<th>Case 1: $K_p = 25$</th>
<th>Case 2: $K_p = 50$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$distance$ (m)</td>
<td>1234</td>
<td>1232</td>
</tr>
<tr>
<td>$rd$ (m)</td>
<td>8.7</td>
<td>8.6</td>
</tr>
<tr>
<td>$CE_u ((rps)^2)$</td>
<td>7.9</td>
<td>27.7</td>
</tr>
</tbody>
</table>

### 4.7 Summary

A waypoint tracking mission has been described in this chapter, and will serve as an example on which to simulate various navigational algorithms developed in this thesis. A motion model of the vehicle has also been described, and is used as a proxy of the actual vehicle in simulation studies.

The remainder of the chapter has been dedicated to describing the NGC subsystems of Figure 1.1 used for the autonomous operation of Springer, concluding with a simulation of the way point tracking mission initially described, showing the outputs of each of these subsystems.

Though the workings of each subsystem has been detailed, particular emphasis has been given to the KF as a tool for estimating the heading of the vehicle. Several KF configurations based on the vehicle’s steering model, gyroscope readings, and compass measurements, or combinations thereof, have been described. The purpose of KF has been conveyed through an example demonstrating the superiority of fused sensor data estimates over individual
sensor readings. Having established the virtue of Kalman filtering, the next chapter will expose its main shortcoming, which will serve as the motivation for the research carried out in this thesis.
Chapter 5
Interval Kalman Filtering

“We demand rigidly defined areas of doubt and uncertainty!”
— Douglas Adams

The previous chapter illustrated the benefit of using Kalman filtering for heading estimation of the Springer as opposed to directly using noisy sensor measurements. In this chapter it is firstly demonstrated that the KF’s optimality is highly dependent on the accuracy of the model considered, which constitutes a serious practical limitation. In view of this limitation, the concept of interval model, or model that is described with a finite degree of uncertainty, is described, and the Kalman filtering scheme for such models, the IKF, is detailed. The IKF is then applied to the problem of heading estimation of the Springer USV, an exercise that exposes the main practical difficulty of implementing the IKF algorithm.

5.1 Kalman Filter Limitations

Section 4.3.3 justified the use of a KF to estimate the heading of the Springer USV. However, for the KF to provide optimal estimates, the predictive and measurement models (Equations 2.29 to 2.32) must be described accurately. The following example illustrates the effect of deficient modelling on the KF estimates.

Consider a vehicle dynamic steering model of the kind described in Section 4 (Equations 4.6 and 4.7),

\[ x(k+1) = Ax(k) + Bu(k) + \omega(k) \]  \hspace{1cm} (5.1)

\[ y(k) = Cx(k) \]  \hspace{1cm} (5.2)
with

\[
A = \begin{bmatrix} 0.8 & -0.2225 \\ 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 0.004225 \end{bmatrix},
\]

\[
\omega \sim N(0, Q), \quad Q = \text{cov}(\omega) = 10^{-4} \times \text{diag}(1,1), \quad T_s = 1 \text{ s}
\]  

(5.3)

in which \( \mathbf{x} \in \mathbb{R}^{2\times1} \) is the state vector, \( u \) the applied differential propeller speed in rpm, \( y \) the rate change in heading of the vehicle in deg/s, and \( \omega \) represents a random input disturbance. The output of this model can be integrated and combined with compass readings in a KF as described in Section 4.3.1. Let the compass readings be given by

\[
z_{\text{compass}}(k) = C \mathbf{x}(k) + \nu(k), \text{ with } \nu(k) \sim N(0, R), \quad R = 1^2(\text{deg})^2
\]  

(5.4)

The KF can then be applied to the combined model

\[
\bar{\mathbf{x}}(k+1) = \tilde{\mathbf{A}} \bar{\mathbf{x}}(k) + \tilde{\mathbf{B}} u(k) + \tilde{\omega}(k)
\]  

(5.5)

\[
z_{\text{compass}}(k) = \tilde{\mathbf{C}} \bar{\mathbf{x}}(k) + \nu(k)
\]  

(5.6)

\[
\tilde{\mathbf{A}} = \begin{bmatrix} A & 0 \\ T_s C & 1 \end{bmatrix}, \quad \tilde{\mathbf{B}} = \begin{bmatrix} B \\ 0 \end{bmatrix}, \quad \tilde{\mathbf{C}} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}, \quad \bar{\mathbf{x}} = \begin{bmatrix} \mathbf{x} \\ \theta \end{bmatrix}, \quad \tilde{\omega} = \begin{bmatrix} \omega \\ 0 \end{bmatrix}
\]  

(5.7)

where \( \theta \) represents the heading of the vehicle, and in which it is assumed that \( \{x(0), \omega(0), ... \omega(k), \nu(0), ..., \nu(k)\} \) are mutually independent.

Consider however that the steering dynamics have been incorrectly modelled, such that all the model coefficients (Equation 5.3) have been underestimated by 5%. In other words, that the true vehicle steering dynamics is given instead by

\[
x(k+1) = A_1 x(k) + B_1 u(k) + \omega(k)
\]  

(5.8)

\[
y(k) = C_1 x(k)
\]  

(5.9)

with

\[
A_1 = A + 0.05 \times \text{abs}(A), \quad B_1 = B + 0.05 \times \text{abs}(B), \quad C_1 = C + 0.05 \times \text{abs}(C),
\]

\[
\omega \sim N(0, Q), Q = \text{cov}(\omega) = 10^{-4} \times \text{diag}(1,1), \quad T_s = 1 \text{ s}
\]  

(5.10)
The vehicle heading rate is simulated according to Equations (5.8) to (5.10) and then integrated according to Equation (4.4) to obtain its heading, for the following input sequence:

\[
    u(k) = \begin{cases} 
        20 \text{ rpm} & 0 \leq k < l/3 \\
        50 \text{ rpm} & l/3 \leq k \leq 2l/3 \\
        -15 \text{ rpm} & 2l/3 \leq k \leq l 
    \end{cases}
\]  (5.11)

where \( l = 100 \) is the number of simulation time-steps. The initial state vector is set to \( x(0) = 0 \) and the initial heading \( \theta(0) = 0 \). The disturbance \( \omega(k) \) is generated pseudo randomly according to the statistics given in Equation (5.10). Figure 5.1a depicts the deterministic input sequence \( u(k) \), and the resulting heading is shown in Figure 5.1b.

Also shown in Figure 5.1b are the simulated compass measurements, generated according to Equation 5.4, and the KF estimate according to the (incorrect) nominal model given by Equations 5.5 to 5.7, initialised with the correct initial state and zero error covariance. It is clear that whilst the noisy compass readings are centred around the true vehicle heading, the KF estimate is systematically biased due to the incorrect model assumed.

Alternatively, a KF based on fusing rate estimates from the steering model with gyroscope readings could be used, as detailed in Section 4.3.1, that is, based upon the equations

\[
    x(k+1) = A \, x(k) + B \, u(k) + \omega(k) 
\]  \hspace{1cm} (5.12)

\[
    z^{gyro}(k) = C \, x(k) + v(k) 
\]  \hspace{1cm} (5.13)

with the nominal values given in Equation 5.3, and the gyroscope measurement noise modelled as

\[
    v(k) \sim N(0, R), R = 0.05^2 (\text{deg/s})^2 \]  \hspace{1cm} (5.14)

For the same controllable input as before given by Equation 5.11, Figure 5.2 depicts the KF estimate of the rate change in heading based on Equations 5.12 to 5.14. The KF's initial estimate was taken equal to the true initial state (zero), and the initial error covariance as zero. The true vehicle turning rate and noisy gyroscope readings are also plotted on the same figure. Again, because of the incorrect model assumed by the KF, the estimates of the turning rate are systematically biased.
Figure 5.1 (a) system input, or difference in propeller revolution rates (rpm); (b) actual vehicle heading, compass measurements, and KF estimates based on nominal model (deg).

Figure 5.2 KF heading rate estimate, gyroscope readings and actual vehicle heading rate change.
In practice, 100 percent accurate system modelling is utopian: at best, even in the case of systems modelled via a first-principles approach, small modelling errors exist because values of parameters (mass, geometry, resistivity, permittivity, absorbance, etc.) can only be measured with a finite tolerance, not to mention that such parameters are often sensitive to external variable factors, such as temperature, etc. Tolerances, though, provide bounds to the models obtained. In the case of the SI steering models obtained for Springer, it suffices to consider that the conditions during which the modelling data were collected may not be exactly the same as when the model is applied in operation.

5.2 The Interval Kalman Filter

Interval analysis is a field of mathematics which began to be formally studied in the 1950s with the intention of finding a way to bound rounding errors in finite-precision numerical computations. As computer representation is limited by the machine epsilon, only a small subset of real numbers can be represented accurately on a computer, but every real number can be represented by an enclosure consisting of two bounding, machine-representable values, by appropriate upward and downward rounding. Consequently, an interval arithmetic (IA) was defined (Moore 1966) in such a way that calculations on interval values yield further intervals which guarantee to enclose (though not necessarily be equal to) the range of possible results, a fundamental quality known as the inclusion property. By virtue of which, if an initial bound for the error is known, then the error propagation inherent in any calculation carried out with finite precision is automatically bounded by the result of computing this propagation using IA. Since the publication of the first book on interval analysis (Moore, 1966), there has been a keen interest in applying interval based verified-computing in numerous fields, ranging from computer-assisted proofs in mathematical analysis (Lanford III, 1986) to practical engineering and industrial applications (Corliss, 1990). An introduction to interval analysis with a review of applications can be found in Kearfott (1996) and in Alefeld and Mayer (2000).

In the late 90s a KF applied to dynamical interval systems, that is, systems whose parameters are described in terms of intervals, was proposed by Chen et al. (1997). Uncertainty in system modelling is often naturally described in this form. Physical parameters are usually not known precisely but specified with a certain tolerance, or have a varying nature: for example, with reference to the vehicle steering, the dynamics depends, amongst other parameters, on its mass. In manned marine vehicles this varies depending on the number of passengers
aboard, but even in the case of USVs, payload may vary depending on the mission as on-board equipment may be configurable, and fuel-driven USVs typically have a large fuel capacity to total platform weight ratio, making such variations significant. In the *Springer* state-space steering model (Equations 5.1 to 5.3), the coefficients of the matrices were obtained using SI techniques using a certain data set registered during a specific trial. During other trials, the values obtained may vary slightly due to reasons such as those previously outlined. The effect of incorrectly describing these values was illustrated in Figures 5.1 and 5.2. However, if a whole range of trials is performed, the varying results can be enclosed in intervals, resulting in an interval system model that contains each of the individual models obtained. In this way, all the possible dynamics are taken into account.

Chen et al. (1997) used the theory of IA to construct a KF for the interval system model. They obtained the IKF equations using the same derivation as the regular KF. Suppose some elements of the matrices A, B and C are uncertain within some definite bounds. The system state equation and measurement model then take on the form

\[ x(k + 1) = A^I x(k) + B^I u(k) + \omega(k) \]  

\[ z(k) = C^I x(k) + \nu(k) \]

where \( M^I = M \pm \Delta M = [M - |\Delta M|, M + |\Delta M|] \) for \( M \in \{A, B, C\} \). Conditions similar to those hypothesised by the KF regarding the random variables are assumed, namely that \( \omega(k) \) and \( \nu(k) \) are white noise sequences with zero-mean Gaussian distributions with known variances \( var(\omega(k)) = Q \), \( var(\nu(k)) = R \), and that \( E[\omega(l)\nu^T(k)] = 0 \forall l,k \), \( E[x(0)\omega^T(k)] = 0 \), and \( E[x(0)\nu^T(k)] = 0 \forall k \).

The IKF algorithm is summarised by Equations 5.17 to 5.21 (Chen et al., 1997), which mimic those of the ordinary KF but are described in terms of intervals. Given an initial interval estimate \( \hat{x}^I(0) \), and its uncertainty, characterized by \( P^I(0) \equiv var(\hat{x}^I(0)) \), together with the input to the system and the measurement of the output at each time-step, the resulting state estimate is an interval vector \( \hat{x}^I(k) \) at each time-step \( k \), providing an upper and lower boundary to the estimate, as illustrated in Figure 5.3.
IKF equations

Prediction:

\[
\hat{x}^l(k|k - 1) = A^l \hat{x}^l(k - 1|k - 1) + B^l u(k - 1)
\]  

(5.17)

\[
P^l(k|k - 1) = A^l P^l(k - 1|k - 1) A^{lT} + Q
\]  

(5.18)

Kalman gain:

\[
K^l(k) = P^l(k|k - 1) C^{lT} \left\{ C^l P^l(k|k - 1) C^{lT} + R \right\}^{-1}
\]  

(5.19)

Correction:

\[
\hat{x}^l(k|k) = \hat{x}^l(k|k - 1) + K^l(k) \{ z(k) - C^l \hat{x}^l(k|k - 1) \}
\]  

(5.20)

\[
P^l(k|k) = \{ I - K^l(k) C^l \} P^l(k|k - 1)
\]  

(5.21)

Figure 5.3 IKF estimate depicting its upper and lower boundaries.

Having been derived from the same principles, the IKF is statistically optimal in the same sense as the standard KF, and it maintains the same recursive formulation. However, the main advantage of the computed interval estimates, as opposed to point estimates, is that they guarantee to contain all the KF estimates of the individual models contained in the interval model, a consequence of the inclusion property of IA by which if initial imprecise data is enclosed within rigorous bounds, then computation with these bounds will carry on yielding rigorous bounds of the actual solution range.
This can be important if one is to have confidence in the estimate. If the true system dynamics is known to be contained in the interval model, then the IKF provides a guaranteed enclosure of the optimal state estimate. While the precise value of this estimate will not be known, an interval may be acceptable for the required purpose: for example, if the goal is to maintain a state variable between two limiting values, then as long as the interval estimates remain within these limits no control action is required. Likewise, should the estimation boundaries permeate into and undesired operating region, this can be used to raise an alarm or trigger some other contingency mechanism.

5.3 Application of the IKF to the Navigation of *Springer*

To illustrate the application of the IKF algorithm to the estimation of the heading angle of the *Springer* USV, consider the following steering dynamic model described in Naeem et al. (2008), obtained through early SI trials:

\[
\begin{align*}
x(k + 1) &= A x(k) + B u(k) + \omega(k) \quad (5.22) \\
y(k) &= C x(k) \quad (5.23)
\end{align*}
\]

\[
A = \begin{bmatrix} 1.002 & 0 \\ 0 & 0.9945 \end{bmatrix}, B = \begin{bmatrix} 6.354 \\ -4.699 \end{bmatrix} \times 10^{-6}, C = [34.13 \quad 15.11], \\
\text{var}(\omega(k)) = Q = 10^{-9}
\]

with a sampling time of 1 s, in which \( u(k) \) represents the difference in propeller revolution rates in rpm and \( y(k) \) is the heading angle in radians with respect to some reference heading.

Now suppose that the values of \( B \) in Equation 5.24 are only known with a tolerance of 25%. The smallest interval system that contains all the possible crisp models is described by

\[
\begin{align*}
x(k + 1) &= A^I x(k) + B^I u(k) + \omega(k) \quad (5.25) \\
y(k) &= C^I x(k) \quad (5.26)
\end{align*}
\]

\[
A^I = \begin{bmatrix} 1.002 & 0 \\ 0 & 0.9945 \end{bmatrix}, B^I = \begin{bmatrix} [0.75 \times 6.354, \ 1.25 \times 6.354] \\ [-1.25 \times 4.699, \ 0.75 \times 4.699] \end{bmatrix} \times 10^{-6}, \\
C^I = [34.13 \quad 15.11], \text{var}(\omega(k)) = Q = 10^{-9}
\]

\[ 77 \]
Then, based on simulated values of $\omega(k)$, compass measurements $z^\text{compass}(k)$ according to Equation 5.4, and the deterministic input sequence $u(k)$ shown in Figure 5.4a, the KF and IKF heading estimates associated with the nominal and interval models, respectively, are depicted in Figure 5.4b, using zero initial conditions.

![Differential propeller speed](image1)

(a) System input: applied differential motor speed $n_d$ (rpm).

![Heading estimates](image2)

(b) Heading estimates: nominal KF and IKF (deg).

Figure 5.4 (a) System input: differential propeller revolution rates (rpm); (b) comparison of nominal-system KF heading estimate, and boundaries of IKF estimate (deg).

It is clearly seen in Figure 5.4b how the nominal-system KF estimate lies within the IKF boundaries, and this holds for KF estimates obtained using any crisp model contained in the interval model. However, another phenomenon is evidenced: that of the rapid separation of the IKF boundaries. A consequence of the inclusion property of IA, which on the one hand is the raison d’être of most interval analysis applications, on the other constitutes one of the major practical difficulties in implementing IA based algorithms, rendering results overly pessimistic and of little practical value. Hence, the consequently excessively conservative IKF boundaries may be attributed not to the theoretical framework of the algorithm laid out by Chen and co-researchers (1997), but with the way in which the IA calculations are implemented.
With regards to IA implementation, the IKF simulation was carried out with the aid of the open-source extension of MATLAB for IA, INTerval LABoratory (INTLAB), developed by Rump (1999). Numerous programming languages now contain libraries that extend the basic variable types to include intervals, and incorporate routines for carrying out IA operations (Kearfott, 1996). Albeit in themselves highly efficient, the sharpness of the results obtained in a computation involving IA is highly dependent on the particular numerical formulation adopted, whereby even with the aid of such software, a naive use of IA may provide results that are of no practical use. In fact, the cardinal reason for this reliance of the resulting interval on the precise numerical formulation adopted to compute some function is adequately known as the dependency problem of IA.

This phenomenon can be illustrated with a simple example: consider an interval value \( a^I = [-a, a], a \in \mathbb{R} \), and \( b \in \mathbb{R} \). Then,

\[
a^I \times (b - b) = a^I \times b - a^I \times b = [-a, a] \times b - [-a, a] \times b = [-a \times b, a \times b] - [-a \times b, a \times b] = [-2a \times b, 2a \times b]
\]

However, clearly the exact solution set is the single number zero. The overestimation occurs because, after expansion of the brackets in the initial product, the arithmetic does not remember that the interval variable \( a^I \) in both terms represent the same variable.

Overestimation caused by the dependency effect occurs because each occurrence of a variable in a mathematical expression is implicitly assumed to be independent; therefore, this effect is suppressed if the expression can be reformulated so that every interval variable appears only once. Though this may not be possible for complicated expressions, one can in general minimise the occurrence of a single variable at a time in the expression. Consider, for example, the computation of the first element of the \( a \) priori error covariance matrix \( P^I(k|k-1) \) (Equation 5.18). For a second-order system, this takes on the form

\[
p^I_{11}(k|k-1) = \{(a_{11}p_{11}(k-1|k-1) + a_{12}p_{21}(k-1|k-1))a_{11} + (a_{11}p_{12}(k-1|k-1) + a_{12}p_{22}(k-1|k-1))a_{12} + q_{11} : a_{ij} \in a^I_{ij} , p_{ij} \in p^I_{ij} \}
\]

(5.29)
Then, noting that $x'(y' + z') \subseteq x'y' + x'z'$ for any $x',y',z' \in I\mathbb{R}$, the expression in Equation (5.29) can be reformulated in the following ways, each one taking advantage of a different factorization and yielding a different interval enclosure for $p_{11}(k|k-1)$:

$$f_1 = (a_{11}'p_{11}'(k-1|k-1) + a_{12}'p_{21}'(k-1|k-1))a_{11} + (a_{11}'p_{12}'(k-1|k-1) + a_{12}'p_{22}'(k-1|k-1))a_{12} + q_{11}'$$  

(5.30)

$$f_2 = (a_{11}'^2 p_{11}'(k-1|k-1) + 2a_{11}'a_{12}'p_{12}'(k-1|k-1)) + (a_{12}'^2 p_{22}'(k-1|k-1) + q_{11}')$$

(5.31)

$$f_3 = a_{11}' (a_{11}'p_{11}'(k-1|k-1) + 2a_{12}'p_{12}'(k-1|k-1)) + (a_{12}'^2 p_{22}'(k-1|k-1) + q_{11}')$$

(5.32)

$$f_4 = (a_{11}'^2 p_{11}'(k-1|k-1) + a_{12}'(2a_{11}'p_{12}'(k-1|k-1) + a_{12}'p_{22}'(k-1|k-1)) + q_{11}'$$

(5.33)

The last three additionally making use of the symmetry of the error covariance matrix. It is not clear which one of these formulations, if any, yields the tightest enclosure, and even so, it would depend on the particular interval values taken on by each variable, which vary for each iteration. However, taking the intersection of all the intervals $f_i$, $f_\cap \equiv \cap_{i=1}^4 f_i$, yields the narrowest enclosure of $p_{11}'(k|k-1)$, that is, $f_\cap \subseteq f_i \forall i$. Using this approach for every component of each of the IKF equations for the interval model (Equations 5.25 to 5.27), the improved IKF boundaries obtained are shown in Figure 5.5b. Also shown are the boundaries obtained using naive IA, and the KF estimate based on the nominal model (Equations 5.22 to 5.24), which were shown in Figure 5.4a.

It is clear again from Figure 5.5b that without any special treatment, the widths of the interval bounds grow very quickly; however, in comparison to the improved boundaries it is evidenced that they encompass not just the actual set of possible KF estimates, but a much larger set that dwarfs the former, rendering the bounds meaningless in practice. The simple treatment carried out here to reduce the dependency effect has shown a significant decrease in the overestimation of the interval widths.
However, consider again the model given by Equations 5.1 to 5.3. The effect on the KF estimates of the nominal model’s underestimation of the actual parameter values by 5% was shown in Section 5.1. Nevertheless, if an interval model that contains both the true dynamics (Equations 5.8 to 5.10) as well as the nominal model were constructed, then the corresponding IKF estimates would include the optimal estimates (those of an ideal KF, or KF based on the true dynamic equations). The following interval model is centred around the nominal values (Equations 5.1 to 5.3), with interval widths of 5% on either side of the central values:

\[ x(k + 1) = A^I x(k) + B^I u(k) + \omega(k) \]  
\[ y(k) = C^I x(k) \]
\[
A^I = \begin{bmatrix}
[0.95 \times 0.8, 1.05 \times 0.8] & [-1.05 \times 0.2225, -0.95 \times 0.2225] \\
[0.95 \times 1, 1.05 \times 1] & [0, 0]
\end{bmatrix},
\]
\[
B^I = \begin{bmatrix}
[0.95 \times 1, 1.05 \times 1] \\
[0, 0]
\end{bmatrix},
\]
\[
C^I = \begin{bmatrix}
[0, 0] & [0.95 \times 0.004225, 1.05 \times 0.004225]
\end{bmatrix}
\]

For the input given by Equation 5.11 (reproduced in Figure 5.6a) and simulated gyroscope measurements according to Equations 5.13 and 5.14, and using zero initial conditions, Figure 5.6b depicts the bounds of the IKF estimates obtained using the naive IA and those obtained using the advantageous factorisations (Equations 5.30 to 5.33). Also shown are the KF estimates obtained using the nominal model, and those of an ideal KF, or KF that uses the true system dynamic equations.

(a) Applied differential motor speed \( n_d \) (rpm).

(b) Heading estimates from a KF based on the nominal model, an ideal KF, and two IKFs.

Figure 5.6. (a) system input, or difference in motor revolution rates (rpm); (b) heading estimates by a KF based on the nominal model, an ideal KF, an IKF using naive IA, and an IKF using the advantageous factorisations to reduce the dependency effect (deg).
In this case, clearly the advantageous factorisations do not provide enclosures to the actual solution sets that are significantly tighter than those obtained using naive IA. This is mainly due to the interval-valued state-transformation matrix which was used in this example. The recursive application of the IKF equations leads to an undesirable effect known as \textit{wrapping}, and will be addressed in the next chapter.

5.4 Summary

This chapter was dedicated to detailing the IKF algorithm after the main limitation of the KF was exposed, i.e. its high reliability on accurate system modelling, which in practice is not fully attainable. The IKF was applied to the heading estimation of the \textit{Springer} USV, but in doing so, a major implementation difficulty was exposed: the rapid growth of the IKF bounds due to the overly conservative nature of IA.

The underlying cause of the conservatism of IA was analysed, and a pragmatic solution based on intersecting various solution sets was attempted. Though this worked in reducing the overestimation in one case, it was found to make little difference when applied to another, and this was postulated to be because of an effect known as \textit{wrapping} that occurs when vectors are subject to iterative interval transformations. Addressing this problem is crucial to successful implementation of the IKF, and is the topic of the next chapter.
Chapter 6

The Wrapping Effect

"Every limit is a beginning as well as an ending."  — George Eliot

In the previous chapter it was shown how modelling uncertainty is a key limitation to the applicability of the classical KF for state estimation of dynamic systems. For such systems with bounded modelling uncertainty, the IKF is a direct extension of the former to interval systems. However, its usage is not yet widespread owing to the over-conservatism of the resulting interval bounds. The main reason for this is that the successive transformation of interval vectors undergoes an effect known as wrapping, owing to the fact that interval sets are represented as hyper-rectangles. In this chapter, the IKF equations are adapted to use an alternative ellipsoidal arithmetic that, in some cases, provides tighter bounds than direct, rectangular-based interval arithmetic. However, in order for the IKF to be useful, it must be able to provide reasonable enclosures under all circumstances. To this end, a hybrid ellipsoidal-rectangular interval arithmetic enclosure algorithm is proposed, and its robustness is evidenced by its application to two characteristically different systems for which it provides stable estimate bounds, whereas the rectangular and ellipsoidal approaches fail to accomplish this in either one or the other case.

6.1 The Wrapping Effect

Consider again the nominal model of the Springer USV described in Section 5.1, along with the gyroscope measurement equation:

\[ x(k+1) = A \ x(k) + B \ u(k) + \omega(k) \]  

(6.1)
\[ z(k) = C x(k) + v(k) \]  \hspace{1cm} (6.2)

\[
A = \begin{bmatrix} 0.8 & -0.2225 \\ 1 & 0 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, C = [0 \ 0.004225], \omega \sim N(0, Q),
\]

\[ Q = \text{cov}(\omega) = 10^{-4} \times \text{diag}(1,1), v(k) \sim N(0, R), R = 0.05^2 (\text{deg/s})^2, \]  \hspace{1cm} (6.3)

\[ T_s = 1 \text{ s} \]

where \( x(k) \) is the system state at time \( k \), \( u(k) \) is the controllable system input (differential propeller speed in rpm), \( \omega(k) \) is a random input disturbance, and \( z(k) \) represents the gyroscope measurement of the rate of change of the vehicle heading, \( v(k) \) being the measurement noise, and where it assumed that the random variables \{\( x(0), \omega(0),...\omega(k),v(0),..., v(k) \)\} are mutually independent.

Consider a set-point tracking problem in which the system is controlled via a state-feedback control law according to

\[ u(k) = -K_c x(k) + K_s y_r(k) \]  \hspace{1cm} (6.4)

where \( K_c \) is the vector of state-feedback gains, \( K_s \) is a scaling gain calculated to ensure that the overall steady-state closed-loop gain is unity, and \( y_r \) is a prescribed reference target for the system output. Let \( K_c \) be chosen so that the closed-loop step-response has zero overshoot (or unit damping), and a rise-time (0 to 90%) of \( t_r < 5 \text{ s} \), for which a dominant closed-loop pole at 0.6 and a fast pole at 0.1 suffice, requiring a value of \( K_c = [0.1204, -0.1530] \) and \( K_s \approx 78 \) (note that this control law is used by way of example to generate a realistic stabilising input \( u(k) \), but for the sole purpose of state-estimation, any other input could be prescribed instead).

A simulation of the system is carried out for one hundred time-steps, with the initial state being zero, and the target \( y_r \) being set at 0.2, 0.5 and -0.15 deg/s for each one-third of the simulation respectively. The disturbance and measurement noise sequences are generated pseudo randomly according to the statistics given in Equation 6.3. The state trajectory of the system is then shown in Figure 6.1, and the output in Figure 6.2. Also shown in the two figures are the KF estimates of the system state and output heading, for the same control input, disturbance input, and measurement noise sequences, in which the model parameters used by the filter have been overestimated by 5%. The KF initial estimate was taken equal to the true initial state, and the initial error
covariance as zero. As in the previous section, it is observed that the KF estimates are biased due to the modelling error.

![Figure 6.1 State trajectory and KF state trajectory estimate.](image1)

In addition, recall the interval model of the previous section, centred around the nominal model, with interval widths of 5% of the nominal values on either side of these,

![Figure 6.2. Measured output $z(k)$, noise-less output $C x(k)$, and KF output estimate.](image2)
\[ x(k + 1) = A^\mathbb{I} x(k) + B^\mathbb{I} u(k) + \omega(k) \quad (6.5) \]

\[ z(k) = C^\mathbb{I} x(k) + v(k) \quad (6.6) \]

\[ \begin{bmatrix} 0.95 \times 0.8, 1.05 \times 0.8 \\ 0.95 \times 1, 1.05 \times 1 \end{bmatrix} \quad [ -1.05 \times 0.2225, -0.95 \times 0.2225] \]

\[ B^\mathbb{I} = \begin{bmatrix} [0.95 \times 1, 1.05 \times 1] \\ [0, 0] \end{bmatrix} \]

\[ C^\mathbb{I} = [[0, 0], [0.95 \times 0.004225, 1.05 \times 0.004225]] \quad (6.7) \]

Based on the same inputs \( u(k) \) used in the previous simulation, and an initial state vector and error covariance given by

\[ \hat{x}^\mathbb{I}(0) = \begin{bmatrix} [−0.005, 0.005] \\ [−0.005, 0.005] \end{bmatrix}; \quad P^\mathbb{I}(0) = \begin{bmatrix} [0, 0] \\ [0, 0] \end{bmatrix} \quad (6.8) \]

the upper and lower bounds of the IKF estimate of the output, \( \hat{y}^\mathbb{I}(k) = C^\mathbb{I} \hat{x}^\mathbb{I}(k) \), are shown in Figure 6.3, along with the actual measured output \( z(k) \).

As established in the preceding chapter, the widths of the IKF interval-estimates expand exponentially, leading to bounds of the estimate that lose any practical value. Moreover the figure only depicts the simulation of the IKF until
$kT_s = 80\ s$; in fact, estimates of the bounds for $kT_s = 96$ and above were not possible as they surpassed the maximum and minimum IEEE 754 double precision floating point representation ($\approx 1.8 \times 10^{308}$).

The underlying reason that direct or naive IA computation yields over-conservative results is a consequence of the inclusion property described in Section 5.3. In the example given therein (Equation 5.28), the cause of the over-estimation stems from the fact that after expansion of the initial brackets, the arithmetic does not retain the knowledge that the interval-valued variable in both summands represent the same variable. In the case of operations including interval vectors, the memory-less nature of IA has an even more severe effect.

Consider an interval vector $x^I = \begin{bmatrix} [-3,3] \\ [-1,1] \end{bmatrix}$, which can be represented as a rectangle (Figure 6.4), and the rotation matrix $A = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}$, $\alpha = 20 \times \frac{\pi}{180}$. Then $x^I := A x^I$ represents the anticlockwise rotation of this rectangle by the angle $\alpha$ (20°). Upon performing this operation using IA, the resultant enclosure is not the rotated rectangle, but a rectangle with sides parallel to the coordinate axes that encloses the former, which is consistent with the memory-less nature of IA described earlier, since the correlation between the two dimensions given by the set of individually rotated points is lost. The visual interpretation of this consequence has resulted in its being known as the wrapping effect (Neumaier, 1993). The left column of Figure 6.4 depicts this process, first for a single rotation, and then ten successive applications of the same, evidencing the ever-growing volume of the resulting rectangle; whereas the actual transformed set is always the same size, given the volume preserving nature of $A$, which is an orthogonal matrix. This process clearly shows that direct, or rectangular IA, can be overly-conservative. However, as shown in the right column of Figure 6.4, if the initial rectangle is first enclosed by an ellipse, then each successive rotation of the ellipse does not need to be “wrapped” before a successive rotation is applied, since ellipses are preserved under rotations, and in general, ellipsoids under affine transformations. Indeed, let $X$ be the set of points of the ellipsoid $E(z, L, r) := \{z + L \xi : \xi \in \mathbb{R}^n, \|\xi\| \leq r\}$, $z \in \mathbb{R}^n, L \in \mathbb{R}^{n \times n}, r \in \mathbb{R}_+$, then, the set of points transformed by the affinity $x := Ax + b$, $A \in \mathbb{R}^{n \times n}, x, b \in \mathbb{R}^n$, is given by the ellipsoid $E(\tilde{z}, \tilde{L}, r)$, with $\tilde{z} = Az + b$ and $\tilde{L} = AL$. 

88
6.2. A Hybrid Ellipsoidal-Rectangular IA
Enclosure Algorithm for Recursive Interval
Affine Transformations

In the previous section, the advantage of transforming sets by ellipsoidal
arithmetic rather than rectangular IA to avoid the wrapping effect was made
apparent. However, ellipsoids are only conserved under affine transformations,
and not interval transformations. In the case of an interval affine
transformation, \( x := A^I x + b^I \), one must find a (tight) ellipsoidal enclosure to
the set of transformed points. Neumaier (1993) has developed a method that
guarantees such an enclosure with a certain degree of optimality. The main
results of this method are summarised next.

Let \( X \) be the set of points of the ellipsoid \( E(z, L, r) := \{ z + L \xi : \xi \in \mathbb{R}^n, \| \xi \| \leq r \} \), \( z \in \mathbb{R}^n, L \in \mathbb{R}^{n \times n}, r \in \mathbb{R}^+ \), where \( \| \cdot \| \) represents the Euclidean norm, and \( x := A x + b, A \in A^I, b \in b^I \) an interval affine transformation of \( x \in \mathbb{R}^n \), with \( A^I \) and \( b^I \) an \( n \times n \) and
\( n \times 1 \) interval-valued matrix and vector respectively. Neumaier shows that for any
\[ z \in \mathbb{R}^n, L \in \mathbb{R}^{n \times n}, \quad \text{if} \quad \tilde{r} = \|L^{-1}(A^t z + b^t - z)\| + \|L^{-1}A^t\| r, \quad \text{then} \quad A x + b \in E(\tilde{z}, \tilde{L}, \tilde{r}) \forall A \in A^t, b \in b^t, x \in X. \] 
This allows one to find an (ellipsoidal) enclosure for the interval affine transformation of \( X \), but such enclosure may by no means be “tight”.

However, let \( \tilde{z} = \text{mid}(A^t z + b^t), d = |A^t z + b^t - z|, B = \text{mid}(A^t L), \) and \( d' = \nu(A^t L - B) \), where \( \nu \) is the Frobenius norm, i.e., \( \nu(A) = (\sum \sum |a_{i,j}|^2)^{1/2} \). Then, for any non-singular \( \tilde{L} \) and any non-singular diagonal matrix \( D \), it can be verified that \( \tilde{r} \leq \|\tilde{L}^{-1}B\| r + \|\tilde{L}^{-1}D\| q \), where \( q = \|D^{-1}d\| + \|D^{-1}d'\| r \). In particular, for an \( \tilde{L} \) that satisfies \( r^2 B B^T + q^2 D D^T = \tilde{L} \tilde{L}^T \), then it is shown that \( \tilde{r} = \|\tilde{L}^{-1}B\| r + \|\tilde{L}^{-1}D\| q \leq 2 \) and that \( |\text{det} \tilde{L}| \leq |\text{det} \tilde{L}| \left(\|\tilde{L}^{-1}B\| r + \|\tilde{L}^{-1}D\| q \right)^n \) for any non-singular \( \tilde{L} \), that is, that this choice of \( \tilde{L} \) provides an ellipsoid enclosure that is optimal to within a factor of 2 of the radius (note that \( |\text{det} \tilde{L}| \left(\|\tilde{L}^{-1}B\| r + \|\tilde{L}^{-1}D\| q \right)^n \) is the volume of the ellipsoid for the arbitrarily chosen \( \tilde{L} \)). However, it is also noted that the optimality of the radius chosen is subject to that of the bound \( \tilde{r} \leq \|\tilde{L}^{-1}B\| r + \|\tilde{L}^{-1}D\| q \).

The choice of \( \tilde{L} \) and \( D \) still remains non-uniquely determined, and for reasons of computational stability, \( \tilde{L} \) is obtained as a Cholesky factor of \( r^2 B B^T + q^2 D D^T \) and \( D = \text{diag}\{d_i + d'_i r\} \).

In summary, this procedure provides an ellipsoidal enclosure for the image-set of the points belonging to an initial ellipsoid by an interval affine transformation. In practice, to use this method for propagating the state vector of a dynamic system given an initial interval vector, which is representable by an \( n \) dimensional hyper rectangle, or box, an ellipsoidal enclosure to this initial state is first calculated. Although the ellipsoid circumscribing a given box is not unique, one option is to choose the minimal volume circumscribing ellipsoid.

Consider the simulation of the interval system given by Equations (6.5) to (6.7), with the same inputs \( u(k) \) used in the previous simulation, and with an initial interval-valued state vector given by
Figure 6.5a depicts the propagation of the state intervals using rectangular IA, whereas Figure 6.5b shows the same calculations but using ellipsoidal arithmetic with Neumaier’s enclosures. Clearly the elliptical enclosures are far tighter than the corresponding rectangular ones. Note that the interval affine transformation of the state vector at each time-step is given by Equation 6.5, in which $A^I$ represents a linear interval transformation (i.e., set of linear transformations), and $b^I(k) \equiv B^I u(k) + \omega(k)$ an interval translation vector (i.e. set of translations). The eigenvalues of the nominal linear transformation, $A$, are the complex conjugate pair $0.4 \pm 0.25i$, so the system is somewhat underdamped.

Now consider the simulation of an interval system with nominal overdamped dynamics given by

$$A = \begin{bmatrix} 0.6 & -0.05 \\ 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 0.0045 \end{bmatrix}$$

and interval matrices centred around these nominal ones with interval widths of 5% on either side. As before, simulation is carried out for one-hundred time-steps, with a reference target $y_r$ being set at 0.2, 0.5 and -0.15 deg/s for each one-third of the simulation respectively, and a state-feedback control law as in Equation 6.4 with $K_c = [-0.1, 0.01]$ and $K_s = 80$, calculated to have the same closed-loop poles as before (0.6 and 0.1) (note that for generating the control input, the true state of the system is used, as the main concern here is not the control law generated but the calculation of the interval state trajectory). The results using direct (rectangular) IA and ellipsoidal arithmetic are shown in Figures 6.6a and 6.6b respectively, with the initial state being described by Equation 6.9.
(a) Simulation of the state-vector for the interval system with nominal poles at $0.4 \pm 0.25i$ using rectangular IA.

(b) Simulation of the state-vector for the interval system with nominal poles at $0.4 \pm 0.25i$ using ellipsoidal arithmetic.

Figure 6.5 Simulation of the state-vector for the interval system with nominal poles at $0.4 \pm 0.25i$ using (a) rectangular IA and (b) ellipsoidal arithmetic.
(a) Simulation of the state-vector for the interval system with nominal poles at 0.5 and 0.1 using rectangular IA.

(b) Simulation of the state-vector for the interval system with nominal poles at 0.5 and 0.1 using ellipsoidal arithmetic.

Figure 6.6 Simulation of the state-vector for the interval system with nominal poles at 0.5 and 0.1 using (a) rectangular IA and (b) ellipsoidal arithmetic.
In this case, rectangular IA provides tighter bounds to the state-vector sets than does the ellipsoidal arithmetic. It is interesting to note that in this case, the nominal poles of the system are at 0.5 and 0.1.

The tightness of the ellipsoidal enclosures computed according to Neumaier’s algorithm depends on the nature of the linear interval transformation matrix $A^I$. If the eigenvalues are predominantly complex conjugate (corresponding to underdamped systems), then the transformation has a greater rotation component, and Neumaier’s enclosures work well. On the other hand, for real eigenvalues (corresponding to overdamped systems), the transformation has a larger shear component, and the ellipsoid enclosures become very elongated in the direction of largest stretching. Thus it becomes a necessity to develop a method that can be of use in either situation.

In order to provide enclosures that are tight for both under and overdamped systems, the following algorithm is proposed for recursive affine transformations. Given an initial interval vector, which is an $n$-dimensional box, $B_0$, obtain the smallest ellipsoid containing it, $E_0$. The word “smallest” is used here vaguely, but one option (and the one implemented here) is to choose the ellipsoid with the smallest volume that contains the box, that is, an ellipsoid with the same eccentricity as the box. Then proceed to apply the interval affine transformation to both enclosures, $B_0$ via rectangular IA and $E_0$ via ellipsoidal arithmetic, resulting in $B_1$ and $E_1$, respectively. Note that the tightest set containing the transformed points at this stage will be given by the intersection of $B_1$ and $E_1$, which however is neither a box nor an ellipsoid. Thus, obtain the tightest box $B_2$ that contains $E_1$, and intersect these to obtain $B_3$. The box $B_3$ is now the smallest box-enclosure that contains the transformed set, and is used as the starting box for the next interval transformation. On the other hand, obtain $E_3$ as the smallest ellipsoid containing $B_3$. Then select the smallest of $E_1$ and $E_3$ as the starting ellipsoid to be transformed by the next interval affine transformation. Again, smallest in this case could mean smallest in volume, sum of semi-axes, etc. The algorithm is summarised in Figure 6.7.
6.3 The IKF Equations as Affine Transformations

The ellipsoidal arithmetic approach is possible because ellipsoids are invariant under affine transformations. In order to be able to apply this method to the IKF equations, these must first be expressed as recursive affine transformations. The IKF formulation was given by Equations 5.17 to 5.21 and can be divided into three sets: propagation of the state-vector, propagation of the error covariance, and calculation of the Kalman gain.

With respect to propagation of the state vector, Equations 5.17 and 5.20 must be applied in turn. The prediction equation 5.17 is already in the form of an interval affine transformation, whereas Equation 5.20 can be written as

\[ \hat{x}^I (k|k) = \{ I - K^I (k) C^I \} \hat{x}^I (k|k - 1) + K^I (k) z(k) \]  

(6.11)
which, if \( K^i(k) \) has already been obtained, is also clearly an interval affine transformation.

For the error covariance estimates, consider the prediction equation 5.18 written using indicial notation:

\[
p_{i,j}^1(k|k-1) = a_{i,l}^1 a_{j,m}^1 p_{l,m}^1(k-1|k-1) + q_{i,j} ; \quad i,j,l,m = 1,\ldots, n \quad (6.12)
\]

In order to represent the state estimate error and disturbance covariance matrices as vectors, consider \( \tilde{p}^1(k|k-1), \tilde{p}^1(k-1|k-1) \), and \( \tilde{q} \in \mathbb{R}^{n^2} \) defined as:

\[
\tilde{p}_{n(i-1)+j}^1(k|k-1) \triangleq p_{i,j}^1(k|k-1) \ , \ i,j = 1,\ldots, n \quad (6.13)
\]

\[
\tilde{p}_{n(i-1)+j}^1(k-1|k-1) \triangleq p_{i,j}^1(k-1|k-1) \ , \ i,j = 1,\ldots, n \quad (6.14)
\]

\[
\tilde{q}_{n(i-1)+j} \triangleq q_{i,j} \ , \ i,j = 1,\ldots, n \quad (6.15)
\]

Then by Equation 6.12, the relationship between these is

\[
\tilde{p}_{n(i-1)+j}^1(k|k-1) = p_{i,j}^1(k|k-1) = a_{i,l}^1 a_{j,m}^1 p_{l,m}^1(k-1|k-1) + q_{i,j} = a_{i,l}^1 a_{j,m}^1 \tilde{p}_{n(i-1)+m}^1(k-1|k-1) + \tilde{a}_{n(i-1)+j} \quad (6.16)
\]

Next, define

\[
(\tilde{a}_{i,j}) \in \mathbb{R}^{n^2 \times n^2} : \quad \tilde{a}_{n(i-1)+j,n(l-1)+m} \triangleq a_{i,l}^1 a_{j,m}^1 , \ i,j,l,m = 1,\ldots, n \quad (6.17)
\]

and also

\[
r \triangleq n(i-1) + j, \quad s \triangleq n(l-1) + m \quad (6.18)
\]

from which

\[
j = r - n(i-1), \quad m = s - n(l-1) \quad (6.19)
\]

Thus,

\[
\tilde{a}_{r,s}^1 = a_{i,l}^1 a_{r-n(i-1),s-n(l-1)} \quad \text{for} \quad i,l \in \{1,\ldots,n\} \quad \text{and} \quad r - n(i-1),

s - n(l-1) \in \{1,\ldots,n\} \quad (6.20)
\]
that is,

\[
\tilde{a}_{r,s}^1 = a_{i,l}^1 \ a_{r-n(i-1),s-n(l-1)}^{1} \quad \text{for} \ i, l \in \{1, \ldots, n\},
\]

\[
r \in \{1 + n(i - 1), 2 + n(i - 1), \ldots, n + n(i - 1)\},
\]

\[
s \in \{1 + n(l - 1), 2 + n(l - 1), \ldots, n + n(l - 1)\}
\]

(6.21)

With these definitions, Equation 6.16 is expressed as an affine transformation:

\[
\bar{p}_{r}^1(k|k - 1) = \tilde{a}_{r,s}^1 \ ar{p}_{s}^1(k - 1|k - 1) + \tilde{q}_{r}; \ r, s = 1, \ldots, n^2
\]

(6.22)

Similarly, the correction equation 5.21 in indicial notation is

\[
p_{i,j}^1(k|k) = \{1 - K^1(k) \ C^1\}_{i,l} \ p_{i,j}^1(k|k - 1), \ i, j, l = 1, \ldots, n
\]

(6.23)

As before, define

\[
\bar{p}_{i}^1(k|k) \in \mathbb{R}^{n^2}: \ \bar{p}_{n(i-1)+j}^1(k|k) \equiv p_{i,j}^1(k|k), \ i, j = 1, \ldots, n
\]

(6.24)

and with \(\bar{p}_{i}^1(k|k - 1)\) defined as before, Equation 6.23 can be written as

\[
\bar{p}_{n(i-1)+j}^1(k|k) = p_{i,j}^1(k|k) = \{1 - K^1(k) \ C^1\}_{i,l} \ p_{i,j}^1(k|k - 1) = \{1 - K^1(k) \ C^1\}_{i,l} \ \tilde{p}_{n(l-1)+j}^1(k|k - 1)
\]

(6.25)

Next, define

\[
(\tilde{h}_{i,j}^1) \in \mathbb{R}^{n^2 \times n^2}:
\]

\[
\tilde{h}_{n(i-1)+j,n(l-1)+j}^1 \equiv \begin{cases} \{1 - K^1(k) \ C^1\}_{i,l} & \text{for} \ i, l = 1, \ldots, n, j = 1, \ldots, n \\ 0 & \text{otherwise} \end{cases}
\]

(6.26)

and also

\[
r \equiv n(i - 1) + j, \quad s \equiv n(l - 1) + j
\]

(6.27)

from which

\[
i = \frac{r-j}{2} + 1, \quad l = \frac{s-j}{2} + 1
\]

(6.28)
Thus,

\[ \tilde{h}^1_{r,s} \equiv \begin{cases} \left\{ I - K^1(k)C^1 \right\}_{r-j+1/2}^{s-j+1/2} & \text{if } \frac{r-j}{2} + 1, \frac{s-j}{2} + 1 \in \{1, \ldots, n\}, \\ 0 & \text{otherwise} \end{cases} \]

\[ j = 1, \ldots, n; \ r, s = 1, \ldots, n^2 \] (6.29)

i.e.,

\[ \tilde{h}^1_{r,s} \equiv \begin{cases} \left\{ I - K^1(k)C^1 \right\}_{r-j+1/2}^{s-j+1/2} & \text{if } r, s \in \{j, j + 2, \ldots, j + n(n-1)\}, \\ 0 & \text{otherwise} \end{cases} \]

\[ j = 1, \ldots, n; \ r, s = 1, \ldots, n^2 \] (6.30)

Then, Equation 6.25 can be written as

\[ \tilde{p}^1_r(k|k) = \tilde{h}^1_{r,s} \tilde{p}^1_s(k|k-1); \ r, s = 1, \ldots, n^2 \] (6.31)

which is in the form of an affine transformation.

For example, for \( n = 2 \), the error covariance prediction equation is

\[
\begin{bmatrix}
    p^1_{1,1}(k|k-1) & p^1_{1,2}(k|k-1) \\
    p^1_{2,1}(k|k-1) & p^1_{2,2}(k|k-1)
\end{bmatrix} =
\begin{bmatrix}
    a^1_{1,1} & a^1_{1,2} \\
    a^1_{2,1} & a^1_{2,2}
\end{bmatrix}
\begin{bmatrix}
    p^1_{1,1}(k-1|k-1) & p^1_{1,2}(k-1|k-1) \\
    p^1_{2,1}(k-1|k-1) & p^1_{2,2}(k-1|k-1)
\end{bmatrix}
\begin{bmatrix}
    a^1_{1,1} & a^1_{1,2} \\
    a^1_{2,1} & a^1_{2,2}
\end{bmatrix}
+ \begin{bmatrix}
    q_{1,1} & q_{1,2} \\
    q_{2,1} & q_{2,2}
\end{bmatrix}
\] (6.32)

which can be expressed as

\[
\begin{bmatrix}
    \tilde{p}^1_1(k|k-1) \\
    \tilde{p}^1_2(k|k-1) \\
    \tilde{p}^1_3(k|k-1) \\
    \tilde{p}^1_4(k|k-1)
\end{bmatrix} =
\begin{bmatrix}
    a^1_{1,1} & a^1_{1,2} & a^1_{1,1} & a^1_{1,2} \\
    a^1_{2,1} & a^1_{2,2} & a^1_{2,1} & a^1_{2,2} \\
    a^1_{1,1} & a^1_{1,2} & a^1_{1,1} & a^1_{1,2} \\
    a^1_{2,1} & a^1_{2,2} & a^1_{2,1} & a^1_{2,2}
\end{bmatrix}
\begin{bmatrix}
    \tilde{p}^1_1(k-1|k-1) \\
    \tilde{p}^1_2(k-1|k-1) \\
    \tilde{p}^1_3(k-1|k-1) \\
    \tilde{p}^1_4(k-1|k-1)
\end{bmatrix}
\begin{bmatrix}
    a^1_{1,1} & a^1_{1,2} \\
    a^1_{2,1} & a^1_{2,2}
\end{bmatrix}
+ \begin{bmatrix}
    \tilde{q}_1 \\
    \tilde{q}_2 \\
    \tilde{q}_3 \\
    \tilde{q}_4
\end{bmatrix}
\] (6.33)
where

\[
\begin{bmatrix}
\hat{p}_1^1(k|k-1) \\
\hat{p}_2^1(k|k-1) \\
\hat{p}_3^1(k|k-1) \\
\hat{p}_4^1(k|k-1)
\end{bmatrix} = 
\begin{bmatrix}
p_{1,1}^1(k|k-1) \\
p_{1,2}^1(k|k-1) \\
p_{2,1}^1(k|k-1) \\
p_{2,2}^1(k|k-1)
\end{bmatrix}.
\]

Similarly, the error covariance correction equation is

\[
\begin{bmatrix}
p_{1,1}^1(k|k) & p_{1,2}^1(k|k) \\
p_{2,1}^1(k|k) & p_{2,2}^1(k|k)
\end{bmatrix} = 
\begin{bmatrix}
h_{1,1}^1 & h_{1,2}^1 \\
h_{2,1}^1 & h_{2,2}^1
\end{bmatrix}
\begin{bmatrix}
p_{1,1}^1(k|k-1) & p_{1,2}^1(k|k-1) \\
p_{2,1}^1(k|k-1) & p_{2,2}^1(k|k-1)
\end{bmatrix}
\]

which is equivalent to

\[
\begin{bmatrix}
\hat{p}_1^1(k|k) \\
\hat{p}_2^1(k|k) \\
\hat{p}_3^1(k|k) \\
\hat{p}_4^1(k|k)
\end{bmatrix} = 
\begin{bmatrix}
h_{1,1}^1 & 0 & h_{1,2}^1 & 0 \\
0 & h_{1,1}^1 & 0 & h_{1,2}^1 \\
h_{2,1}^1 & 0 & h_{2,2}^1 & 0 \\
0 & h_{2,1}^1 & 0 & h_{2,2}^1
\end{bmatrix}
\begin{bmatrix}
\hat{p}_1^1(k|k-1) \\
\hat{p}_2^1(k|k-1) \\
\hat{p}_3^1(k|k-1) \\
\hat{p}_4^1(k|k-1)
\end{bmatrix}
\]

if

\[
\begin{bmatrix}
\hat{p}_1^1(k|k) \\
\hat{p}_2^1(k|k) \\
\hat{p}_3^1(k|k) \\
\hat{p}_4^1(k|k)
\end{bmatrix} = 
\begin{bmatrix}
p_{1,1}^1(k|k) \\
p_{1,2}^1(k|k) \\
p_{2,1}^1(k|k) \\
p_{2,2}^1(k|k)
\end{bmatrix}
\]

Finally, the Kalman gain is updated at each step according to Equation 5.19 using direct (rectangular) IA.

### 6.4 Interval Kalman Filtering using the Hybrid Ellipsoidal-Rectangular IA Enclosure Algorithm

Consider again the simulation of the two interval systems of the preceding section, namely, the systems described by Equations 6.5 and 6.6 with interval
matrices \( A^I, B^I, \) and \( C^I \) centred around the point valued matrices given, in the first case, by Equation 6.3 (underdamped nominal dynamics), and in the second, by Equation 6.10 (overdamped nominal dynamics), with interval widths equal to 5\% of the nominal values on either side. Using the same input sequence \( \{u(k)\} \) used in the preceding sections, and the same disturbance and noise processes described earlier, the state-trajectory and output of the nominal systems are simulated from an initial state situated at the origin of the state-space. Also, based on the initial estimates given in Equation 6.8, a rectangular-IA IKF, an ellipsoidal-arithmetic IKF, and a hybrid-enclosure IKF are simulated in each case to estimate the interval system’s state vector and output. These simulations are depicted in Figures 6.8 to 6.11.

Figures 6.8 and 6.9 present the results for the underdamped system. Figure 6.8a depicts the evolution of the state vector of the nominal system, in which each circle represents the point on the state plane that describes the state of the system at each time-step. Estimates of these states of the interval system by the rectangular IA IKF are shown in Figure 6.8b, and are given by rectangular enclosures. Clearly, these rectangles increase rapidly in size, retaining little association with the states they represent. Figure 6.8c depicts the state estimates of the interval system given by the ellipsoidal arithmetic IKF, described by ellipsoidal enclosures. Although rectangles that enclose these ellipses are shown in the figure as well, it is only the elliptical sets that are propagated from one iteration to the next. The figure shows that in this case the enclosures do follow the states of the nominal system, and that the overestimation does not grow in an unstable manner. Lastly, Figure 6.8d depicts the evolution of the hybrid-enclosure estimates of the interval system states. In this case, they follow those of the ellipsoidal arithmetic IKF.

These results are summarised in Figure 6.9, which depicts the nominal system output and the bounds output estimates of the interval system given by each of the three filters. Again, it is seen how the bounds given by the ellipsoidal arithmetic IKF remain stable, whereas those of the rectangular IKF diverge.
(a) Simulation of the state-vector for the nominal underdamped system with poles at $0.4 \pm 0.25i$. States are depicted as circles for clarity, although they represent point-values.

(b) Rectangular IA IKF estimates of the interval system state. Only the first 20 iterations shown, as the rectangles keep growing exponentially. The arrow follows the initial propagation of rectangular state enclosures.
(c) Ellipsoidal arithmetic IKF estimates of the interval system state. The dotted rectangles correspond to the smallest box enclosure of each ellipse, but are not used for propagation.

(d) Hybrid IA IKF estimates of the interval system state. The dotted rectangles correspond to the box $B_3$ at each iteration.

Figure 6.8 Simulation of the nominal underdamped system with poles at $0.4 \pm 0.25i$: (a) actual state vector; (b) IKF state estimates using rectangular IA; (c) IKF state estimates based on ellipsoidal arithmetic; (d) IKF estimates using the hybrid ellipsoidal-IA enclosure algorithm.
Figure 6.9 Output of the nominal underdamped system, and IKF bounds to the estimates of the output of the corresponding interval system using rectangular, ellipsoidal, and hybrid IA respectively.

Figures 6.10 and 6.11 present the corresponding results for the overdamped system. Figure 6.10a shows the evolution of the nominal system state – again, these are points on the state plane represented by small circles. Figure 6.10b shows the state estimates of the interval system by the rectangular IKF. In this case, the rectangles enclose the true states without expanding indefinitely over time. However, Figure 6.10c, which depicts the state estimates by the ellipsoidal arithmetic filter, shows that the elliptical bounds become excessively elongated, yielding enclosures that are not representative of the true state sets. The hybrid enclosures, shown in Figure 6.10d, in this case conform mostly to the rectangular enclosures.

Figure 6.11 compares the nominal system output with the estimate bounds of the interval system output by each of the three filters, reflecting the results seen in the previous figure.
(a) Simulation of the state-vector for the nominal overdamped system with poles at 0.5 and 0.1. States are depicted as circles for clarity, although they represent point-values.

(b) Rectangular IA IKF estimates of the interval system state. The arrows indicate the direction of propagation of rectangular state enclosures.
Figure 6.10 Simulation of the state-vector for the nominal overdamped system with poles at 0.5 and 0.1: (a) actual state vector; (b) IKF state estimates using rectangular IA; (c) IKF state estimates based on ellipsoidal arithmetic; (d) IKF estimates using the hybrid ellipsoidal-IA enclosure algorithm.
In summary, in the case of the underdamped system with complex conjugate poles, the rectangular IA IKF overestimates the state enclosures to such an extent that simulation cannot continue after a certain point, whereas the ellipsoidal arithmetic IKF provides much tighter bounds. The hybrid-enclosure IKF in this case offers bounds similar to those of the ellipsoidal arithmetic IKF. In the case of the overdamped system with real poles, however, it is the rectangular IA IKF enclosures that provide the tighter bounds to the sets of state vectors, and so the hybrid-enclosure IKF relies mostly on these.

In practice, the dynamics of a system can be prone to large variations, and control design based on a multiple model representation is a commonly used approach. For such systems, even if these nominal large-scale dynamical changes are known, obtaining the IKF state estimates requires a computational algorithm that is operationally robust to such changes. Hence it is advantageous to implement the IKF using the hybrid-enclosure method rather than with direct IA.
It should also be noted that if the closed-loop dynamics of the system are specified, such as in the examples shown here, then the IKF may operate directly on the resulting closed loop interval model obtained from applying the feedback control law to the uncertain open-loop model. Concretely, if the nominal steering model of the vehicle is given by the underdamped dynamics of Equations 6.1 to 6.3, then it can easily be verified that a state feedback gain of \( K_c = [0.200, -0.1725] \) would result in the closed-loop dynamics specified by Equation 6.10. Hence if this is the control law used for piloting the vehicle, the IKF state estimation could be based directly upon the uncertain system centred around the nominal closed-loop model, with the (controlled) inputs to said model now being the desired or reference rate of change of heading of the vehicle rather than the differential motor speed. In this sense, the hybrid IA enclosure IKF also provides robustness to the choice of desired closed-loop specifications, which could otherwise trigger an undue overestimation of the state enclosures.

### 6.5 Summary

In this chapter the wrapping effect that causes the overestimation when computing the IKF bounds was analysed. It was seen that for the successive transformation of a state vector by a pure rotation matrix, this effect could be avoided completely by using an ellipsoidal arithmetic in lieu of the standard rectangular IA. For general interval affine transformations the guaranteed ellipsoidal enclosure of the image set developed by Neumaier is used instead. Because the effectiveness, or tightness, of these ellipsoidal enclosures depends on the particular transformation, a hybrid enclosure algorithm was proposed, based on using ellipsoidal arithmetic and rectangular IA to propagate an ellipsoidal and a rectangular bound of the state vector at each time step, respectively, and then obtain fused versions of these before the next iteration. The fusions only require intersecting boxes and circumscribing these by ellipsoids, and are computationally inexpensive.

In order to apply this to interval Kalman filtering, the IKF formulation was then adapted so that it could be described as recursive interval affine transformations. The case studies presented in this chapter have shown that the hybrid IA enclosure algorithm for the IKF provides stable bounded estimates for a wider range of system dynamic characteristics than does the use of direct, rectangular IA for computation of the IKF equations.
Chapter 7

Weighted Interval Kalman Filtering

“If you torture the data long enough, it will confess.” — Ronald Coase

Thus far it has been shown how, for uncertain systems, the IKF provides guaranteed bounds to the optimal KF estimates, or KF estimates that would be obtained if a completely accurate model of the system dynamics were available. In practice however a single point-valued estimate is often needed instead: in the case of the Springer USV, the autopilot requires a single estimate of the heading at each time step. This chapter is dedicated to the adequate selection of individual values from the interval estimates.

7.1 The Weighted IKF

The Springer steering model and gyroscope measurement equation used in the preceding chapters are given by

\[ x(k + 1) = A \, x(k) + B \, u(k) + \omega(k) \quad (7.1) \]

\[ y(k) = C \, x(k) \quad (7.2) \]

\[ z(k) = y(k) + v(k) \quad (7.3) \]

\[ A = \begin{bmatrix} 0.8 & -0.2225 \\ 1 & 0 \end{bmatrix}, \, B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \, C = \begin{bmatrix} 0 & 0.004225 \end{bmatrix}, \, \omega \sim N(0, Q), \]
\[ Q = \text{cov}(\omega) = 10^{-4} \times \text{diag}\{1,1\}, \; \nu(k) \sim \mathcal{N}(0,R), R = 0.05^2 (\text{deg/s})^2, \quad (7.4) \]

\[ T_s = 1 \text{ s} \]

with \( x(k) \) being the state vector, \( u(k) \) the input representing the differential propeller speed in rpm, \( \omega(k) \) a random input disturbance, \( y(k) \) the rate of change of the vehicle heading, and \( z(k) \) the gyroscope measurement of the same, with \( \nu(k) \) the measurement noise. Assuming the usual KF hypotheses, let a KF based on this model be denoted as KF-1.

Now, let it be supposed that the modeller is uncertain of the precise values of the elements of the matrices \( A, B \) and \( C \), and in fact declares that

‘the values of the matrix coefficients could not be ascertained with complete exactitude, however, it is possible to warrant that they are not further departed from these than by an amount equating to five per cent of the same.’

Upon this revelation, it is apparent that the filter KF-1 will no longer provide an optimal estimate if the actual values of these matrices, denoted by \( A_1, B_1 \) and \( C_1 \), depart from those of \( A, B \) and \( C \), and in particular, the difference between the measurements \( z(k) \) and the model’s predicted output \( C x(k) \) will no longer have a zero mean value.

In order to account for the imprecisely modelled values, consider the following interval model centred around the nominal model (Equations 7.1 to 7.4) already encountered in the previous chapter:

\[ x(k + 1) = A^I x(k) + B^I u(k) + \omega(k) \quad (7.5) \]

\[ z(k) = C^I x(k) + \nu(k) \quad (7.6) \]

\[
A^I = A \pm 0.05|A| = \\
\begin{bmatrix}
0.95 \times 0.8 & 1.05 \times 0.8 \\
0.95 \times 1 & 1.05 \times 1
\end{bmatrix} \\
\begin{bmatrix}
-1.05 \times 0.2225 & -0.95 \times 0.2225 \\
0 & 0
\end{bmatrix},
\]

\[
B^I = B \pm 0.05|B| = \\
\begin{bmatrix}
0.95 \times 1 & 1.05 \times 1 \\
0 & 0
\end{bmatrix},
\]

\[
C^I = C \pm 0.05|C| = \\
\begin{bmatrix}
0 & 0 \times 0.004225 \\
0.95 \times 0.004225 & 1.05 \times 0.004225
\end{bmatrix} \quad (7.7)
\]
with $\omega$ and $\nu$ characterised as in Equation 7.4, and $T_s = 1\,s$. Based on this interval model, an IKF, which provides interval valued estimates, can be designed.

Let it also be supposed that the true dynamics, whilst not corresponding exactly to the values given in Equation 7.4, are contained within the interval model (Equations 7.5 to 7.7), and are given by

$$x(k + 1) = A_1 x(k) + B_1 u(k) + \omega(k) \quad (7.8)$$

$$z(k) = C_1 x(k) + \nu(k) \quad (7.9)$$

$$A_1 = A + 0.05|A| = \begin{bmatrix} 0.84 & -0.2114 \\ 1.05 & 0 \end{bmatrix},$$

$$B_1 = B + 0.05|B| = \begin{bmatrix} 1.05 \\ 0 \end{bmatrix},$$

$$C_1 = C + 0.05|C| = \begin{bmatrix} 0 & 0.0044 \end{bmatrix},$$

$$\omega \sim N(0, Q), \quad Q = \text{cov}(\omega) = 10^{-4} \times \text{diag}(1,1),$$

$$\nu(k) \sim N(0, R), \quad R = 0.05^2 (\text{deg}/s)^2, \quad T_s = 1\,s \quad (7.10)$$

Then, a KF based on this model would provide a statistically optimal estimate of the state-vector and system output. Let such a KF be denoted by KF-ideal.

Consider the following arbitrarily chosen signal as input to the system (Figure 7.1a),

$$u(k) = a \times \text{sign} [\sin(0.02\,k\,T_s)]; \quad a = \begin{cases} 
10, & 0 \leq k < 200 \\
20, & 200 \leq k < 400 \\
6.67, & 400 \leq k < 630 \\
30, & 630 \leq k < 940 \\
10, & 940 \leq k < 1250
\end{cases} \quad (7.11)$$

Then, based on simulated values of $\omega(k)$ and $\nu(k)$, the state trajectory and respective estimates of the three filters can be calculated. The estimates of the system output of each filter are shown in Figure 7.1b. The system was simulated with zero initial conditions, and the filters initialised with zero initial state estimates and zero initial error covariance matrices. It is to be noted that in all cases the measurements are simulated using the true system’s dynamics (Equations 7.8 to 7.10) and not the respective models, since they represent the actual measurements.
Several observations can now be made. First, it can be seen that the KF-1 estimate deviates from the KF-ideal estimate due to the incorrect model assumed by the former. However, both of these lie within the bounds of the IKF interval estimate, as the latter must in principle contain every single KF estimate arising from a model contained within the interval model (Chen et al., 1997). Finally, it can also be verified that the arithmetic average of the IKF bounds approximately coincides with the KF-1 estimate.

![Figure 7.1](image)

(a) Applied system input.

![Figure 7.1](image)

(b) System output estimates.

Figure 7.1 KF-ideal, IKF, and KF-1 estimates of the output of the system to the piecewise step input \( u(k) = a \times \text{sign} \{\sin(0.02 \ k T_S)\} \).
Let $y^{IKF}(k)$ be the IKF estimate of the system output. If a weight $w \in [0, 1]$ is chosen at each time step, then the weighted average of its bounds, henceforth the weighted IKF (wIKF) estimate, is given by

$$y^{wIKF}(k) = y^{IKF-}(k) + w(k)[y^{IKF+}(k) - y^{IKF-}(k)]$$

with $y^{IKF+}(k) = \max\{y^{IKF}(k)\}$ and $y^{IKF-}(k) = \min\{y^{IKF}(k)\}$

(7.12)

and lies within the boundaries of the IKF interval estimate. In addition, based upon the previous observations, there exists a particular value of $w$ at each time-step for which the wIKF estimate matches the KF-ideal estimate. (Note also that the KF-1 estimate can be computed from Equation 7.12 with $w(k) = 0.5$).

Figure 7.2a depicts, at each time step, these desired weights that produce a wIKF estimate coincident with that of KF-ideal, easily calculated from Equation 7.12 when the KF-ideal estimate is known. The key question is, can these weights be calculated in practice without the knowledge of the true system dynamics, and hence, without the availability of the KF-ideal estimate? The answer, fortunately, is yes, as is explained in what follows.

It is well established that under optimal conditions, the innovations of the KF, or difference between and a priori prediction and measured output, should be comprised of a white noise sequence (Shimkin, 2009). However, under erroneous modelling assumptions, the optimality of the KF estimate is lost, resulting in an innovations sequence that ceases to correspond to white noise. The innovations sequences of both the KF-1 and KF-ideal estimates of Figure 7.1b are shown in Figure 7.2b.

It seems likely that there should exist a deterministic relationship between the innovations sequence and the desired weighting sequence, and as such, it should be possible to model such a relationship. It is also well established that ANNs are capable of replicating complex cause-effect relationships, enabling one to predict the output of such processes for new inputs (Abdi et al, 1999).
7.2 An ANN as the Missing Link

The recurrent multi-layer perceptron (RMLP) type ANN shown in Figure 7.3 was trained using as input the innovations sequence of KF-1 and as target the desired weights (Figure 7.2). Owing to the fact that the relationship between innovations and desired weights in most likelihood depends not just on the instantaneous values but on the *trends* of the innovations as well, these trends were incorporated into the ANN model by considering six consecutive values of the innovations for each desired output, consisting of the present value as well as the previous five values: $\text{inn}(k), \text{inn}(k-1), \ldots, \text{inn}(k-5)$. Although not apparently necessary, another feature was added to the input of the network: the width of the IKF interval, $\Delta ikf(k) = y^{IKF^+}(k) - y^{IKF^-}(k)$. The addition of this extra input was seen to enhance the performance of the network, the reason for which will be discussed in a later section.
It was also observed that the use of feedback from the output also helped increase the network’s accuracy, and so five time-delayed values from the output were fed back as inputs to the network. Such a network is known as a recurrent network in the literature, and is often used to model infinite impulse response (IIR) systems. Thus the combined input to the network at time-step $k$ (not counting the bias unit) can be described as the following twelve-dimensional feature vector:

$$x(k) = \begin{bmatrix}
    x_1 \\
    x_2 \\
    \vdots \\
    x_6 \\
    x_7 \\
    \vdots \\
    x_{12}
\end{bmatrix} = \begin{bmatrix}
    \text{inn}(k) \\
    \text{inn}(k-1) \\
    \vdots \\
    \text{inn}(k-5) \\
    \Delta ikf(k) \\
    \hat{w}(k-1) \\
    \vdots \\
    \hat{w}(k-5)
\end{bmatrix}$$

(7.13)

where $\hat{w}$ is the output of the network.

Figure 7.3 RMLP used, consisting of 12 input units, 5 hidden units, and single output unit.
The training process consists of adequately altering the initially random parameters of the network \((\Theta^{(1)} \text{ and } \Theta^{(2)})\) so that the mapping defined by it fits the data. The mathematical details of the training process used are given in Appendix B. The training results are shown in Figure 7.4b. The virtue of the fit is evaluated by calculating the MSE between the predicted output \(\hat{w}\) and the desired, or target, output \(w_t\), and comparing it to the MSE between a constant weighting sequence of 0.5 (the default weighting that would be used to select a nominal value from the IKF estimate in the absence of any specific criterion) and the target \(w_t\). In this case the MSE decreases from 0.036885 for the latter to 0.004670 for the ANN prediction, a decrease of almost one order of magnitude (87.4%).

Figure 7.4 clearly shows that the trained ANN establishes a mapping between the inputs (innovations sequence of KF-1 and IKF interval width) and the desired weighting. However, it is crucial to investigate if this model generalises well to new data.

In order to test the trained ANN on new data, two new data-sets were generated from new input signals applied to the dynamic system (Equations 7.8 to 7.10), from which the KF and IKF estimates, desired IKF weighting, and KF innovation sequences were generated. These are summarised in the following figures. For the first of these test cases, Figure 7.5a depicts a different piecewise step input to the one originally used for training the ANN, given by

\[
\begin{align*}
u(k) &= a \times sign\left[\sin\left(2\pi k \frac{\tau_s}{400}\right)\right]; \\
a &= \begin{cases} 
15, & 0 \leq k < 400 \\
22.5, & 400 \leq k < 800 \\
7.5, & 800 \leq k < 1200 \\
18.5, & 1200 \leq k < 1400
\end{cases}
\end{align*}
\] (7.14)

while Figures 7.5b and 7.5c show the data-set generated from it. Also in Figure 7.5c is the predicted output of the previously trained ANN to the signals shown in Figure 7.5b. Similar graphs are shown for the second test case in Figure 7.6. In this case, an input consisting of a superposition of sinusoidal waveforms of various frequencies and amplitudes was applied to the system (Figure 7.6a). Table 7.1 summarises the test performances of the trained ANN on these new test sets.
(a) KF-1 innovations sequence and $\Delta ikf$.

(b) Comparison of desired weighting sequence and ANN output.

Figure 7.4 Comparison of desired weighting sequence and output of trained RMLP. The MSE between the desired weight and ANN output is 0.004670, compared to 0.036885 between the desired weight and a constant value of 0.5.

As can be observed, in both test cases, the MSE of the trained ANN output is considerably lower than the mean square difference between the target weighting and the constant weight that represents the arithmetic mean of the IKF boundaries.

Table 7.1 Test performances for the ANN trained on data-set shown in Figure 7.4.

<table>
<thead>
<tr>
<th>Test case 1</th>
<th>Test Case 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE 0.5</td>
<td>MSE 0.5</td>
</tr>
<tr>
<td>0.036886</td>
<td>0.005269</td>
</tr>
</tbody>
</table>
Test case 1:

(a) Alternative piecewise constant signal $u(k)$.

(b) KF-1 innovations sequence and $\Delta_{ikf}$

(c) Comparison of desired weighting sequence and ANN output.

Figure 7.5 Performance of trained ANN on test set generated from an alternative piecewise constant input signal: (a) system input; (b) sequences used to generate ANN input; (c) prediction performance.
Test case 2:

(a) $u(k) = 5 \sin(0.25k) + 10 \cos(0.15k) + 10 \sin(0.08k) - 8 \sin(0.04k) + 5 \cos(0.05k)$

(b) KF-1 innovations sequence and $\Delta ikf$

(c) Comparison of desired weighting sequence and ANN output.

Figure 7.6 Performance of trained ANN on test set generated from a input of superimposed sinusoids: (a) system input; (b) sequences used to generate ANN input; (c) prediction performance.
Despite the positive results obtained thus far, it should be noted that in practice, the real system dynamics may differ from that which was used to generate the data on which the ANN was trained. In fact, this is most likely to be the case, and one would not know the precise values representative of the real system dynamics, for if that were so, then there would be no need for using an IKF in the first place.

Hence, let it now be supposed that the true dynamics of the system are given by

\[ x(k + 1) = A_1 x(k) + B_1 u(k) + \omega(k) \]  
\[ z(k) = C_1 x(k) + \nu(k) \]

rather than the values given in Equation 7.10. One should wonder whether the ANN trained under the previous (initial) assumptions would still be able to correlate innovations with desired weightings in this new situation. Let an input sequence given by Equation 7.14 (Test Data 1) now be applied to the system described by Equations 7.15 to 7.17 and the corresponding KF and IKF estimates be calculated. Figures 7.7a, b and c show the input, the KF-1 innovations sequence together with the IKF interval widths, and a comparison of the desired weighting sequence with the output of the trained ANN, respectively. The MSE of the ANN prediction with respect to the desired weighting is 0.006058, a 77.6% reduction compared to the value of 0.027085 that results if a constant sequence of 0.5 is used.

This ascertains that the ANN trained from data generated through simulation by using some assumed system dynamics can still be applied successfully to the prediction of the desired IKF weighting sequence even when the true system dynamics differ from those assumed for training, as long as both lie within the intervals constitutive of the interval model.

A case study presenting how these concepts may be used in practise is detailed in the following section.
(a) Input signal $u(k)$ defined by Equation 7.14.

(b) KF-1 innovations sequence and $\Delta ikf$

(c) Comparison of desired weighting sequence and ANN output.

Figure 7.7 Performance of trained ANN on test set generated from modified system (Equations 7.15 to 7.17). (a) Input defined by Equation 7.14; (b) KF-1 innovations sequence and IKF interval width; (c) superposition of desired weighting sequence and ANN output.
7.3 Application to the Navigation of *Springer*

In this section, the ideas described previously are applied to the problem of estimating the heading angle of the *Springer* USV in a realistic mission scenario. Consider the autonomous tracking mission shown in Figure 7.8, consisting of the series of way-point coordinates with respect to a known origin given in Table 7.2.

Table 7.2. Coordinates of mission way points.

<table>
<thead>
<tr>
<th>Way point</th>
<th>start</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>x (m)</td>
<td>0</td>
<td>100</td>
<td>150</td>
<td>200</td>
<td>250</td>
<td>300</td>
<td>350</td>
<td>360</td>
<td>300</td>
<td>250</td>
</tr>
<tr>
<td>y (m)</td>
<td>0</td>
<td>10</td>
<td>20</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>-15</td>
<td>-45</td>
<td>-65</td>
<td>-40</td>
</tr>
</tbody>
</table>

Figure 7.8 Way-point tracking mission and trajectory followed assuming knowledge of the system state and actual dynamics. The dotted line shows the ideal path, whereas the continuous line the actual path taken by the vehicle in a simulation.

Figure 7.8 also depicts a simulation of the trajectory undertaken by the vehicle. Simulation the vehicle’s motion is carried out as described in Section 4.2, with the speed of the vehicle being constant and equal to $1 \text{ ms}^{-1}$, for which the steering dynamics are modelled by Equations 7.1 and 7.2 with the values of Equation 7.4. The vehicle is guided according to the line of sight method described in Section 4.4, in which the vehicle's location is assumed known at each sampling instant (e.g. via a GPS receiver). The autopilot regulates the differential speed component of the motors to steer the vehicle as necessary. A state feedback controller is used in this simulation, as described in Section 4.5.2.
Concretely, the control variable considered is the heading angle itself, and thus the steering model is augmented as,

\[
\theta(k+1) = \theta(k) + y(k) = \theta(k) + Cx(k)
\]  
(7.18)

\[
\begin{bmatrix}
x_i(k+1) \\ x_z(k+1) \\ \theta(k+1)
\end{bmatrix} =
\begin{bmatrix}
A & 0 & 0 \\ -C & 0 & 1 \\ 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
x_i(k) \\ x_z(k) \\ \theta(k)
\end{bmatrix} +
\begin{bmatrix}
B \\ 0 \\ 0
\end{bmatrix}
\begin{bmatrix}
u(k) \\ \omega_z(k)
\end{bmatrix}
\]  
(7.19)

\[
\theta(k) =
\begin{bmatrix}
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x_i(k) \\ x_z(k) \\ \theta(k)
\end{bmatrix}
\]  
(7.20)

so that the output is the heading angle. With the appropriate definitions, this is written compactly as

\[
\tilde{x}(k+1) = \tilde{A}\tilde{x}(k) + \tilde{B}u(k) + \tilde{\omega}(k)
\]  
(7.21)

\[
\theta(k) = \tilde{C}\tilde{x}(k)
\]  
(7.22)

The state-feedback control law is then implemented by:

\[
u(k) = -K\tilde{x}(k) + K_s r(k)
\]  
(7.23)

where \(r\) is the reference or desired heading, calculated by the guidance system.

The values of K are chosen so that the closed loop system dynamics (given by Equation 7.24) has a rise time of 10 s, deemed sufficient taking into account that physical constraints would constantly lead to actuator saturation if higher feedback gains were used (Section 4.2). The corresponding gain values are \(K = [-0.1500 \ 0.0950 \ 10.148]\). \(K_s\) is a scaling gain selected \textit{a posteriori} to ensure that the steady-state gain of the closed-loop system (given by Equations 7.24 and 7.22) is unity.

\[
\tilde{x}(k+1) = (\tilde{A} - \tilde{B}K)\tilde{x}(k) + \tilde{B}K_s r(k) + \tilde{\omega}(k)
\]  
(7.24)
Finally, the value of $u$ obtained from Equation 7.23 is subsequently hard-limited to between -1200 and 1200 rpm, and the maximum change from one time sample to the next to between -900 and 900 rpm, in order to emulate the actuator saturation described in Section 4.6.

Using the motion model, guidance system, and control law hitherto described, the vehicle’s trajectory and heading are then simulated from zero initial conditions,

$$
\begin{bmatrix}
    x_{USV}(0)
    \\
    y_{USV}(0)
    \\
    \theta(0)
    \\
    x(0)
\end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}
$$

(7.25)

where $(x_{USV}, y_{USV})$ represents the coordinates of the vehicle.

Advertently during this process, a change in the steering dynamics of the vehicle was prescribed upon its arrival at the first way-point as follows:

$$
A = A + 0.05|A|, \quad B = B + 0.05|B|, \quad C = C + 0.05|C|
$$

(7.26)

where it is understood that the absolute value is taken element-wise; that is, all the coefficients of the model were increased by 5%. It is to be noted nevertheless that the autopilot was furnished with the true (augmented) state-vector of the system at all time, and that on occurrence of the event described by Equation 7.26 the controller gains were recalculated in accordance with the changed vehicle dynamics. Hence, no detrimental effect is apparent in the waypoint tracking capability of the vehicle due to the alteration of its steering dynamics.

In practice however, the change in the system’s dynamics may not be known or predictable, nor the state-vector measurable. For the system modelled by Equations 7.1, 7.2 and 7.4, the components of the state vector $x$ have no particular correspondence with physical quantities since the model was obtained from input-output data alone. The components of the state vector must therefore be obtained via estimation: for example, using the gyroscope measurements, by employing a KF based on the predictive and measurement models given by Equations 7.1 to 7.4, as was done in the previous section.

Estimates of the augmented state vector $\tilde{x}$, rate of change of heading, and heading angle then ensure straightforwardly (Equations 7.27 to 7.29).
\[ \hat{x}(k) = \begin{bmatrix} \hat{x}(k) \\ \hat{\theta}(k) \end{bmatrix} \]  
(7.27)

\[ \hat{y}(k) = C \hat{x}(k) \]  
(7.28)

\[ \hat{\theta}(k) = \hat{\theta}(k-1) + C \hat{x}(k-1) \]  
(7.29)

However, as discussed previously, under incorrect modelling assumptions the KF estimate will be biased. For the previously generated trajectory (Figure 7.8), estimates of a KF (initialised with \( \hat{x}(0) = 0 \) and \( P(0) = 0 \)) of the heading-rate and heading angle are shown in Figures 7.9a and 7.9b respectively, along with the corresponding true (simulated) values of these. The controller action is shown in Figure 7.9c. It can be observed how after an initial period of approximately 70 s (by which time the vehicle has just crossed the first way-point whereupon the vehicle's dynamics was altered according to Equation 7.26), the KF heading-estimate starts to drift away from the real heading, due to mismatches between the estimated and real heading-rates as the KF continues to predict based upon the initial model.

To illustrate the effect of this biased KF estimate on the mission performance, a simulation in which the KF estimate of the state vector is used in the control law is shown in Figure 7.10. It can be observed that the vehicle struggles significantly during the latter, more demanding stage of the course, deviating substantially from the ideal path, and failing to complete the mission (recall that the guidance system has additional incorporated logic that decides when a way-point is considered to have been missed, whereupon it shifts target to the following way-point).

Table 7.3 provides a summary of the mission performances of both simulations. It must be noted though that performance is quite similar until way point 7, as the estimation error of the KF remains small up until this point because the demand on vehicle turning is low.
Figure 7.9 Data corresponding to simulation shown in Figure 7.8. (a) True and estimated rate of change of heading; (b) true and estimated heading angle; (c) applied differential motor speed. The KF estimates are based on the initially assumed steering dynamics only.

Figure 7.10 Simulation of the trajectory followed when the KF estimate is used in the control law.
Table 7.3 Comparison of mission performances.

<table>
<thead>
<tr>
<th></th>
<th>Using true state vector</th>
<th>Using KF estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of way points reached</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>Total distance travelled (m)</td>
<td>508</td>
<td>545</td>
</tr>
<tr>
<td>Average deviation (m)</td>
<td>5.07</td>
<td>5.88</td>
</tr>
<tr>
<td>Average controller energy (rps²)</td>
<td>4.29</td>
<td>38.8</td>
</tr>
</tbody>
</table>

In view of the effects of erroneous modelling assumptions, henceforth let it be admitted that the precise model of the vehicle’s steering dynamics may vary and cannot be not known, though it can nevertheless be ascertained to be contained within the interval model described by Equations 7.5 to 7.7. In this scenario, the technique described in the previous section may be put into practice: a set of dynamics contained within the interval model may be assumed as the ‘true’ vehicle dynamics (and hence used to simulate the vehicle’s turning motion), and the estimates of both a KF based on it and a KF based on the nominal model (Equations 7.1 to 7.4) can be simulated, together with the interval estimates from an IKF founded on the interval model. The desired weighting sequence can then be obtained as that which is necessary for the wIKF values to match those of the KF estimates that were obtained using the assumed ‘true’ dynamics. Finally, an ANN can be trained to obtain these desired weights from the innovations sequence of the biased KF (KF that uses the nominal system model).

In order to train the ANN, rather than use the way-point mission described earlier to generate the required input and target data, a different mission was used instead. This allows the mission described previously to be used to test the method in order to evaluate its performance. The training mission, consisting of eight way-points as shown in Figure 7.11, was crafted to elicit representative USV manoeuvre requirements and of varied magnitudes, in order to excite a wide range of dynamics and obtain a richer training set.

Another point for consideration is what model to choose from the interval model to represent the ‘true’ dynamics of the vehicle for simulation. Instead of choosing a single model for the whole mission, different sets of models were chosen for different time intervals. The coefficients are given by the matrices $A_1(k)$, $B_1(k)$, and $C_1(k)$ as specified in Table 7.4.
The training data set is thus generated by effectively simulating several different systems, consecutively, that span various combinations of values selected from the interval model, hence providing a richer training set than would be generated using a constant set of dynamics for simulating the vehicle for the entire duration of the mission.

Based on these varying dynamics, Figure 7.11 shows the actual path taken by the vehicle. It is to be noted that for the trajectory followed, the autopilot was again given the true value of the state vector. The control input generated by the autopilot is shown in Figure 7.12a. In addition, the rate estimates of both a KF based on the nominal model, and a KF based on the true dynamics used to simulate the vehicle, are shown in Figure 7.12a, along with the estimates of an IKF based on the interval model (Equations 7.5 to 7.7).
Figure 7.12 Data set and training results obtained from training mission: (a) applied differential motor speed; (b) KF, ideal KF and IKF estimates; (c) innovations sequence of the nominal KF; (d) IKF interval widths; (e) comparison of desired weighting sequence and trained ANN output.
Figure 7.12c shows the innovations sequence of the nominal (biased) KF, and Figure 7.12d the IKF interval widths, both of which are used as inputs to the ANN. Lastly, Figure 7.12e shows the desired (target) weighting sequence (calculated so that the wIKF estimate matches the unbiased, or ideal, KF estimate). This input and target data-set was used to train the ANN of Figure 7.3. The trained network’s output is plotted alongside the target output in Figure 7.12e. The training accuracy is quantified by a MSE of 0.002913, in contrast to the average squared difference of 0.01374 between the desired weighting and a constant value of 0.5, a reduction of approximately 80%.

In the original way point mission (Figure 7.8), the dynamics of the vehicle was initially given by the nominal model (Equations 7.1 to 7.4), but then changed as described by Equation 7.26 upon arrival at the first way point. However, in order to test the trained ANN on the original way point mission, as well as this change, two other scenarios in which different changes occur are considered as well, and are summarised in Table 7.5.

Table 7.5 Change of dynamics prescribed upon reaching the first way point for each of the test missions.

<table>
<thead>
<tr>
<th>Test mission 1</th>
<th>Test mission 2</th>
<th>Test mission 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A = A + 0.05</td>
<td>A</td>
<td>$</td>
</tr>
<tr>
<td>$B = B + 0.05</td>
<td>B</td>
<td>$</td>
</tr>
<tr>
<td>$C = C + 0.05</td>
<td>C</td>
<td>$</td>
</tr>
</tbody>
</table>

In all three test missions, the initial vehicle dynamics coincides with that of the nominal model (Equations 7.1 to 7.4), and the estimates of a KF based on the nominal model and those of an ideal KF coincide. From the first way point onwards, the values of the ‘true’, or simulated, vehicle dynamics were changed according to Table 7.5.

Using the true value of the state vector for feedback, simulations of the three test missions were carried out. For each one, the innovations of a nominal-system KF and the interval widths of an IKF were calculated, as well as the desired weighting sequence (by imposing that the wIKF estimate equal that of an ideal KF). The previously trained ANN was then used to predict the desired weight based on innovations and IKF widths. The results for each test case are depicted in Figures 7.13a to 7.13c respectively, and Table 7.6 summarises for each one the reduction in MSE with respect to the desired weighting sequence.

129
(a) Test mission 1 desired weighting sequence and ANN output.

(b) Test mission 2 desired weighting sequence and ANN output.

(c) Test mission 3 desired weighting sequence and ANN output.

Figure 7.13 Comparison of desired weighting sequence and trained ANN output for (a) test mission 1; (b) test mission 2; (c) test mission 3.
Table 7.6 Test performances for the trained ANN.

<table>
<thead>
<tr>
<th></th>
<th>MSE0 (based on arithmetic average of IKF bounds)</th>
<th>MSE (based on ANN output)</th>
<th>Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test mission 1</td>
<td>0.027428</td>
<td>0.004051</td>
<td>85.23%</td>
</tr>
<tr>
<td>Test mission 2</td>
<td>0.003063</td>
<td>0.001097</td>
<td>64.2%</td>
</tr>
<tr>
<td>Test mission 3</td>
<td>0.010679</td>
<td>0.001585</td>
<td>85.2%</td>
</tr>
</tbody>
</table>

The results show that the trained ANN can be applied successfully to predict the desired weights required for the wIKF estimate to approximate the optimal (ideal KF) estimate.

As stated, the comparison of Table 7.6 is based on data generated from simulating the test missions using the true state vector for control (as well as updating the control law itself in accordance with the changed dynamics). Figure 7.14 depicts the trajectories followed for each test mission when the control law is not adapted to the changed vehicle dynamics, and in which the respective wIKF estimates are used instead of the true state. For these simulations, the mission performances are benchmarked against those of the original mission in which the true state vector was used by the controller (Figure 7.8), and tabulated in Table 7.7. As seen, in all three cases all of the way points are reached successfully, and there is little difference in mission performance except for a modest increase in the average controller energy used with respect to the benchmark performance.

Table 7.7 Comparison of mission performances.

<table>
<thead>
<tr>
<th></th>
<th>Benchmark mission</th>
<th>Test mission 1</th>
<th>Test mission 2</th>
<th>Test mission 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of way points reached</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>Total distance travelled (m)</td>
<td>508</td>
<td>506</td>
<td>505</td>
<td>514</td>
</tr>
<tr>
<td>Average deviation (m)</td>
<td>5.07</td>
<td>4.47</td>
<td>4.76</td>
<td>5.51</td>
</tr>
<tr>
<td>Average controller energy ((rps)^2)</td>
<td>4.29</td>
<td>8.57</td>
<td>7.73</td>
<td>8.03</td>
</tr>
</tbody>
</table>
Figure 7.14 Trajectory followed by the vehicle for the three test missions when the wIKF estimates are used in the control law.

### 7.4 Summary

This chapter has demonstrated how an ANN can be trained successfully to use residual KF data (the innovations sequence) to infer advantageous weightings for obtaining point-valued estimates from IKF boundaries, as compared to simply using, for example, the arithmetic mean of the boundaries (which provides similar estimates to that of the KF that uses the incorrect nominal model). The test results for the case study presented here show that the trained
ANN is capable of generalising well to new situations. Depending on the application, it is always possible to develop an adequate training set that will enable effective prediction for new missions within the scope of the application. This required analysis, the training process, and evaluation of the trained network on a set of new missions can all be done beforehand and via simulation alone, rendering this method cost-effective, reliable, and practically realisable.

A pending comment with regard to the particular example used in this paper to demonstrate the technique developed, is the use of $\Delta ikf(k)$ as additional input to the ANN. Its use was seen to increase the predictive accuracy of the network, especially to correct the scaling of the prediction. A heuristic explanation for this phenomenon is the following. The IKF intervals themselves inevitably tend to widen more or less depending on the sharpness of the interval computation, and may vary significantly depending on the method used. However, they all represent the same ‘optimal interval’ that would be obtained if interval computation could be carried out with infinite sharpness. In other words, the ANN developed here should be immune to the exact width of the IKF interval and sensitive only to the innovations of the biased KF estimate. It thus requires information of the former, which should somehow be incorporated into the ANN prediction process, since its target output, the optimal $wIKF$ estimate, is computed from the IKF bounds themselves.

It should also be said that the ANN architecture presented here (Figure 7.3), and its particular characteristics (layer sizes, etc.), was found to provide a good balance between prediction accuracy and number of neurons employed, but is by no means the only valid architecture that can be used, and furthermore, for each system the most suited type and size of ANN should be explored.
Chapter 8

Multi-Sensor Data Fusion

“The goal is not to sail the boat, but rather to help the boat sail herself.” — John Rousmaniere

The three types of digital compass units available on the Springer USV, whose characteristics were described in Chapter 3, provide redundant heading information for the vehicle to navigate. This information redundancy is finally exploited in this chapter by means of a fuzzy logic based multi-sensor fusion algorithm that is capable of fusing various wIKF estimates to provide a heading estimator for vehicle that is both robust and fault tolerant.

Concretely, the inertial data from the vehicle’s gyroscope, prone to sporadic bias drifts, is fused individually with readings from each of the compasses via a wIKF which is robust to gyroscope bias drifts. The three ensuing wIKF estimates of the heading angle of the vehicle are then fused via a fuzzy logic algorithm designed to provide an accurate heading even in the face of a failure of up to two of the compasses at any time. Simulations demonstrate the effectiveness of the proposed method.

8.1 A Fuzzy Data-Fusion Algorithm for Kalman Estimates

As detailed in Chapter 3, the navigational suite of Springer includes a low cost MEMS gyroscope unit and three digital magnetic compasses for heading determination, the main characteristics of which were summarised in Table 3.3.
The sensor redundancy may appear wasteful, but in practice, sensor failure is a common occurrence, especially when low cost hardware is involved. By way of example, during some recent trials undertaken with the vehicle, a sporadic communications error between one of the compasses and the main on board PC impeded the data being sent from the former to the latter. In this case it sufficed to manually switch to another compass, but during an autonomous mission, such a luxury would not exist and a hardware failure in the middle of a mission would most likely result in its forced abortion.

In order to filter measurement noise, a KF can be built to fuse data between the gyroscope and each individual compass, as detailed in Section 4.3. The state and measurement models are reproduced in Equations 8.1 and 8.2:

\[
x(k + 1) = x(k) + T_s u(k) - T_s \omega(k) \quad (8.1)
\]

\[
z(k) = x(k) + v(k) \quad (8.2)
\]

where \( x \) represents the heading of the vehicle, the input \( u(k) \) is the gyroscope reading, also denoted as \( \Omega_0(k) \), and the output \( z(k) \) the compass measurement. The gyroscope and compass measurement noise are modelled as

\[
\omega(k) \sim N(0, Q), \quad Q = 0.05^2 (deg/s)^2 \quad (8.3)
\]

\[
v(k) \sim N(0, R), \quad R = (0.5 \, deg)^2 \text{ or } (1 \, deg)^2 \quad (8.4)
\]

as per Table 3.3.

Hence this would result in three distinct KFs that are identical in their predictive models (Equation 8.1) but with different compass measurement noise variances. However, if a compass were to fail (either permanently or intermittently), the corresponding KF performance would be significantly degraded. There should henceforth exist some mechanism by which a faulty KF estimate should automatically be rejected in the vehicle’s navigation system.

Denoting the heading estimate of the \( i^{th} \) KF as \( \hat{\theta}_{KF_i}(k) \), the fuzzy data-fusion algorithm proposed here assigns a weight to each of the three KF estimates, so that the fused state estimate may be computed as

\[
\hat{\theta}_{fKF}(k) = \sum_{i=1}^{3} w_i(k) \hat{\theta}_{KF_i}(k) \quad (8.5)
\]
The weighting decision is based on assessing the innovation sequence of each KF. Recall that the innovation sequence of a KF is defined as the difference between the measurement of the output and the predicted value described as

\[
\{inn(k)\} = \{z(k) - C \hat{x}(k | k - 1)\}
\]  

(8.6)

which in this case corresponds simply to the difference between the compass measurement at time \(k\) and the predicted heading given the previous heading estimate and the gyroscope reading at \(k - 1\). It is well established that under an ideal scenario, the innovation sequence should be comprised of a zero-mean white noise sequence (Bijker and Steyn, 1997; Subramanian et al., 2009). Thus a KF’s innovation sequence could be monitored to detect a failure in the KF which may then be used to penalise its contribution in Equation 8.6.

In order to monitor the innovation sequence, which in general is a random process hence its values when considered individually are meaningless, a simple moving average (SMA) of the innovation sequence of each KF is computed:

\[
SMA(k) = m^{-1} \sum_{i=k-m+1}^{k} inn(i)
\]  

(8.7)

where \(m\) is the number of samples considered in the moving average. Since the SMA is, in the ideal case, a sum of zero-mean independent random variables, it is in itself a zero-mean random variable, tending to be normally distributed by the Central Limit Theorem. However, its variance is \(m\) times smaller than that of the innovation random variable. Thus, sporadic high values of the SMA are more improbable than for the innovation, and will almost only occur when the innovation stops being a white sequence. Hence it is this value that is chosen to indicate a compass fault in the KF estimate.

It is intuitive to define a set of rules based on this idea to decide whether or not to penalise the contribution of a KF to the fused estimate; basically such rules should say: if the \(SMA\) is somewhat larger or smaller than zero, then decrease the weight of the corresponding KF; else if it is zero, then increase the weight of that KF. In order to quantify these, consider the following membership functions, shown graphically in Figure 8.1, in which SMAN, SMAP, DWN and DWP are some threshold values:
**Input membership functions:**

Negative function: 
\[ \mu_N^i = \begin{cases} 
1 & \text{if } SMA < SMAN \\
SMA/SMAN & \text{if } SMAN \leq SMA < 0 \\
0 & \text{if } SMA \geq 0
\end{cases} \]  
\[(8.8)\]

Zero function: 
\[ \mu_Z^i = \begin{cases} 
(1 - SMA/SMAN) & \text{if } SMAN \leq SMA < 0 \\
(1 - SMA/SMAP) & \text{if } 0 \leq SMA \leq SMAP
\end{cases} \]  
\[(8.9)\]

Positive function: 
\[ \mu_P^i = \begin{cases} 
0 & \text{if } SMA < 0 \\
SMA/SMAP & \text{if } 0 \leq SMA < SMAP \\
1 & \text{if } SMA \geq SMAP
\end{cases} \]  
\[(8.10)\]

**Output membership functions:**

Negative function: 
\[ \mu_N^o = \begin{cases} 
1 & \text{if } DWN \leq SMA < 0 \\
0 & \text{otherwise}
\end{cases} \]  
\[(8.11)\]

Positive function: 
\[ \mu_P^o = \begin{cases} 
1 & \text{if } 0 \leq SMA < DWP \\
0 & \text{otherwise}
\end{cases} \]  
\[(8.12)\]

As indicated by the output fuzzy membership functions, the output to the fuzzy logic inference system is chosen to be a change in the weight of the filter, \(\Delta w\), rather than the weight itself. This is to avoid abrupt transitions in the overall estimate.

Based upon the aforedescribed membership functions, the following fuzzy rules are established:

---

(a) Input membership functions.  
(b) Output membership functions.

Figure 8.1. Input and output membership functions.
Rule 1: If SMA negative then $\Delta w$ is negative.

Rule 2: If SMA is zero then $\Delta w$ is positive.

Rule 3: If SMA is positive then $\Delta w$ is negative.

Then, at each sampling time $k$, depending upon the value of the SMA, $\Delta w$ is computed as follows:

- Case 1: SMA $<$ SMAN

Rule 1 applies and $\Delta w$ is given by the horizontal projection of the centroid of the negative output membership function, i.e. $\Delta w = \frac{DWN}{2}$.

- Case 2: SMAN $<\text{SMA} \leq 0$

Both Rule 1 and Rule 2 apply. Let $\mu^i_N$ represent the degree of membership of the input to the Negative input membership function (Rule 1), and $\mu^i_Z$ its degree of membership to the Zero input membership function (Rule 2). Then $\Delta w$ is computed as the horizontal projection of the centroid of the area comprising the portions of the Negative and Positive output membership functions below the values $\mu^i_N$ and $\mu^i_Z$ respectively (Figure 8.2):

$$\Delta w = \frac{-\frac{1}{2} DWN^2 \times \mu^i_N + \frac{1}{2} DWP^2 \times \mu^i_Z}{-DWN \times \mu^i_N + DWP \times \mu^i_Z} \quad (8.13)$$

- Case 3: $0 <$ SMA $<$ SMAP

Both Rule 3 and Rule 3 apply. Let $\mu^i_Z$ represent the degree of membership of the input to the Zero input membership function (Rule 2), and $\mu^i_P$ its degree of membership to the Positive input membership function (Rule 3). Then $\Delta w$ is computed as the horizontal projection of the centroid of the area comprising the portions of the Positive and
Negative output membership functions below the values $\mu^i_Z$ and $\mu^p_i$ respectively:

$$\Delta w = -\frac{1}{2} DWN^2 \times \mu^p + \frac{1}{2} DW P^2 \times \mu^i_Z - DWN \times \mu^p + DW P \times \mu^i_Z$$  \hspace{1cm} (8.14)

- **Case 4: SMAP ≤ SMA**

*Rule 3* solely applies, and $\Delta w$ is given by the horizontal projection of the centroid of the negative output membership function, i.e. $\Delta w = \frac{DWN}{2}$.

![Figure 8.2](image)

Figure 8.2. Calculation of the output $\Delta w$ for Case 2 (SMAN < SMA ≤ 0).

Once $\Delta w$ has been calculated at time step $k$ for each KF ($\Delta w_i(k), i = 1,2,3$), these values can be normalised so that their sum equals zero to ensure that the sum of the weights themselves will remain equal to one, as the weights are initialised equally at $1/3$ for $k = 0$,

$$\Delta w^*_i(k) := \Delta w_i(k) - \alpha, \ i = 1,2,3, \ \text{with } \alpha \text{ such that}$$

$$\sum_{i=1}^{3}(\Delta w_i - \alpha) = 0, \ \text{i.e. } \alpha = \frac{1}{3} \sum_{i=1}^{3} \Delta w_i$$  \hspace{1cm} (8.15)

resulting in the updated weights of each filter given by

$$w_i(k) := w_i(k - 1) + \Delta w^*_i(k), \ i = 1,2,3$$  \hspace{1cm} (8.16)

However, direct application of Equation 8.16 might result in updated values of the weights not restricted to the interval $[0, 1]$. To restrict the values of the weights to this interval, the following redistribution procedure is applied.
Instead of directly updating all the weights according to Equation 8.16, these are tentatively updated in some auxiliary variables:

\[ w_i^* := w_i(k-1) + \Delta w_i^*(k), \quad i = 1,2,3 \]  \hfill (8.17)

Three possibilities exist:

- If all \( w_i^* \)'s are between 0 and 1 (inclusive), then these are taken directly as the updated weights \( w_i(k) \), (Equation 8.16).

- If (only) one of the \( w_i^* \) is less than zero, e.g. \( w_j^* < 0 \), then \( \Delta w_j^{**} \) is defined as:

\[ \Delta w_j^{**}(k) := -w_j(k-1), \quad i = 1,2,3 \quad \text{and} \quad i \neq j, \]

with \( \alpha \) such that

\[ \Delta w_j^{**}(k) + \sum_{i=1}^{3} \Delta w_i^* - \alpha = 0, \]

whereby \( \alpha = \frac{1}{2} [\Delta w_j^{**}(k) + \sum_{i=1}^{3} \Delta w_i^*] \). The new weights are then given by

\[ w_i^{**} := w_i(k-1) + \Delta w_i^{**}(k), \quad i = 1,2,3, \]

where in particular \( w_j^{**} := w_j(k-1) + \Delta w_j^{**}(k) = 0 \). If none of the resulting \( w_i^{**} \) are negative, then these are the updated weights \( w_i(k) \); however, if one of them is negative, e.g. \( w_i^{**} < 0 \), then the updated weights are \( w_j(k) := 0, w_l(k) := 0, \) and

\[ w_i(k) := 1, \quad i \in \{1,2,3\} \quad \text{and} \quad i \neq j, l. \]

- If two of the \( w_i^* \) obtained using Equation 8.17 are negative, e.g. \( w_j^* < 0 \) and \( w_l^* < 0 \), then that implies that the third weight, \( w_i^*, i \in \{1,2,3\} \quad \text{and} \quad i \neq j, l, \) will be larger than one, since the sum of the three is always equal to unity. Therefore it suffices to take \( w_j(k) := 0, w_l(k) := 0, \) and \( w_i(k) := 1. \)

This scheme allows for weights that at some point devolve to a zero value, signifying complete rejection of the corresponding KF, to start recovering if and when they are subsequently prescribed positive weight increments. A similar scheme without recovery is easily implemented by initially assigning a raised flag to each KF. Upon a KF being assigned a weight of zero at some time-step, its flag is lowered, or labelled inactive, meaning that its weight is permanently kept at zero from then onward regardless of the weight changes prescribed by the fuzzy logic system, and the weight redistribution process would involve only the remaining active KFs.
The fuzzy logic assignment of weight increments and ensuing normalisation/redistribution occurs only after an initial $m$ time-steps have elapsed, which is the number of samples required to compute the SMA of the innovations sequences. During the initial period, the weights are maintained with equal values of $1/3$.

Consider the way point tracking mission of Section 4.1, reproduced in Figure 8.3. Based on the vehicle motion model used in Section 4.6, given by Equations 4.2 to 4.5, with steering dynamics specified by $\text{eig}(A) = 0.2 \pm i 0.25$ and a constant speed of $v = 1.5 \text{ms}^{-1}$, simulation of the mission is shown in Figure 8.4. The guidance is based on line of sight as described in Section 4.4, and the autopilot consists of the state feedback control law

$$u(k) = -K \tilde{x}(k) + K_s r(k) \tag{8.18}$$

described in Section 4.5.2, where $\tilde{x} \equiv [x, \theta]^t$ is the augmented state vector, and the feedback gains chosen as $K = [-0.5500 \ 0.2150 \ 6.1032]$ so that the closed loop steering dynamics has a rise time of 10 s, deemed sufficient taking into account that physical constraints would constantly lead to actuator saturation if higher feedback gains were used. The scaling gain $K_s$ is selected to ensure that the steady-state gain of the closed-loop system is unity. The simulation is carried out using the simulated state vector in Equation 8.18, although in practice the estimated state must be used instead.

![Figure 8.3 Way point tracking mission.](image)
Based on the simulated vehicle heading at each time step, the noisy gyroscope readings and compass measurements are simulated as well by generating pseudo random values from a normal distribution with zero mean and corresponding variance. Three different compasses are simulated, with noise processes $\nu_i \sim N(0, r_i)$, along with their respective KFs (fusion of gyroscope with each individual compass), as shown in Table 8.1, initialised with $\hat{\theta}_{KF_i}(0) = 0$ and $P_{KF_i}(0) = 0$, $i = 1, 2, 3$.

Table 8.1. KF characteristics.

<table>
<thead>
<tr>
<th></th>
<th>Gyroscope noise model</th>
<th>Compass noise model</th>
</tr>
</thead>
<tbody>
<tr>
<td>KF₁</td>
<td>$q = 0.05^2 (\text{deg/s})^2$</td>
<td>$r_1 = 0.25 \text{ deg}^2$</td>
</tr>
<tr>
<td>KF₂</td>
<td></td>
<td>$r_2 = 1 \text{ deg}^2$</td>
</tr>
<tr>
<td>KF₃</td>
<td></td>
<td>$r_3 = 9 \text{ deg}^2$</td>
</tr>
</tbody>
</table>

In order to test the fault tolerance of the fused KF estimate, during the course of the simulation two of the compasses are made to fail in such a way that their readings remain frozen at the last value before failure. In particular, the compass associated with KF₃ freezes at $k = 150$ whilst the one associated with KF₁ does so at $k = 350$. The actual vehicle heading at each time-step and the three individual KF estimates are shown Figure 8.5a. It can be observed how, after a compass failure, the KF estimates tend to the respective frozen compass
measurements when the gyroscope reading is small (i.e. during the straight-line segments of the trajectory).

The fused KF estimate is also shown in Figure 8.5a. In order to understand the fused estimate, the innovations of each KF are shown in Figure 8.5b, and their SMAs in Figure 8.5c, based upon which the fuzzy-logic system calculates the weight increments for each filter, the resulting weights being those shown in Figure 8.5d. The SMA length and fuzzy membership function threshold values were chosen heuristically and are given in Table 8.2. Once assigned a zero value, weights were not permitted to subsequently recover.

(a) Actual heading, KF estimates, and fuzzy-logic fused KF estimate.

(b) Innovations sequence of each KF.
Figure 8.5 Simulation of way-point tracking mission: (a) actual heading, KF heading estimates, and fuzzy-logic fused KF estimate; (b) innovations sequence of each KF; (c) SMA of each KF innovations; (d) fuzzy weights assigned to each KF.

Table 8.2 Parameter values for fusion algorithm.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMA length</td>
<td>20</td>
</tr>
<tr>
<td>SMAN</td>
<td>-5</td>
</tr>
<tr>
<td>SMAP</td>
<td>5</td>
</tr>
<tr>
<td>DWN</td>
<td>-0.05</td>
</tr>
<tr>
<td>DWP</td>
<td>0.05</td>
</tr>
</tbody>
</table>

Table 8.3 summarises the RMS errors of the three KF estimates and the fused estimate. Note that the majority of the error of the fused estimate occurs due to
the transient periods shortly after the compass failures, as the fusion weights need time to adjust.

Table 8.3 KF and fused KF estimate errors.

<table>
<thead>
<tr>
<th>Method</th>
<th>Heading RMS error (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>KF_1</td>
<td>112.75</td>
</tr>
<tr>
<td>KF_2</td>
<td>0.21</td>
</tr>
<tr>
<td>KF_3</td>
<td>147.28</td>
</tr>
<tr>
<td>fused estimate</td>
<td>0.72</td>
</tr>
</tbody>
</table>

8.2 Robustness and Interval Kalman Filtering

It has been shown in previous chapters how inaccurate system modelling degrades the KF estimate. In the case of the KF used here to fuse gyroscope and compass data, consider what happens if the gyroscope is susceptible to developing some bias, as low cost MEMS gyros are typically subject to null drift due to various reasons (Shiau et al., 2012). The gyroscope reading can then be considered to be the sum of three components: the actual turning rate $\Omega_i$, a bias $b$, and a measurement noise $\omega$ (Equation 8.19, Figure 8.6),

$$\Omega_0 = \Omega_i + b + \omega$$  \hspace{1cm} (8.19)

Figure 8.6 Gyro measurement model: $\Omega_i$ is the actual rate of change of heading angle of the vehicle whereas $\Omega_0$ is the value output by the gyroscope mounted on the vehicle.

The predictive model of the USV heading angle based on gyroscopic readings then becomes

$$\theta(k + 1) = \theta(k) + T_s \times [\Omega_0(k) - b - \omega(k)]$$  \hspace{1cm} (8.20)
If the precise value of the gyroscope bias cannot be known (e.g., if it is susceptible to change) but however its value can be bounded, \( b_{\text{min}} \leq b \leq b_{\text{max}} \), then this predictive model can be written as an interval model. For the purpose of applying an IKF, the state dynamic and measurement equations may be expressed as

\[
\begin{bmatrix}
    x(k+1) \\
    0
\end{bmatrix} = \begin{bmatrix}
    1 & 0 \\
    0 & 0
\end{bmatrix} \begin{bmatrix}
    x(k) \\
    0
\end{bmatrix} + \begin{bmatrix}
    T_s & -T_s [b_{\text{min}}, b_{\text{max}}] \\
    0 & 0
\end{bmatrix} \begin{bmatrix}
    u(k) \\
    1
\end{bmatrix} + \begin{bmatrix}
    -T_s \omega(k) \\
    0
\end{bmatrix} \tag{8.21}
\]

\[
z(k) = \begin{bmatrix}
    1 & 0
\end{bmatrix} \begin{bmatrix}
    x(k) \\
    0
\end{bmatrix} + v(k) \tag{8.22}
\]

where \( x(k) \) represents the vehicle heading, \( \theta(k) \). The IKF then yields an interval estimate at each time step, \( \hat{x}_{\text{IKF}}(k) \). A point valued model contained in the interval model (Equation 8.21) would simply be

\[
\begin{bmatrix}
    x(k+1) \\
    0
\end{bmatrix} = \begin{bmatrix}
    1 & 0 \\
    0 & 0
\end{bmatrix} \begin{bmatrix}
    x(k) \\
    0
\end{bmatrix} + \begin{bmatrix}
    T_s & -T_s b \\
    0 & 0
\end{bmatrix} \begin{bmatrix}
    u(k) \\
    1
\end{bmatrix} + \begin{bmatrix}
    -T_s \omega(k) \\
    0
\end{bmatrix} \tag{8.23}
\]

\[
z(k) = \begin{bmatrix}
    1 & 0
\end{bmatrix} \begin{bmatrix}
    x(k) \\
    0
\end{bmatrix} + v(k) \tag{8.24}
\]

for some value \( b \in [b_{\text{min}}, b_{\text{max}}] \).

Though the true value of \( b \) is not known, the ideal KF estimate, or estimate of a KF that is based on the true system dynamic model, is equal to some weighted average of the IKF estimate, with the weight being between 0 and 1 if the true value of \( b \) indeed lies in the interval \([b_{\text{min}}, b_{\text{max}}]\). Thus, based on the IKF interval estimate of the system output, or heading angle, \( y_{\text{IKF}} = C^t \hat{x}_{\text{IKF}}(k) \), a wIKF estimate can be obtained using the methodology described in the previous chapter. This is done by training an ANN to predict the weight that yields a weighted average of bounds of \( y_{\text{IKF}} \) at each time step that equals the output that would be estimated by an ideal KF,

\[
w(k) \in [1, 0] : \min\left( C^t \hat{x}_{\text{IKF}}(k) \right) + w(k) \left[ \max\left( C^t \hat{x}_{\text{IKF}}(k) \right) - \min\left( C^t \hat{x}_{\text{IKF}}(k) \right) \right] = \hat{x}_{\text{ideal KF}}(k) \tag{8.25}
\]

where \( \max \) and \( \min \) refer to the maximum and minimum values of the interval.
As detailed in the previous chapter, the ANN is trained to model the correlation between the innovations of a nominal KF (based on some nominal model contained within the interval model) and the desired weight, that is, provide a mapping \( \text{inn}_{K_{F}n_{0}m_{a}}(k) \xrightarrow{\text{ANN}} w(k) \): Equation 8.25 is satisfied.

Of course, in order to train the ANN, it is necessary therefore to have the ideal KF estimate, and hence, the correct model of the system. This training procedure can however be based on simulation alone, i.e., hypothesised true and nominal dynamics. The trained ANN can then be used to predict this desired weight independently of the assumed true system dynamics and nominal models used for training, as long as they lie within the interval model that describes the bounded uncertainty.

Consider an example in which \( b_{\text{min}} = -4 \ \text{deg/s} \) and \( b_{\text{max}} = 4 \ \text{deg/s} \). In order to train an ANN as described earlier, a training mission was established consisting of a different set of way points. In this training mission, an IKF was simulated along with a KF based on a nominal model given by Equations 8.23 and 8.24 with \( b = 0 \). The gyroscope readings were simulated for different lengths of time with different biases between \(-4\) and \(4\) deg/s, and an ideal KF was also simulated (based on a model given by Equations 8.23 and 8.24 with \( b \) always being equal to the "true" bias, or bias used to simulate the gyroscope readings).

As in the previous chapter, an MLP ANN architecture was used, this time with up to three time-delays in the inputs, one hidden layer of five neurons with hyperbolic tangent activation functions, and a linear output neuron.

Results of simulating the way-point mission described in the previous section are given in Figure 8.7, during which the gyroscope was prescribed a bias of \(-1\) deg/s for \( 0 \leq k \leq 300 \), \(3.95\) deg/s for \( 300 < k \leq 650 \), and of \(-3.75\) deg/s for the remainder of the simulation (Figures 8.7a and 8.7b). The actual heading and IKF estimate bounds are shown in Figure 8.7c, and the nominal KF and wIKF estimates in Figure 8.8d, which clearly shows how the KF estimate is degraded due to the incorrect model used. Figure 8.8e shows the innovations sequence of the nominal KF, which is no longer white.
(a) Actual turning rate and gyroscope measurement.

(b) Gyroscope bias.

(c) Actual heading and IKF estimate bounds.

(d) Actual heading, nominal KF estimate, and wIKF estimate.
8.3 Robust and Fault Tolerant Heading Estimation

Fault tolerance here refers to being able to operate in spite of compass failure, which the fuzzy KF fusion algorithm was designed to provide by exploiting the sensor redundancy. On the other hand, robustness is used with reference to a KF being able to predict accurate heading estimates even in the face of modelling uncertainty, in this case, unknown (but bounded) gyroscope bias, through the use of the wIKF. This section proposes the fuzzy fusion of wIKF estimates to provide both fault tolerance and robustness.

Consider the same way-point tracking simulation of Section 8.1, with estimates of three KFs, each associated with one of the three compass units previously described (Table 3.3), and during which the readings of the compasses associated with KF$_3$ and KF$_1$ are frozen as before at $k = 150$ and $k = 350$ respectively. In addition, however, the same gyroscope biases described in Section 8.2 are prescribed, whilst the KFs assume zero gyroscope bias models.

The simulation results are shown in Figure 8.8, with the true heading and three KF estimates shown in Figure 8.8a, along with the fused estimate of the same. Because none of the KF estimates are accurate, neither can the fused estimate be expected to be so. Moreover, if one analyses the innovations of the KFs

Figure 8.7 Simulation of way-point tracking mission with imposed gyroscope bias, (a) actual turning rate and gyroscope measurement; (b) gyroscope bias; (c) actual heading and IKF estimate bounds; (d) actual heading, nominal KF estimate, and wIKF estimate; (e) nominal KF innovations.
(Figure 8.8b), these are not white even during the period before compass failure occurs. Hence, even the assignment of weights by the fuzzy algorithm is unsatisfactory, as can be seen in Figure 8.8d, as KF₃ is eventually awarded the largest weight, even though its estimate is completely erroneous.

(a) Actual heading, KF estimates, and fused KF estimates.

(b) KF innovations.

(c) SMA of KF innovations.
The question that remains is whether the fuzzy logic based algorithm would work to fuse the wIKF estimates, which are, as shown in the previous section, robust to gyroscope biases. However, no standard definition of the wIKF innovation sequence currently exists. The most intuitive proposition would be to define the wIKF innovations as weighted averages of the corresponding IKF innovations (which are intervals at each time step), applying the same weights used to compute the wIKF state and output estimates. It turns out however that the most adequate weights for defining the wIKF innovations are instead the complements to unity of these.

**Conjecture.** Let $\hat{x}^{IKF}(k)$ be the IKF state estimate of a system based on an interval state space model as given by Equations 5.15 and 5.16, and $\hat{x}^{KF}(k)$ the estimate of a KF based on some point-valued model contained within the interval model. Consider the weights

$$w(k) \in [1, 0]:$$

$$\min(C^{1} \hat{x}^{IKF}(k)) + w(k)[\max(C^{1} \hat{x}^{IKF}(k)) - \min(C^{1} \hat{x}^{IKF}(k))] = C \hat{x}^{KF}(k) \quad (8.26)$$

and

$$w_{2}(k): z(k) - \{\min(C^{1} \hat{x}^{IKF}(k|k - 1)) + w_{2}(k)[\max(C^{1} \hat{x}^{IKF}(k|k - 1)) -$$

$$- \min(C^{1} \hat{x}^{IKF}(k|k - 1))\} = z(k) - C \hat{x}^{KF}(k|k - 1) \quad (8.27)$$

Figure 8.8 Simulation of way-point tracking mission with imposed gyroscope bias, (a) actual heading, KF estimates, and fused KF estimates; (b) KF innovations; (c) SMA of KF innovations; (d) fuzzy weights assigned to each KF.
where \( z(k) \in \mathbb{R} \) is the measurement at time step \( k \), and \( \hat{x}^{IKF}(k|k-1) \) and \( \hat{x}^{KF}(k|k-1) \) the predictions by the IKF and KF, respectively, of the system state at time-step \( k \) given measurements up to time-step \( k-1 \). Then \( w_2(k) \approx 1 - w(k) \).

Based upon this conjecture, the innovations of the wIKF calculated as

\[
z(k) = \{ \min(C_1 \hat{x}^{IKF}(k|k-1)) + (1 - w(k))[\max(C_1 \hat{x}^{IKF}(k|k-1)) - \min(C_1 \hat{x}^{IKF}(k|k-1))] \}
\]  

with \( w(k) \) obtained as the ANN prediction of the weight in Equation 8.25, should approximate an innovations of an ideal KF, making the fuzzy fusion algorithm applicable to fuse wIKF estimates.

For the previously described simulated way point tracking mission, three wIKFs, constructed as described in Section 8.2, were simulated to combine gyroscope and compass data, using the same values of \( b_{min}, b_{max} \), and the same trained ANN used therein. Each wIKF is associated with a single compass, and initialised with \( \hat{x}^{wIKF}_{i}(0) = \theta(0), \text{var} \left( x - \hat{x}^{wIKF}_{i}(0) \right) = 0 \). Figure 8.9 depicts the simulation results: Figure 8.9a compares the actual vehicle’s heading to those obtained from each wIKF, as well as the fused wIKF estimate. The innovations of each wIKF, and the SMA of these, are shown in Figures 8.9b and 8.9c respectively. It can be seen how the innovations, defined as in Equation 8.28, are mostly comprised of white noise sequences prior to compass failure, except for small transient periods after a sudden change in gyroscope bias, as the ANN weight prediction requires time to adapt to the new dynamics. However, these transients are common to all three wIKFs and so none of them are discriminated during the same. However, after compass failure, the innovations deviate substantially from the ideal, especially during sharp turning manoeuvres, and it is these deviations that result in the fusion algorithm penalising the corresponding wIKF weights, as shown in Figure 8.9d.
(a) Actual heading, WiKF estimates, and fused WiKF estimates.

(b) WiKF innovations.

(c) SMA of WiKF innovations.
Fuzzy weights assigned to each wIKF.

Figure 8.9 Simulation of way-point tracking mission with imposed gyroscope bias, (a) actual heading, wIKF estimates, and fused wIKF estimates; (b) wIKF innovations; (c) SMA of wIKF innovations; (d) fuzzy weights assigned to each wIKF.

The RMS heading errors of the KFs and fused KF estimate shown in Figure 8.8, together with those of the wIKFs and fused wIKF estimate of Figure 8.9, are given in Table 8.4. From both the figures and the table, it is seen that the fused wIKF estimate is able to provide both a fault tolerant and robust heading estimate in the face of sporadic compass failure and gyroscope bias drifts.

Table 8.4. Heading RMS errors.

<table>
<thead>
<tr>
<th>Method</th>
<th>Heading RMS error (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>KF₁</td>
<td>120.37</td>
</tr>
<tr>
<td>KF₂</td>
<td>57.55</td>
</tr>
<tr>
<td>KF₃</td>
<td>145.92</td>
</tr>
<tr>
<td>Fused KF estimate</td>
<td>115.64</td>
</tr>
<tr>
<td>wIKF₁</td>
<td>115.36</td>
</tr>
<tr>
<td>wIKF₂</td>
<td>1.54</td>
</tr>
<tr>
<td>wIKF₃</td>
<td>147.78</td>
</tr>
<tr>
<td>Fused wIKF estimate</td>
<td>6.49</td>
</tr>
</tbody>
</table>

8.4 Summary

This chapter detailed the design of a multi-sensor data-fusion algorithm for fusing data from various KFs associated with different compass units in order to
detect compass failure and penalise the corresponding KF’s contribution to the fused estimate. Although only one type of compass failure was shown in which the reading remained frozen after a certain time period, this is in fact quite a subtle kind of fault to detect – the algorithm would work equally well (or better) to detect faults such as a zero reading from the compass. Simulations showed that the algorithm allowed the vehicle to continue successful autonomous operation even when all but one of its compass units failed to provide correct readings. Such capability is referred to as fault tolerance. Moreover, the algorithm can easily be extended to more or even other types of sensors, and applied in general where sensor redundancy exists.

In order to provide a degree of robustness to system modelling uncertainty, in this case caused by a drifting gyroscope bias, the wIKF was proposed as a solution. When the innovations of the wIKF are appropriately defined, then the fault tolerant fusion algorithm can be applied to fuse wIKF estimates. This confers the heading estimation subsystem with both robustness as well as fault tolerance. The importance of both these qualities was demonstrated in a simulated mission, allowing the vehicle to successfully complete its mission.
Chapter 9

Experimental Verification

“You can never cross the ocean until you have the courage to lose sight of the shore.” — Christopher Columbus

Previous chapters have focused on the development of various navigational algorithms and their verification through simulation studies. This chapter describes the setup used to conduct experimental trials for testing these techniques on the Springer USV and the results therefrom obtained.

9.1 Experimental Verification

Two sets of experimental trials relevant to the navigational algorithms described in this thesis were conducted with the Springer USV for estimating its heading. The first of these was carried out to test the wIKF that was described in Section 8.2 designed to be robust to an unknown but bounded gyroscope bias. The second set of trials tested the MSDF technique described in Section 8.3 for obtaining a fused wIKF heading estimate that is both robust to gyroscope bias and tolerant to compass failure.

9.1.1 Trials set-up and objectives

Both sets of trials described herein were conducted at Roadford Lake in the north of Devon, England (Figure 9.1). The man-made reservoir was chosen to conduct the trials in because of the availability of a launch and recovery facility for small boats such as Springer provided by the South West Lakes Trust’s Outdoor Plus Active Centre. In addition, the Centre also provided the Springer research team with a rigid-hulled inflatable boat (RHIB) so that they could navigate the lake and stay within range of Springer’s WiFi network in order to
be able to closely monitor the variables of each mission as it was being undertaken, as well as manually abort a mission when necessary (e.g. to avoid collision with curious surfers and other recreational sailing vessels).

In both sets of trials the missions were initiated from the tip of a jetty near the Outdoor Plus Active Centre, with the initial heading of the vehicle thus approximately constant and known. The GPS locations of a series of pre-
existing buoys were programmed into the vehicle’s guidance system to act as waypoints, as described in Section 4.1. However, unlike the waypoint coordinate description specified in Table 4.1, in practice the user specifies the mission waypoints by their latitudes and longitudes.

Assuming a spherical model of the Earth with radius $R_{\text{Earth}} = 6371$ km, then it is straightforward to derive that a change in one minute of a degree latitude corresponds to a distance of 1854 m. In addition, the latitude of the trials’ location (Roadford Lake) is approximately 50.698°, and the radius of that circle of latitude is $R_{\text{Roadford}} = R_{\text{Earth}} \cos(50.698°) = 4063.5$ km, from which it is easy to calculate that a change in one minute of a degree of longitude corresponds to a distance of 1177.5 m.

When the vehicle’s guidance system’s (Section 4.4) code is initialised prior to a mission, the current latitude and longitude of the vehicle indicated by its GPS receiver at that instant is stored as a reference GPS location in the computer memory. The guidance system then uses the above conversion factors to calculate the parallel and meridional distances of every waypoint subsequently specified by the user to this reference GPS location, and internally stores them as such. Thus, these can be thought of as points on an imaginary $xy$ grid whose axes measure distances, with the origin of the grid corresponding to the reference GPS location, and the $x$ axis pointing due east.

With respect to heading measurements, it has thus far not been noted that the compass units actually provide angles with respect to the Magnetic North rather than True North. The magnetic declination being approximately 2.4° at the trials location during the periods the experiments were conducted, the compass readings were offset by this amount before being used. Also, the angles measured are positive in the clockwise direction, whereas the convention used in this thesis is that angles are measured positive in the anticlockwise direction from the horizontal axis (Figure 9.2). A conversion is thus applied as follows

$$
\begin{align*}
\theta &= 90° - \theta_c, \text{ if } \theta_c \leq 90° \\
\theta &= 360° + 90° - \theta_c, \text{ if } 90° < \theta < 360°
\end{align*}
$$

(9.1)

to every compass measurement $\theta_c$. Additionally, because $\theta_c \in [0°,360°)$, so too according to Equation 9.1 $\theta$ will be restricted to the same interval. In practice, to avoid the circularity problem (abrupt change in angle value when moving between the first and fourth quadrants), the heading angle is unwrapped so that the domain is $(-\infty, \infty)$. 
This is done by keeping track of the number of complete turns carried out the vehicle, and adding or subtracting the corresponding multiple of $360^\circ$ to the value of $\theta$ obtained from Equation 9.1 at each sampling time. This then is the measured heading provided to the KF and wIKF, which similarly estimate an integrated heading angle, which is subsequently re-wrapped to the interval $[0^\circ, 360^\circ)$ for comparison with the reference angle generated by the guidance system, as already discussed in Section 4.6.

![Figure 9.2 Compass angle conversion](image)

The trials were carried out using a proportional controller with a fixed controller gain of 25, chosen based on trial and error. Because of the constant natural environmental disturbances, a higher gain was necessary than what was used previously in simulation to try and maintain the vehicle on course as much as possible.

The common mode propeller turning rate $n_c$ of the vehicle (Section 4.2) was maintained constant at 900 rpm to maintain a constant vehicle speed of approximately 3 knots. However, to allow a sharper turning radius, the value of $n_c$ was dropped to 450 rpm when within 20 m of a waypoint. The vehicle speed however was visibly affected by the direction of travel due to the wind conditions, and so the speed assumptions only constitute loose references. The wind speed recorded on the day was roughly 1 ms$^{-1}$ with gusts of up to 3 ms$^{-1}$ in a north/northwesterly direction.
These trials, carried out on 14 March 2014, aimed to demonstrate the robustness of the wIKF heading estimates in the face of a gyroscope bias. The wIKF is the one described in Section 8.2, fusing information from the vehicle’s gyroscope with that of one of its magnetic compass units (Table 3.3). Three pairs of trials were envisaged, each pair utilising one of the vehicle’s three compass units. Within each pair of trials, the first would use the wIKF heading estimate as feedback to the guidance and control of the vehicle, whereas the second would rely on the estimates of an ordinary KF. However, because of a hardware communications problem with the TCM2 compass, the HMR 3000 was used for two pairs of trials instead, whilst the KVH C100 for the last pair.

The mission established consisted of five waypoints which corresponded to the locations of four physical buoys that were identified on the lake in the region near the launch point (Figures 9.4a and 9.4b).

Table 9.1 summarises the premise of each experiment. With intention, the gyroscope was not precalibrated and presented a bias of between -3 to -4 deg/s all through the trials. The experiment results are shown graphically in Figures 9.5 to 9.16, and for each case include the actual path taken by the vehicle (plotted as consecutive locations obtained by the GPS readings at each sampling time), the gyroscope readings, compass measurements, KF and wIKF heading estimates. For completeness the corresponding control action is also plotted.
(a) Mission plan drawn on a satellite image of an area of the lake.

(b) Waypoints drawn on a rectangular grid according to their distances from the initial vehicle location.

Figure 9.4 Establishment of the mission plan waypoints.
Table 9.1 Summary of experiments.

<table>
<thead>
<tr>
<th>Experiment No.</th>
<th>Heading Estimation Method</th>
<th>Compass used</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1a</td>
<td>wIKF</td>
<td>HMR 3000</td>
</tr>
<tr>
<td>1.1b</td>
<td>KF</td>
<td>HMR 3000</td>
</tr>
<tr>
<td>1.2a</td>
<td>wIKF</td>
<td>HMR 3000</td>
</tr>
<tr>
<td>1.2b</td>
<td>KF</td>
<td>HMR 3000</td>
</tr>
<tr>
<td>1.3a</td>
<td>wIKF</td>
<td>KVH C100</td>
</tr>
<tr>
<td>1.3b</td>
<td>KF</td>
<td>KVH C100</td>
</tr>
</tbody>
</table>

Experiment 1.1a

Figure 9.5 Experiment 1.1a: actual path taken by the vehicle.
Figure 9.6 Experiment 1.1a data: (a) gyroscope readings; (b) compass measurement and KF and wIKF heading estimates; (c) autopilot output.

Figure 9.5 shows the trajectory followed by the vehicle when using the wIKF estimates for its feedback control, based on the HMR 3000 compass readings. The initial tendency of the vehicle was to stray towards port – this due to the fact that the gyroscope sustained a negative bias, which caused the wIKF to initially underestimate the vehicle’s heading, compelling the autopilot to steer towards port. This initial underestimation of the wIKF is seen in Figure 9.6b,
where it is also observed that unlike the KF estimate, which remains biased all throughout the trial, the wIKF was able to correct itself after an initial transient period of about 40 s. This circumstance and the short distance between way points 1 and 2 also explains the vehicle’s overshoot around these points, as the speed was maintained constant throughout. It should also be noted that the trajectory is significantly smoother on the Eastern side, as this area of the lake is comparatively more sheltered from wind and currents, to which the vehicle is quite sensitive given the thrust limitations of its trolling motors. During the smooth part of the trajectory, where the heading of the vehicle remains fairly constant, the gyroscope’s bias of between -3 to -4 deg/s can be seen in Figure 9.6a (between 700 and 800 s).

**Experiment 1.1b**

![Figure 9.7 Experiment 1.1b: actual path taken by the vehicle.](image-url)
In contrast, the trajectory of the vehicle when using the KF heading estimates, shown in Figure 9.7, appears chaotic. The KF bias (shown in Figure 9.8b) is present throughout the whole trial, substantially underestimating the vehicle’s heading, causing it to continuously steer towards a target port of each way point, explaining the bow-shaped segments between each pair of way points.
Experiment 1.2a

Experiment 1.2 was a repeat of the previous, and the results are qualitatively similar. The shape of the trajectory shown in Figure 9.9 is similar to that of Figure 9.5, albeit somewhat smoother, due solely to the slightly calmer conditions during this trial. However, the effect of the initial underestimation of the heading by the wIKF (Figure 9.14b) is still apparent.

Figure 9.9 Experiment 1.2a: actual path taken by the vehicle.
Figure 9.10 Experiment 1.2a data: (a) gyroscope readings; (b) compass measurement and KF and wIKF heading estimates; (c) autopilot output.
Experiment 1.2b

Once again, using the KF estimates as feedback to the control subsystem, the vehicle was not able to follow the ideal path. Underestimation of the heading caused the vehicle to target a point to the left of the actual target at all times, resulting in the path shown in Figure 9.11.

Figure 9.11 Experiment 1.2b: actual path taken by the vehicle.
Figure 9.12 Experiment 1.2b data: (a) gyroscope readings; (b) compass measurement and KF and wIKF heading estimates; (c) autopilot output.
Experiment 1.3a

The final experiment to validate the wIKF involved using the readings from the KVH C100 compass unit. As in the previous two experiments, the wIKF used to fuse compass and gyroscope data was able to recover from an initial underestimation of the heading caused by the gyroscope bias, which is unknown to the filter. The path followed and navigational and control data are shown in Figures 9.13 and 9.14 respectively.
Figure 9.14 Experiment 1.3a data: (a) gyroscope readings; (b) compass measurement and KF and wIKF heading estimates; (c) autopilot output.
Experiment 1.3b

Figure 9.15 Experiment 1.3b: actual path taken by the vehicle.
Figure 9.16 Experiment 1.3b data: (a) gyroscope readings; (b) compass measurement and KF and wIKF heading estimates; (c) autopilot output.

In contrast, the KF-fused compass and gyroscope heading estimates remained biased, leading to a poor performance of the system for carrying out the tracking mission, as evidenced in Figures 9.13 and 9.14.
Table 9.2 gives a summary of mission-related data. The results are consistent across the three pairs of trials. It is clear that the vehicle struggled significantly to complete the mission when the KF was used – in fact, only managing to reach within the vicinity of each waypoint after undertaking a significant detour. The effect of the negative gyroscope bias is clearly observable, as the KF consistently underestimates the vehicle’s heading angle, which results in the vehicle trying to correct its perceived heading by steering towards its port side. This effect is considerably reduced in the case of the wIKF heading estimates being used, although it is still apparent in the somewhat bowed trajectory that the vehicle follows between two waypoints. However, the severity of the gyroscope bias present during these trials constitutes an extreme case used to put the wIKF to test, and in practice one would always precalibrate the gyroscope at least to within a certain tolerance, and subsequent gyroscope bias drifts would be small in comparison.

Table 9.2 Summary of experiment results

<table>
<thead>
<tr>
<th>Experiment No.</th>
<th>Heading Estimation Method</th>
<th>Compass used</th>
<th>No. Waypoints reached (of 5)</th>
<th>Total Distance Travelled (m)</th>
<th>Total Time Taken (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1a</td>
<td>wIKF</td>
<td>HMR 3000</td>
<td>3</td>
<td>1270</td>
<td>14.2</td>
</tr>
<tr>
<td>1.1b</td>
<td>KF</td>
<td>HMR 3000</td>
<td>0</td>
<td>1727</td>
<td>22.6</td>
</tr>
<tr>
<td>1.2a</td>
<td>wIKF</td>
<td>HMR 3000</td>
<td>5</td>
<td>1260</td>
<td>14.9</td>
</tr>
<tr>
<td>1.2b</td>
<td>KF</td>
<td>HMR 3000</td>
<td>0</td>
<td>1726</td>
<td>21.5</td>
</tr>
<tr>
<td>1.3a</td>
<td>wIKF</td>
<td>KVH C100</td>
<td>4</td>
<td>1262</td>
<td>13.6</td>
</tr>
<tr>
<td>1.3b</td>
<td>KF</td>
<td>KVH C100</td>
<td>0</td>
<td>1771</td>
<td>21.0</td>
</tr>
</tbody>
</table>

9.3 Trial set 2: Multi-Sensor Data Fusion

Trials to test out the proposed fault-tolerant navigation system (Chapter 8) were conducted with Springer at Roadford Lake on 2 July 2014, using the gyroscope and three compass units described in Table 3.3. The mission, consisting of four way-points (three physical buoys), is shown in Figure 9.17. As in the previous experiments, the GPS coordinates of the buoys were obtained prior to the trials, and programmed into the vehicle’s guidance system. Also, as before, the starting point of each the trials was located at the tip of the jetty for consistency.
A total of four trials were carried out to validate the effectiveness of the proposed MSDF algorithm. In each, the estimated heading of the vehicle used by the autopilot was obtained by the fuzzy-fusion of the three individual wIKF estimates, as detailed in Section 8.3. In a first, reference experiment, used as benchmark, none of the compasses were made to fail. However, in the subsequent experiments, two of the three compasses were made to fail at different stages of the mission. As before, the gyroscope was not precalibrated, and suffered from a bias of roughly -3.5 deg/s. Wind speed was also monitored during the trials and averaged from 1 to 2 ms$^{-1}$ in a north-westerly direction, with gusts of up to 5 ms$^{-1}$.

**Reference Experiment**

Figure 9.18 shows the actual trajectory taken by the vehicle. Note again that because of the negative gyroscope bias, the trajectories are somewhat bow-shaped rather than straight, although the mission was completed successfully by virtue of the robustness of the wIKF heading estimation algorithm used.
Experiment 2.1 (TCM2 fail @ 182 s, HMR3000 fail @ 472 s)

In this experiment, the readings of the TCM2 were frozen at $k = 182$ s, roughly halfway to the first way point, whereas those of the HMR3000 were frozen at $k = 472$ s, approximately 50 m before reaching the second. Figure 9.19 depicts the actual trajectory followed by the vehicle, as well as the locations of the vehicle when the respective compass faults were provoked. The gyroscope readings are shown in Figure 9.20a, in which its strong bias is apparent. The compass readings are shown in Figure 9.20b, which clearly shows the two compass failures as their data subsequently remains constant. The individual gyro-compass wiIKF heading estimates, labelled 1 to 3 for TCM2, HMR300 and KVH C100, respectively, are shown in Figure 9.20c, along with the fused wiIKF. For completeness, the innovations of each wiKF, SMA values, and fuzzy weights are shown in Figures 9.20d to 9.20f respectively.

The TCM2 compass fault was provoked about halfway between the start and the first way point. Since, until reaching the first way point, no sharp manoeuvres were necessary, the innovations sequences of each of the wiIKFs do not deviate much from the ideal. As noted in Section 8.3, it is only upon
reaching the way point, where a sharp turning manoeuvre is necessary, that the discrepancy between gyroscope prediction and compass measurement becomes prominent. It is at such a point where the innovations of the filter associated with the failed compass deviates substantially from the ideal, and the fuzzy fusion algorithm is capable of detecting this anomaly and penalise the corresponding filter’s weights. In the case of the TCM2 failure, this can clearly be seen in Figure 9.20d, where the innovations cease to correspond to white noise circa $k = 300$, which is when the vehicle approaches the first way point. This translates into a sharp rise in the magnitude of the pertaining SMA (Figure 9.20e), and a sharp rejection of the filter (Figure 9.20f). In contrast, the second fault (HMR 3000 compass) occurs very near the second way point, and as a consequence, the fuzzy algorithm rejects the corresponding wIKF almost immediately (approximately $k = 500$, as seen in Figure 9.20e).

Table 9.3 benchmarks the overall estimates from the various wIKFs as well as the fused wIKF against that of the wIKF corresponding to the KVH C100 compass (wIKF3), as this was the only compass that was not made to fail. It should be noted that the error of the fused wIKF was of transient nature,
(a) Gyroscope readings.

(b) Compass measurements.

(c) wIKF and fused wIKF estimates.
Figure 9.20 Trial outcome 1: (a) gyroscope measurements; (b) compass measurements; (c) wIKF estimates and fused wIKF estimate; (d) wIKF innovations; (e) SMA of each wIKF; (f) fuzzy weights assigned to each wIKF.
occurring during the periods following the respective compass failures, but recovering in due course (Figure 9.20c), whereas the errors shown in Table 9.3 are relative to the length of the mission, and the fused wIKF RMS error in particular would therefore tend to zero as the mission length increased.

Table 9.3 Experiment 2.1 results.

<table>
<thead>
<tr>
<th>Method</th>
<th>Heading RMS error (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>wIKF₁ (gyro – TCM2)</td>
<td>160.4</td>
</tr>
<tr>
<td>wIKF₂ (gyro – HMR 3000)</td>
<td>156.1</td>
</tr>
<tr>
<td>wIKF₃ (gyro – KVH C100)</td>
<td>0.0</td>
</tr>
<tr>
<td>Fused wIKFs</td>
<td>16.5</td>
</tr>
</tbody>
</table>

**Experiment 2.2** (TCM2 and HMR 3000 fail simultaneously @ 250 s)

In this experiment, the TCM2 and HMR3000 were made to fail simultaneously at $k = 250$ s, just prior to reaching the first waypoint. The results are depicted graphically in Figures 9.21 and 9.22, and heading errors shown in Table 9.4.

Figure 9.21 Trial outcome 2: actual vehicle trajectory.
(a) Gyroscope readings.

(b) Compass measurements.

(c) wIKF and fused wIKF estimates.
Figure 9.22 Trial outcome 2: (a) gyroscope measurements; (b) compass measurements; (c) wIKF estimates and fused wIKF estimate; (d) wIKF innovations; (e) SMA of each wIKF; (f) fuzzy weights assigned to each wIKF.
Table 9.4. Experiment 2.2 results.

<table>
<thead>
<tr>
<th>Method</th>
<th>Heading RMS error (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(w\text{IKF}_1) (gyro – TCM2)</td>
<td>95.1</td>
</tr>
<tr>
<td>(w\text{IKF}_2) (gyro – HMR 3000)</td>
<td>85.4</td>
</tr>
<tr>
<td>(w\text{IKF}_3) (gyro – KVH C100)</td>
<td>0.0</td>
</tr>
<tr>
<td>Fused wIKFs</td>
<td>10.4</td>
</tr>
</tbody>
</table>

The simultaneous failure of two compasses was purposefully devised to challenge the fuzzy fusion algorithm. However, although the immediate disruption was more pronounced (notice that just after the failure, the vehicle’s heading is substantially off the target), the two flawed wIKFs were almost completely rejected within 50 s of the failures, and completely soon after the vehicle reached the third way point (circa \(k = 540\)), as seen in Figure 9.22f. However, because they were not rejected completely until this point, the trajectory of the vehicle between way points 1 and 2 remained somewhat overly bowed, even though the target was not missed. Once the two offending filters were fully rejected, the vehicle completed the mission in an almost ideal manner.

\textbf{Experiment 2.3} (TCM2 fail @ 153 s, HMR3000 fail @ 434 s)

In this final experiment, the TCM2 reading was frozen at \(k = 153\) s and that of the HMR3000 at \(k = 434\) s. Although somewhat of a replica of the situation of Experiment 2.1, the similar qualitative and quantitative results obtained (Figures 9.23 and 9.24, and Table 9.5) indicate a consistency in the same, and serves to demonstrate the high degree of repeatability of the tests under similar conditions and build confidence in the reliability of the outcomes.

Note though that in this case, the TCM2 wIKF was not completely rejected until the vehicle reached way point 2, although its weight was already considerably reduced upon reaching way point 1. Although this dependency to some extent of the fusion algorithm’s performance on turning manoeuvre requirements, it should also be stressed that it is not until such occasions that accurate heading estimates become very necessary.
Figure 9.23 Trial outcome 3: actual vehicle trajectory.

(a) Gyroscope readings.
(b) Compass measurements.

(c) wIKF and fused wIKF estimates.

(d) wIKF innovations.
Figure 9.24 Trial outcome 3: (a) gyroscope measurements; (b) compass measurements; (c) wIKF estimates and fused wIKF estimate; (d) wIKF innovations; (e) SMA of each wIKF; (f) fuzzy weights assigned to each wIKF.

Table 9.5. Experiment 2.3 results.

<table>
<thead>
<tr>
<th>Method</th>
<th>Heading RMS error (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>wIKF₁ (gyro – TCM2)</td>
<td>104.1</td>
</tr>
<tr>
<td>wIKF₂ (gyro – HMR 3000)</td>
<td>105.1</td>
</tr>
<tr>
<td>wIKF₃ (gyro – KVH C100)</td>
<td>0.0</td>
</tr>
<tr>
<td>Fused wIKFs</td>
<td>6.4</td>
</tr>
</tbody>
</table>
The results of all three experiments show that the MSDF algorithm was able to reject the faulty compasses completely within a limited period of time. The rejection is accelerated when the vehicle is forced to make sharp turning manoeuvres, as it is at these times when the discrepancy in the failed compass data with that of the gyroscope is most pronounced.

A similar mission was also attempted with the autopilot using feedback from fused ordinary KF estimates (as described in Section 8.1), but even without compass failure, the vehicle was unable to complete the mission. Thus the trial results corroborate the importance of an accurate heading estimation system for the vehicle to operate successfully, and that the methods proposed in this thesis achieve this, demonstrating the robustness to unaccounted for shifts in gyroscope bias as well as sporadic failure of up to two of the three available compasses.

9.4 Summary

In this chapter, the mechanism for experimental verification via real-time trials with the USV has been described. Each trial demands that the USV complete a way point tracking mission autonomously. In a first set of trials, the performances of the vehicle when using a KF and when using a wIKF for heading estimation were compared. In both cases, the gyroscope presented a bias that was not manually corrected for. Repeated tests showed that the mission was accomplished with far greater superiority when the wIKF heading estimates were used.

In a second set of trials, the MSDF algorithm of the previous chapter was put to the test for fusing three distinct wIKFs, each associated with one of the vehicle’s three compass units. In each trial, two of the three compass units were made to fail at different stages. Again, the gyroscope too presented a strong bias that was not calibrated against. The outcome of the experiments show that the MSDF algorithm is able to intelligently reject wIKFs associated with the failed compasses after a limited transient period, agreeing with the observed simulation results of Chapter 8. In all cases, the vehicle was able to complete the mission with relative ease in spite of the provoked hardware failures.
Chapter 10
Discussion, Conclusions, and Recommendations for Future Work

"Believe me, my young friend, there is nothing -- absolutely nothing -- half so much worth doing as simply messing about in boats." — Kenneth Grahame, Wind in The Willows.

In this chapter, the main aspects of the work described in this thesis are summarised and discussed, and proposals for further research are given.

10.1 Discussion and Conclusions

The focus of this thesis has been on the study and exploitation of the IKF as a robust alternative to the standard KF, and its concrete application to the heading estimation of the Springer USV. In Chapter 4, the advantage of applying Kalman filtering to obtain an improved estimate of the vehicle’s heading rather than relying on direct sensor measurement was highlighted. However, no sooner than this advantage was established, the KF’s biggest limitation was exposed at the beginning of the next chapter, namely, its reliance on a perfectly accurate dynamic model of the underlying process, as well as of the statistics of the random disturbances it is subject to. Needless to say, such an ideal situation is rarely likely in most cases and for most systems; in fact, modelling is often the most time consuming and costly aspect of systems engineering, and, in many cases, the least accurate or reliable. It is for this reason, for example, that the most commonly encountered control strategies
used in the industry are not optimal, model-based approaches, but algorithms such as the PID which do not rely heavily on having accurate system models and can be tuned to offer good performance for a wide range of system dynamics. It is also for this reason that the KF, so dependent upon an accurate specification of the system dynamics, is neither robust to outright uncertainties in the same or to variations in the system characteristics that may occur naturally over time, and constitutes a significant practical limitation to its use in a large variety of systems.

The issue of robustness in filtering algorithms to modelling uncertainty is a widely studied topic, and it is where the IKF finds it place as the most natural extension of the KF for uncertain systems. The theoretical framework of the IKF was developed by Chen et al. (1997) – it is actually not a modification of the KF equations themselves, but the interpretation of these in the domain of interval mathematics, where scalar elements cease to be single or point-valued, and instead are constituted by finite intervals. Once the conceptual principles of the KF were interpreted on the arithmetical and probabilistic framework of interval mathematics, the resulting IKF equations appear to be a stark copy of those of the KF, although of course, this is only so cosmetically, for the elements they operate on are different. The IKF works with interval elements, and as such, its state estimates are also given by intervals.

For systems with bounded modelling uncertainty, the IKF provides guaranteed bounds to the optimal estimate of the state vector. Even if the exact values of the states are not known, guaranteed bounds to these can be useful, for example, to ensure that they remain within some desirable or permissible operating region. The IKF was presented in Chapter 5 and applied to the problem of heading estimation. But in doing so, the major practical difficulty of obtaining meaningful estimates from the IKF was evidenced by the rapidly widening intervals, which not just contained, but were far wider than the actual solution sets.

Chapter 6 set out to investigate the main reason for the overestimation, namely the “wrapping effect”, and to develop a method to counter this effect. The solution was found in adopting another existing set-propagation arithmetic, the ellipsoidal arithmetic, much less prone to wrapping, by adapting the morphology of the IKF equations to be able to propagate its solutions via this arithmetic. This approach, in combination with a hybrid enclosure algorithm to fuse ellipsoids with rectangular sets in order to minimise wasteful expansion,
enabled the IKF estimates to be computed with far less over-estimation due to wrapping, providing tighter interval bounds that, crucially, do not diverge.

Once stable IKF estimate bounds could be obtained, for the purpose of control, the question of how to select the actual (point-valued) state of the (true) system from the interval estimate needed to be addressed. Of course, this had to be done without knowledge of the true system dynamics, which is the reason why the system model was described in terms of interval elements in the first place.

Chapter 7 addressed this problem by exploiting the machine learning capability of ANNs. Based on previously observed characteristics of IKF data for a particular estimation problem and uncertain system, and the true system state, the network could learn to identify a correlation between these variables that would later be used to make predictions. The result was the development of a wIKF, capable of selecting the adequate point-valued estimate from the IKF interval by observing and processing the residual behaviour of a selected KF, by means of a previously trained ANN. It should be emphasized that the training of the ANN did not require any more precise knowledge of the true system dynamics, and that the process could be carried out solely on simulated data.

The wIKF applied to the heading estimation of Springer was shown, in a simulation exercise in which an uncertain motion model of the vehicle was used, to provide accurate heading estimates in spite of the uncertainty in the system model, whose elements were specified in terms of finite intervals. These results were later replicated in full-scale empirical trials in which a wIKF was constructed to fuse gyroscope and compass data, but allowing the gyroscope to have an unknown (albeit bounded) bias.

In order to exploit the sensorial redundancy (three compass units) of the Springer USV, Chapter 8 was dedicated to the design of a MSDF algorithm for KFs to provide a fault tolerant heading estimation system that would provide accurate heading even when up to two of the three filters provided erroneous data due to hardware (compass) failure. The algorithm developed was then made compatible for use with the wIKF, resulting in a heading estimation system that is both robust to model uncertainty and fault tolerant. The MSDF of three wIKFs was implemented and tested in full scale trials with the Springer, demonstrating that the procedure was effective in practice.

It should be emphasised that though the wIKF and MSDF algorithms were implemented on the Springer platform, and of course customised to estimate heading, the procedures developed in this thesis are of general application. As
already argued, systems with uncertain dynamics are commonplace in many areas, and as long as the uncertainty can be bounded and they can be described via an interval state-space model, all the techniques described in this thesis can be applied to construct a wIKF capable of estimating the true system state. The immediate advantage over the use of a KF is thus made palpable: the wIKF requires only an interval model of the system dynamics, whereas the KF requires a precise model.

Regarding the application of the wIKF to the heading estimation of Springer carried out throughout thiswork, the KFs and wIKFs implemented in practice were designed to fuse gyroscope and compass data, as these are the sensors the vehicle is currently equipped with. However, other low-cost sensors such as a speed sensor (paddle-wheel) could be incorporated into the vehicle and considered in the data-fusion process. In addition, the wIKF could be used as well for GPS-IMU fusion for the localisation of the vehicle, as low cost MEMS-based IMUs are readily available.

A clear application of the wIKF could also be to allow specifying the sampling time of the process in terms of an interval rather than a point-value. This would ensure that the filter is robust to small variations in the sampling time. Because the algorithms developed in this work were implemented on a general purpose PC running a Windows operating system, the algorithm execution time was not guaranteed to always be 100% consistent and regular. Therefore, the effects of an inaccurate execution time could be minimised by describing the sampling time in the state-space model as a bounded interval and applying the wIKF technique developed in this thesis.

With regard to the trial results graphed in Section 9.3, careful observation reveals that the TCM2 compass was somewhat biased as compared to the other two units. The MSDF algorithm as proposed is not able to detect this anomaly. Compass bias might arise due to unknown magnetic declination, and if included in the system state space model, could be modelled as an interval value. However, an incorrect compass bias will not affect the innovations of a KF based on a vehicle steering model and/or gyroscope integration equation together with a compass measurement model, and so a different sensor would be required to achieve this. One option would be to be able to measure the components of the Earth’s magnetic field along two perpendicular directions (using, for example, two separate coils). The measurement models in this case would provide a KF whose innovation sequence would be sensitive to incorrect compass bias, and a wIKF would then provide the required robustness to this kind of uncertainty.
Hopefully these arguments bring to light some of the possible ways in which to exploit the wIKF and sensor fusion. It is hoped that the knowledge provided through this work, the demonstrated feasibility of the implementation described, and the visible effectiveness and advantage that the wIKF has to offer will see an expansion in its use for all sorts of systems.

10.2 Recommendations for Future Work

The following lists several suggestions for future work that due to time limitations were not possible to investigate further in the current work.

- Although seemingly computationally complex, the real-time trials have shown that it is feasible to implement the technique for systems with sampling times of the order of 1 Hz. Although computational optimisation was not a prerogative of the present work, the real-time implementation on dedicated hardware might be required for use on faster systems.

- In Chapter 6, the KF/IKF equations were written as two sets of recursive affine transformations in order to exploit the advantages of using ellipsoidal arithmetic. However, the Kalman gain was still calculated using regular interval arithmetic, in a way breaking the continuity of the recursive ellipsoidal transformations. If a method for incorporating the computation of the Kalman gain into the recursive affine transformation process could be found, this could lead to even tighter IKF estimate enclosures. Alternatively, since calculation of the Kalman gain requires the inversion of an interval valued matrix, techniques to carry out this interval matrix inversion with minimal pessimism should be explored, thus complementing the approach developed here.

- Although set propagation via ellipsoidal arithmetic was exploited in Chapter 6, other types of arithmetics for propagating n-dimensional sets exist, such as polytope arithmetic, and could be explored for use on the IKF.

- The mathematical development of the hybrid enclosure algorithm was presented for generic linear state space models of any order, and its application enables the practical and efficient use of the IKF as a guaranteed state estimator for all kinds of systems with bounded
parametric uncertainty that respond to the linear model. Another useful development would be the extension of the method to the nonlinear case (interval extended KF).

- In Chapter 7, as a proof of principle, a particular type of ANN was developed to provide a learning mechanism by which the wIKF weight could be automatically obtained. Although this innovative approach was applied successfully, it should be observed that the use of an ANN as a learning mechanism that can, once trained, be used to predict the adequate weightings of a wIKF is not the only tool that can be used. Other machine learning tools such as support vector machines may be able to perform similar tasks, and should be further investigated. Even within ANNs, different architectures to the one proposed here and alternative training methods that may be more suited to this type of problem are topics for further research.

- To carry out a detailed comparison of the IKF and wIKF methods developed in this thesis with other robust estimation methods, in particular with those described in Chapter 2 (guaranteed cost estimation approach, the set-valued approach, and $H_\infty$ filtering). Analyses to determine the advantages and disadvantages of each of these, including a comparison of the efficiency and computational cost, performance, and limitations of each method in order to develop general guidelines for selecting the most appropriate method based on the characteristics of the system and the structure of the uncertainty involved.
References


195


Department of the Navy, US. Available at www.navy.mil/navydata/technology/usvmppr.pdf [Accessed on 20 July 2011]


A. Bayesian Data Fusion

In the example given in Section 2.1.1, the random variable $X$ represented the heading of the ship $x$, and its probability density function (pdf) described by $f_X(x)$:

$$X \sim N(h, \sigma_X^2); \quad f_X(x) = \frac{1}{\sigma_X \sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma_X^2}(x-h)^2\right) \quad (A.1)$$

The pdf $f_X(x)$ represents the prior or initial belief about the ship’s heading, that is, before any subsequently obtained information is incorporated. The best (most likely, or minimum mean square error) estimate of the ship’s heading is given by $\hat{x} \equiv E(X) = h$ (see, for example, Papoulis (1991)), and can be demonstrated as follows. The mean square error of the estimate $h$ of the true value of the heading $x$ is given by

$$MSE = E[(X-h)^2] = E[X^2 - 2hx + h^2] = E[X^2] - 2hE[X] + h^2 \quad (A.2)$$

The value of $h$ that minimises this error is found by taking the derivative of the MSE with respect to $h$:

$$\frac{d}{dh} E[(X-h)^2] = -2E[X] + 2h \quad (A.3)$$

and equating this to zero

$$\frac{d}{dh} E[(X-h)^2] = 0 \iff h = E[X] \quad (A.4) \implies Q.E.D.$$ 

Hence the MSE of the estimate coincides with the variance of the distribution of $X$,

$$MSE = E[(X-h)^2] = E[(X-E[X])^2] = \text{var}(X) = \sigma_X^2 \quad (A.5)$$
The sailor then observes the North Star and infers from it that the ship’s heading is $z_1$, accurate to a value of $\sigma_{z_1}$ degrees RMS. This inference process can be thought of as a realisation of the conditional random variable $Z_1 | X = x$ with probability density function

$$Z_1 | (X = x) \sim N(x, \sigma^2_{z_1}); f_{Z_1|X=x}(z_1; x) = \frac{1}{\sigma_{z_1}\sqrt{2\pi}} \exp \left( - \frac{1}{2} \frac{1}{\sigma^2_{z_1}} (z_1 - x)^2 \right)$$ (A.6)

That is, $Z_1 | (X = x) = x + \Omega_1$, where $\Omega_1 \sim N(0, \sigma^2_{z_1})$. The MSE of the estimate $z_1$ is calculated as

$$MSE = E \left[ ((X - Z_1) | X = x)^2 \right] = E \left[ (x - (Z_1 | X = x))^2 \right] = \text{var}(Z_1 | X = x) = \sigma^2_{z_1}$$ (A.7)

In order to obtain a more accurate estimate based on both the prior belief and the value estimated from the observation of the star, the pdf of $X$ given the observation $z_1$, that is, pdf of the random variable $X | Z_1 = z_1$ can be calculated by applying Bayes’s theorem,

$$f_{X|Z_1=z_1}(x; z_1) = \frac{f_{Z_1|X=x}(z_1; x) f_X(x)}{f_{Z_1}(z_1)}$$ (A.8)

In order to calculate this, the pdf of $Z_1$ needs to be known. It can be seen that if $X$ and $\Omega$ are independent random variables, then $Z_1$ is also a normally distributed random variable,

$$Z_1 \sim N(h, \sigma^2_{z_1} + \sigma^2_{X})$$ (A.9)

Note that the variance of $Z_1$ includes the variance of $X$, as the pdf of $Z_1$ reflects the probability of the estimated heading based on observing the North Star for any possible heading of the ship and not the particular heading $x$, as $(Z_1 | X = x)$ does. Then, applying Equation A.8,

$$f_{X|Z_1=z_1}(x; z_1) = \frac{1}{\sqrt{2\pi} \sigma_X \sigma_{z_1}} \exp \left\{ - \frac{1}{2} \left[ \frac{(x-h)^2}{\sigma^2_X} + \frac{(z_1-x)^2}{\sigma^2_{z_1}} - \frac{(x_1-h)^2}{\sigma^2_{z_1} + \sigma^2_X} \right] \right\} =$$

$$\frac{1}{\sqrt{2\pi} \sigma_X \sigma_{z_1}} \exp \left\{ - \frac{1}{2} \left[ \frac{\sigma^2_X + \sigma^2_{z_1}}{\sigma^2_X \sigma_{z_1}} \left[ x - h + \frac{\sigma^2_X}{\sigma^2_X + \sigma^2_{z_1}} (z_1 - h) \right] \right]^2 \right\}$$ (A.10)
that is,

\[
\left( X \mid (Z_1 = z_1) \right) \sim N \left( h + \frac{\sigma^2}{\sigma^2 + \sigma_z^2} (z_1 - h), \frac{\sigma^2 \sigma_z^2}{\sigma^2 + \sigma_z^2} \right)
\]

(A.11)

The estimate \( \hat{x}_1 = E[X \mid (Z_1 = z_1)] = h + \frac{\sigma^2}{\sigma^2 + \sigma_z^2} (z_1 - h) \) is known as Bayes’s estimate of the random variable \( X \) given an observation \( Z_1 = z_1 \) of the same. It is apparent that it is a weighted average of the prior (blind) estimate, \( h \) and the measurement-derived value, \( z_1 \), inversely weighted by the respective variances of the corresponding pdfs. In effect, \( E[X \mid (Z_1 = z_1)] \) can be written as

\[
\hat{x}_1 = E[X \mid (Z_1 = z_1)] = h + \frac{\sigma^2}{\sigma^2 + \sigma_z^2} (z_1 - h) = \frac{\sigma^2_z}{\sigma^2 + \sigma_z^2} h + \frac{\sigma^2}{\sigma^2 + \sigma_z^2} z_1 \quad (A.12)
\]

Note that \( \sigma^2_z \equiv \text{var} [X \mid (Z_1 = z_1)] = \frac{\sigma^2 \sigma^2_z}{\sigma^2 + \sigma^2_z} = \frac{1}{\frac{1}{\sigma^2} + \frac{1}{\sigma_z^2}} < \sigma^2 \) and \( \sigma^2_z < \sigma^2_z \), and that the MSE of the estimate, which can be formulated as,

\[
\text{MSE} = E\left[ (X - E[X \mid (Z_1 = z_1)])^2 \right] = E_{Z_1}\left[ E_X \left( (X - E(X \mid (Z_1 = z_1)))^2 \right) \mid (Z_1 = z_1) \right] = E_{Z_1}\{\text{var}[X \mid (Z_1 = z_1)]\} = E_{Z_1} \left( \frac{\sigma^2 \sigma^2_z}{\sigma^2 + \sigma^2_z} \right) = \frac{\sigma^2 \sigma^2_z}{\sigma^2 + \sigma^2_z} = \sigma^2_z \quad (A.13)
\]

is equal to the variance of the pdf of \( (X \mid (Z_1 = z_1)) \). Also, Bayes’s estimate \( E[X \mid (Z_1 = z_1)] \) is the minimum MSE (MMSE) estimate of \( x \), since

\[
E(X - g(z_1))^2 \geq E\left( (X - E(X \mid Z_1 = z_1))^2 \right) \quad \text{for any function } g(z_1) \quad (A.14)
\]

which makes it an optimal estimator (Papoulis, 1991). Indeed, let \( X \) be a random variable that cannot be observed, and \( x \) a realisation of \( X \). If \( Y \) is an observable random variable that is correlated to \( X \), then the function \( g(Y) \) can be thought of as an estimator of \( x \). The MSE of this estimator is calculated as the average squared error over both \( X \) and \( Y \):

\[
\text{MSE} = E[(X - g(Y))^2] \quad (A.15)
\]
where the expectation is the joint expectation over $X$ and $Y$. Taking into account that $E(Z) = E_X[E_Z(Z|X)]$ (law of total probability), this can be written out as:

$$
\text{MSE} = E[(X - g(Y))^2] = E_Y\{E_X[(X - g(Y))^2|Y]\} = 
\int E_X[ (X - g(Y))^2 | (Y = y)] \, f_Y(y) \, dy
$$

(A.16)

In order to find the function $g(Y)$ for which the MSE is minimum, it suffices to minimise $E_X[ (X - g(Y))^2 | (Y = y)]$ with respect to $g$, since $f_Y(y) \geq 0$ and $E_X[ (X - g(Y))^2 | (Y = y)] \geq 0$. Since $Y$ is fixed at some value $y$, $g(Y)| (Y = y)$ is no longer a random variable, so the minimisation problem is reduced to the one previously described (A.2 to A.4),

$$
\frac{d}{dg} E_X[ (X - g(Y))^2 | (Y = y)] = 0 \Rightarrow g^*(y) = E[X| (Y = y)]
$$

(A.17)

Q.E.D.

This fusion process can be repeated to incorporate more information in a recursive fashion, in which the posterior becomes the new prior when a new observation is obtained. For instance, sighting Sirius, the sailor was able to estimate that the heading of the ship was $z_2$, with an RMS error of $\sigma_{z_2}$. As before, assuming that $z_2$ is a realisation of the random variable $Z_2 (X = x) = x + \Omega_2$, with $\Omega_2 \sim N(0, \sigma_{z_2}^2)$, then the pdf of the random variable $(X | (Z_1 = z_1))(Z_2 = z_2) = X|((Z_1, Z_2) = (z_1, z_2))$ is obtained from the prior (pdf of $X | (Z_1 = z_1)$) and the observation $Z_2 | (X = x)$ using Bayes's formula. Replicating the previous results,

$$
X|((Z_1, Z_2) = (z_1, z_2)) \sim N\left(\hat{x}_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_{z_2}^2} (z_2 - \hat{x}_1), \frac{\sigma_1^2 \sigma_{z_2}^2}{\sigma_1^2 + \sigma_{z_2}^2}\right)
$$

(A.18)

The optimal estimate based on this new posterior belief is

$$
\hat{x}_2 \equiv E[X|((Z_1, Z_2) = (z_1, z_2))] = \hat{x}_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_{z_2}^2} (z_2 - \hat{x}_1) = 
\frac{\sigma_{z_2}^2 \sigma_{z_1}^2}{\sigma_{z_1}^2 \sigma_{z_2}^2 + \sigma_{z_1}^2 \sigma_{z_2}^2 + \sigma_{z_1}^2 \sigma_{z_2}^2} Z_1 + \frac{\sigma_{z_2}^2 \sigma_{z_1}^2}{\sigma_{z_1}^2 \sigma_{z_2}^2 + \sigma_{z_1}^2 \sigma_{z_2}^2 + \sigma_{z_1}^2 \sigma_{z_2}^2} Z_2
$$

(A.19)

and the MSE of the estimate is
\[
\sigma_2^2 \equiv \text{var}[X|((Z_1, Z_2) = (z_1, z_2))] = \frac{\sigma_1^2 \sigma_z^2}{\sigma_1^2 + \sigma_z^2} = \frac{1}{\frac{1}{\sigma_1^2} + \frac{1}{\sigma_z^2}}
\]
(A.20)
B. ANN Training Procedure

The RMLP of Figure 7.3 has a hidden layer with five units, all of which incorporate hyperbolic tangent activation functions. Information is propagated forward through the network at each time step according to (B.1),

\[
\begin{align*}
a^{(1)} &= \left[ \frac{1}{x} \right] ; \\
a^{(2)} &= \left[ \tanh\left( \frac{1}{\Theta^{(1)}} a^{(1)} \right) \right] ; \\
\hat{w} &= a^{(3)} = \Theta^{(2)} a^{(2)} \quad \text{(B.1)}
\end{align*}
\]

where \(a^{(l)}\) are the outputs of the nodes of layer \((l)\), and \(\Theta^{(1)} \in \mathbb{R}^{5 \times 13}\) and \(\Theta^{(2)} \in \mathbb{R}^{1 \times 6}\) are the matrices of parameters of the network such that \(\Theta_{ij}^{(l)}\) represents the strength of the connection between node \(a_j^{(l)}\) and \(a_i^{(l+1)}\) (Figure 7.3).

Training the network consists of finding the parameters \(\Theta_{ij}^{(l)}\) that minimise the cost function:

\[
J = \frac{1}{m-5} \sum_{k=6}^{m} \frac{1}{2} (w_t(k) - \hat{w}(k))^2 \quad \text{(B.2)}
\]

\(m\) being the number of training samples. This process was carried out recursively via the gradient descent (GD) algorithm: after assigning random initial values to \(\Theta_{ij}^{(l)}\), the parameters are updated as

\[
\Theta_{ij}^{(l)} := \Theta_{ij}^{(l)} - \alpha \frac{\partial J}{\partial \Theta_{ij}^{(l)}} \quad \text{for all } \Theta_{ij}^{(l)} \quad \text{(B.3)}
\]

until convergence is reached, \(\alpha\) being the learning rate, chosen adequately based on trial and error. The gradient of the cost function with respect to the network’s parameters was computed using the back-propagation (BP) method:
**Back-propagation:**

for each training pattern \(x(k)(\text{Equation 7.3})\) and target \(w_t(k)\)

1) compute \(\hat{w}\) (Equation B.1)

2) \(\delta^{(3)} = w_t - \hat{w} \)

\[
\delta_i^{(2)} = \left[ 1 - \tanh^2 \left( \sum_j \theta_{ij}^{(1)} a_i^{(1)} \right) \right] \theta_i^{(2)} \delta^{(3)} ; \ i = 1, \ldots, 5
\]

\[
\Delta_i^{(2)} := \Delta_i^{(2)} - \delta^{(3)} a_i^{(2)} ; \ i = 0, \ldots, 5
\]

\[
\Delta_{ij}^{(1)} := \Delta_{ij}^{(1)} - \delta_i^{(2)} a_j^{(1)} ; \ i = 1, \ldots, 5; \ j = 0, \ldots, 12 \quad (B.4)
\]

end

\[
\frac{\partial J}{\partial \theta_{ij}^{(l)}} = \frac{1}{m} \Delta_{ij}^{(l)} \quad (B.5)
\]

The GD process was applied in two stages, depending on how the gradient was calculated. During the first set of iterations, the (delayed) target values \(w_t\) were used to construct \(x(k)\) for computation of \(\hat{w}\) in the first step of the BP process, effectively training a network without feedback. During a second stage, (past) predictions of the network \(\hat{w}\) were used to construct \(x(k)\) in accordance with the true feedback architecture of the network. Although the gradients computed with the BP algorithm in this case are approximations to the true gradient, the errors are small as after the first set of iterations the network is sufficiently trained to output predictions close to the target values.
C. Selected Publications


**Interval Kalman filtering in navigation system design for an uninhabited surface vehicle.**


© Cambridge University Press. Reproduced with permission.
Interval Kalman Filtering in Navigation System Design for an Uninhabited Surface Vehicle

A Motwani¹, SK Sharma¹, R Sutton¹ and P Culverhouse²

¹(School of Marine Science and Engineering, Plymouth University, Plymouth, UK)
²(School of Computing and Mathematics, Plymouth University, Plymouth, UK)
(E-mail: amit.motwani@plymouth.ac.uk)

This paper reports on the potential application of interval Kalman filtering techniques in the design of a navigation system for an uninhabited surface vehicle named Springer. The interval Kalman filter (IKF) is investigated for this task since it has had limited exposure for such usage. A state-space model of the Springer steering dynamics is used to provide a framework for the application of the Kalman filter (KF) and IKF algorithms for estimating the heading angle of the vessel under erroneous modelling assumptions. Simulations reveal several characteristics of the IKF, which are then discussed, and a review of the work undertaken to date presented and explained in the light of these characteristics, with suggestions on potential future improvements.

KEY WORDS


1. INTRODUCTION. In a review of uninhabited surface vehicles (USVs) (also referred to as unmanned surface vehicles) by Motwani (2012), it can clearly be seen that such craft are now being used in a plethora of marine related applications. For an example, in June 2011, the Ministry of Land, Transport and Maritime Affairs, South Korea, announced a four year USV development programme worth a total of $18.5 million (Martin, 2012). Missions envisaged include surveillance, research and monitoring of the oceans. In particular, these vehicles will be used for exclusive economic zone protection duties with a remit for policing illegal fishing and military border intrusions by North Korea.

Irrespective of its allotted task, however, all USVs have two common features. Firstly, such vehicles require robust navigation, guidance and control systems in order to cope with possible changes in the dynamic behaviour of a vehicle which may occur owing to the deployment of different payloads, amendments to mission requirements and varying environmental conditions; and secondly, unlike large commercial ships...
and warships that are equipped with high specification navigational aids such as radio beacons, radar, gyroscopic compasses and inertial measurement units (IMUs), the navigation suites for USVs are invariably low-cost.

As reported by Sharma et al. (2012), in recent years at Plymouth University, the Springer USV has been designed, built, and continues to be developed. The work presented was concerned with the development of a sophisticated autopilot for Springer, whereas this paper explores the potential application of interval Kalman filtering (IKF) in the design of a navigation system for the vehicle. The investigation was instigated as the IKF has received minimal attention for application in such roles, even though, as will be discussed, its fundamental quality of being able to compute rigorous bounds to the optimal estimate can be advantageous when the model parameters themselves are uncertain.

The structure of this paper is as follows: section 2 details the state-space yaw model of the Springer used throughout this paper; section 3 then briefly discusses traditional Kalman filtering (KF), with particular attention to its use in surface navigation systems, and evidences its limitations when the system model is imprecise; an introduction to IKFs is given in section 4, and its application to the Springer yaw model shown in section 5, from which certain characteristics are gleaned leading to a discussion and review of techniques being developed with regard to these. Concluding remarks are in section 6.

2. THE SPRINGER YAW DYNAMICS AND NAVIGATION SUITE. The yaw dynamics of the Springer USV can be represented as a second-order state-space model by Naeem et al. (2008) as shown below. The propulsion system is based on two trolling motors, one situated on each hull. If \( n_1 \) and \( n_2 \) represent the rpm of each motor (Figure 1), then while \( n_1 = n_2 \) the vessel moves in a straight line, whereas differences in the two revolution rates enable the vessel to steer.

Defining \( n_c \) and \( n_d \) as

\[
\begin{align*}
  n_c &= \frac{n_1 + n_2}{2} \\
  n_d &= \frac{n_1 - n_2}{2}
\end{align*}
\]

(1)

it is clear that steering is controlled by varying \( n_d \) while keeping \( n_c \) constant. Several trials have been carried out at Roadford Reservoir in Devon, UK, in which the vehicle was driven for some calculated manoeuvres, maintaining a constant \( n_c \) of 900 rpm. During these trials, both the differential thrust applied to the motors, \( n_{dh} \) and the heading angle of the vehicle, obtained from on board compasses, were recorded. System identification (SI) techniques were then applied and the following state-space model of
the yaw dynamics was obtained and validated (Naeem et al. 2008):

\[
x(k + 1) = A x(k) + B u(k) \tag{2}
\]

\[
y(k) = C x(k) \tag{3}
\]

with

\[
A = \begin{bmatrix} 1.002 & 0 \\ 0 & 0.9945 \end{bmatrix}, \quad B = \begin{bmatrix} 6.354 \\ 4.699 \end{bmatrix} \times 10^{-6}, \quad C = \begin{bmatrix} 34.13 & 15.11 \end{bmatrix} \tag{4}
\]

and a sampling time of 1 s, where \(u(k)\) represents the differential thrust input in rpm and \(y(k)\) the heading angle in radians.

While full details of the Springer's hardware can be found in the Journal of Navigation (Sharma et al., 2012), for the purposes of this paper it is sufficient to mention that the sensor suite combines a Global Positioning System (GPS), three different types of digital compasses, a speed log, and a depth sensor, interfaced to a PC on board the vessel via a NI-PCI 8430/8 (RS232) serial connector.

3. A BRIEF OVERVIEW OF KALMAN FILTERING IN USV SURFACE NAVIGATION. Since its inception in 1960 and its initial application to spacecraft navigation, the KF has been used extensively in innumerable applications. While a detailed study of the KF can be found in Chui and Chen (2008), the main results are summarised in Appendix A.

The algorithm’s inherent structure allows it to naturally combine measurements from various sensors, weighing their respective precisions. This has prompted the use of the KF as a tool for fusing data from low-cost sensors to obtain synergistically highly reliable estimates that would otherwise require more precise sensors. Whereas in spacecraft attitude estimations, KFs are developed for highly accurate inertial sensors, multi-sensor data-fusion has been prominent in the case of land and sea surface-vehicle navigation. Particularly, the integration of low-cost inertial navigation systems (INS) with GPS localisation data has been a commonly adopted strategy in these vehicle types. For example, the USV being developed at Virginia Tech University (VaCAS, 2011), uses differential GPS (DGPS) together with a puck-sized micro-electro-mechanical system (MEMS) based inertial sensor offered by MicroStrain Inc that is popularly used in mobile robotic applications (MicroStrain, 2012).

Others have used KFs to combine GPS, IMU, as well as magnetic compass sensor readings. For example, Zhang et al. (2005) describe the use of an unscented KF (UKF) to combine a low-cost IMU, GPS and digital compass using a sophisticated dynamical model of the USV. Others have successfully implemented KF-based USV navigation without IMUs altogether. In the case of Springer (section 2), data from the digital compasses are combined using various data-fusion architectures based on KFs (Xu, 2007). The use of redundant data (by using three separate compasses simultaneously) allows for the construction of fault-tolerant navigation systems. Successful sea trials have demonstrated the effectiveness of the navigation systems of Springer. Another example is the USV Charlie, equipped solely with GPS and magnetic compass, which uses an extended KF (EKF) (Caccia et al., 2008). The reader interested in the workings of the UKF and EKF may consult Aich and Madhumita (2010).
The basic KF scheme yields a statistically optimal estimate only for linear systems with white Gaussian system and measurement noises with known covariances, when an accurate description of the deterministic matrices, the estimated initial state, and its covariance are available. When these hypotheses are not met, the effectiveness of the KF can be compromised. As an example, consider a computer simulation of a KF estimate of the Springer heading for which the previously described Springer yaw dynamics model (Equations 2 to 4) is used for the predictive stage of the filter, while measurements of the heading angle are obtained directly from one of the compasses and used in the corrective stage. Although dynamic models for each of the compasses were obtained by Xu (2007), since these are several orders of magnitude faster than the Springer yaw dynamics, in practice they can be assumed to provide the instantaneous heading of the vehicle, to within a certain accuracy and precision. Assuming that the system state is affected by a random input disturbance that follows a zero-mean white Gaussian noise sequence with covariance matrix \( \mathbf{Q} = \text{diag}\{1,1\} \times 10^{-9} \), and that compass readings provide accurate but imprecise measurements that likewise can be described as a white Gaussian noise sequence, with mean equal to the true heading, in degrees, and a standard deviation of 2°, the state and measurement models for the yaw dynamics of Springer are given by:

\[
\mathbf{x}(k+1) = \mathbf{A} \mathbf{x}(k) + \mathbf{B} \mathbf{u}(k) + \omega(k) \tag{5}
\]

and

\[
\mathbf{y}(k) = \mathbf{C} \mathbf{x}(k) + \nu(k) \tag{6}
\]

with

\[
\mathbf{A} = \begin{bmatrix} 1.002 & 0 \\ 0 & 0.9945 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 6.354 \\ -4.699 \end{bmatrix} \times 10^{-6}, \quad \mathbf{C} = \begin{bmatrix} 360/2\pi & 34.13 & 15.11 \end{bmatrix}
\]

\[
\text{cov}(\omega) = \text{diag}\{1, 1\} \times 10^{-9}, \quad \sqrt{\text{var}(\nu)} = 2^\circ
\]

\( \mathbf{y}(k) \) being the compass measurement in degrees. In order to illustrate the effect of deficient system modelling in the quality of the KF estimate, consider that the actual Springer yaw dynamics differ slightly from this model: specifically, that the actual value of the first component of the \( \mathbf{A} \) matrix is 1% less than the modelled value, all other values being accurately modelled. Then, based on simulated values of \( \omega(k) \) and \( \nu(k) \) and the input values of \( \mathbf{u}(k) \) shown in Figure 2a, the KF estimate obtained is shown against the true heading in Figure 2b. For the sake of accentuating the effect of the erroneous modelling in the KF estimate, a second KF estimate is shown for which the true model of the system was assumed.

The traditional KF relies on known statistical models (\( \mathbf{Q} = \text{cov}(\omega) \), \( \mathbf{R} = \text{cov}(\nu) \)) to describe uncertainty. However, as suggested earlier, these a priori statistics may not be known accurately, or be changing over time. For example, GPS accuracy is affected by the positions of the satellites, interference of the radio signal, physical barriers to the signal like mountains or those due to atmospheric conditions. Compass readings may be affected by interfering electromagnetic fields. In this scenario, the measurement noise covariance \( \mathbf{R} \) must be continuously adapted to accommodate the changes in sensor accuracy in order to achieve good performance from the KF. Likewise, the varying sea roughness and other random effects that affect the system should ideally be contemplated in a varying system noise covariance \( \mathbf{Q} \) that most accurately reflects.
the conditions at each moment. The KF also assumes that the deterministic dynamics described by matrices $A$, $B$ and $C$ are modelled precisely; however, changing payload, mean wind speeds or current forces, etc would translate into slowly-varying dynamics that would require the model to be continuously updated. While there are many techniques which try to increase the KF’s reliability by incorporating adaptive mechanisms, such as online modification of the covariance matrices based on fuzzy logic (Abdelnour et al., 1993; Xu et al., 2006), such techniques cannot in general guarantee the virtue of the estimates in all situations, or provide rigorous error bounds.

4. INTERVAL KALMAN FILTERING. Interval analysis is a field of mathematics that began to be formally studied in the 1950s with the intention of finding a way to bound rounding errors in finite precision numerical computations. As computer representation is limited by the machine epsilon, only a small subset of real numbers can be represented accurately on a computer, but every real number can be represented by an enclosure consisting of two bounding, machine-representable values, by appropriate upward and downward rounding. Consequently, an interval arithmetic (IA) was defined (Moore, 1966) in such a way that calculations on interval values yield further intervals which guarantee to enclose (though not necessarily be equal to) the range of possible results, a fundamental quality known as the inclusion property. Hence, if an initial bound for the error is known, then the error propagation inherent in any calculation carried out with finite precision is automatically bounded by the result of computing this propagation using IA. Since the publication of the first book on interval analysis (Moore, 1966), there has been a keen interest in applying interval-based verified-computing in numerous fields, ranging from computer-assisted
proofs in mathematical analysis (Lanford III, 1986) to practical engineering and industrial applications (Corliss, 1990). An introduction to interval analysis with a review of applications can be found in Kearfott (1996) and in Alefeld and Mayer (2000).

In the late 1990s a KF applied to dynamical interval systems, that is, systems whose parameters are described in terms of intervals, was proposed by Chen et al. (1997). Uncertainty in system modelling is often naturally described in this form. Physical parameters are usually not known precisely but specified with a certain tolerance, or have a varying nature: for example, as indicated earlier, the dynamics of marine surface vehicles depend, among other parameters, on their mass, which varies depending upon the number of passengers aboard. Even in the case of USVs, payload may vary depending on the mission as on-board equipment may be configurable, and fuel-driven USVs typically have a large fuel capacity to total platform weight ratio, making such variations significant. In the Springer state-space model (Equations 2 to 4), the coefficients of the matrices were obtained using SI techniques using a certain data set registered during a specific trial. During other trials, the values obtained may vary slightly due to reasons such as those previously outlined. The effect of incorrectly describing these values was illustrated in Figure 2. However, if a whole range of trials is performed, the varying results can be enclosed in intervals, resulting in an interval system model that contains each of the individual models obtained. In this way, all the possible dynamics are taken into account.

Chen et al. (1997) used the theory of IA to construct a KF for the interval system model. They obtained the IKF equations (Figure 3) using the same derivation as the regular KF. Suppose some elements of the matrices \( A, B, \) and \( C \) are uncertain within some definite bounds. The system can then be described by:

\[
x(k + 1) = A^I x(k) + B^I u(k) + \omega(k)
\]

\[
y(k) = C^I x(k) + \nu(k)
\]

where \( M^I = M \pm \Delta M = [M - |\Delta M|, M + |\Delta M|] \) for \( M \in \{A, B, C\} \), and \( \omega(k) \) and \( \nu(k) \) are white noise sequences with zero-mean Gaussian distributions with known covariances \( \text{cov}(\omega) = Q, \text{cov}(\nu) = R \), and \( E[\omega(l)\nu^T(k)]=0 \, \forall l,k, \, E[x(0)\omega^T(k)]=0, \, E[x(0)\nu^T(k)]=0 \forall k \).

The IKF algorithm is summarised by the equations shown in Figure 3 (Chen et al., 1997), which mimic those of the ordinary KF but are described in terms of intervals. Given an initial estimate \( x^I(0) \) and its uncertainty, characterized
by \( P(0) = \text{var}[\hat{x}(0)] \), together with the input to the system and the output measurement at each time-step, the resulting state estimate is an interval vector \( \hat{x}(k) \) at each time-step \( k \), providing an upper and lower boundary to the estimate, as illustrated in Figure 4.

Having been derived from the same principles, the IKF is statistically optimal in the same sense as the standard KF, and it maintains the same recursive formulation. However, the main advantage of the computed interval estimates, as opposed to point estimates, is that they guarantee to contain all the KF estimates of the individual models contained in the interval model, consequence of the inclusion property of IA by which if initial imprecise data is enclosed within rigorous bounds, then computation with these bounds will carry on yielding rigorous bounds of the actual solution range.

This can be important if one is to have confidence in the estimate. If the true system dynamics are known to be contained in the interval model, then the IKF provides a guaranteed enclosure of an optimal state estimate. While the precise value of this estimate will not be known, an interval may be acceptable for the required purpose, for example, if the object is to maintain a state variable between two limiting values, as long as the interval estimates remain within these limits no control action is required. Likewise, should the estimation boundaries permeate into an undesired operating region, this can be used to raise an alarm or trigger some other contingency mechanism.

5. IKF SIMULATION AND DISCUSSION. To illustrate the application of the IKF algorithm to the estimation of the yaw angle of the Springer USV, consider again the yaw dynamics modelled by Equations (5) to (7), but that the values of \( B \) are given with a tolerance of 25%. The smallest interval system that contains all the possible crisp models is described by:

\[
x(k + 1) = A\hat{x}(k) + B\hat{u}(k) + \omega(k)
\]

\[
y(k) = C\hat{x}(k) + \nu(k)
\]
with

\[
A^I = \begin{bmatrix} 1.002 & 0 \\ 0 & 0.9945 \end{bmatrix}, \quad B^I = \begin{bmatrix} [0.75 \times 6.354, & 1.25 \times 6.354] \\ [-1.25 \times 4.699, & 0.75 \times 4.699] \end{bmatrix} \times 10^{-6},
\]

\[
C^I = \frac{360}{2\pi} \begin{bmatrix} 34.13 & 15.11 \end{bmatrix},
\]

\[
\text{cov}(\omega) = \text{diag}(1, 1) \times 10^{-9}, \quad \sqrt{\text{var}(\nu)} = 2^\circ.
\]

Suppose also that the true dynamics are contained within the interval model, with the following \( B \) vector

\[
B_r = \begin{bmatrix} 1.25 \times 6.354 \\ -0.75 \times 4.699 \end{bmatrix} \times 10^{-6}
\]

Then, based on simulated values of \( \omega(k) \) and \( \nu(k) \) and the input values of \( n_d \) shown in Figure 5a, the application of the KF and IKF algorithms to the nominal model (Equations 5 to 7) and interval model (Equations 10 to 12), respectively, yield the estimates of the yaw angle shown in Figure 5b, in which the simulated true heading of the vessel is plotted as well.

It is clearly seen in Figure 5b how the nominal-system KF estimate lies within the IKF boundaries, and this holds for KF estimates obtained using any crisp model contained in the interval model. However, another phenomenon is evidenced: that of the rapid separation of the IKF boundaries. A consequence of the inclusion property of IA, which on the one hand is the \textit{raison d’être} of most interval analysis applications and on the other constitutes one of the major practical difficulties in implementing
IA-based algorithms, rendering results overly pessimistic and of little practical value. Hence, the consequently excessively conservative IKF boundaries may be attributed not to the theoretical framework of the algorithm laid out by Chen and co-researchers (1997) (1998), but with the way in which the IA calculations are implemented.

With regards to IA implementation, the IKF simulation was carried out with the aid of the open-source extension of MATLAB for IA, INTerval LABoratory (INTLAB), developed by Rump (1999). Numerous programming languages now contain libraries that extend the basic variable types to include intervals, and incorporate routines for carrying out IA operations (Kearfott, 1996). Albeit in themselves highly efficient, the sharpness of the results obtained in a computation involving IA is highly dependent on the particular numerical formulation adopted, whereby even with the aid of such software, a naïve use of IA may provide results that are of no practical use. In fact, the cardinal reason for this reliance of the resulting interval variable appears only once. Though this may not be possible for complicated expressions, one can in general minimise the occurrence of a single variable at a time in dynamical system simulation.

Overestimation caused by the dependency effect occurs because each occurrence of a variable in a mathematical expression is implicitly assumed to be independent; therefore, this effect is suppressed if the expression can be reformulated so that every interval variable appears only once. Though this may not be possible for complicated expressions, one can in general minimise the occurrence of a single variable at a time in the expression. Consider, for example, the computation of the first element of the a priori error covariance matrix \( P_1^{-1}(k) \) by the IKF algorithm for a second-order system:

\[
\begin{align*}
p_{11,k}^- &= (a_{11} p_{11,k-1}^+ + a_{12} p_{21,k-1}^+) a_{11} + (a_{11} p_{12,k-1}^+ + a_{12} p_{22,k-1}^+) a_{12} + q_{11} \\
&= (a_{11} p_{11,k-1}^+ + a_{12} p_{21,k-1}^+) a_{11} + (a_{11} p_{12,k-1}^+ + a_{12} p_{22,k-1}^+) a_{12} + q_{11}
\end{align*}
\]

where the shorthand notation \( p_k \equiv p(k) \) has been used for convenience. Then, noting that \( x^1(y^1+z^1) \subseteq x^1 y^1 + y^1 z^1 \), for \( x^1, y^1, z^1 \in \mathbb{R} \), the expression in (14) can be reformulated in the following ways, each one taking advantage of a different factorization and yielding a different interval enclosure for \( p_{11,k}^- \):

\[
\begin{align*}
f_1 = (a_{11}^+ p_{11,k-1}^+ + a_{12}^+ p_{21,k-1}^+) a_{11}^+ + (a_{11}^+ p_{12,k-1}^+ + a_{12}^+ p_{22,k-1}^+) a_{12}^+ + q_{11} \\
f_2 = (a_{11}^+ p_{11,k-1}^+ + a_{12}^+ p_{21,k-1}^+) a_{11}^+ + (a_{11}^+ p_{12,k-1}^+ + a_{12}^+ p_{22,k-1}^+) a_{12}^+ + q_{11} \\
f_3 = a_{11}^+ (a_{11}^+ p_{11,k-1}^+ + a_{12}^+ p_{21,k-1}^+) + (a_{12}^+ p_{22,k-1}^+) a_{12}^+ + q_{11} \\
f_4 = a_{11}^+ (a_{11}^+ p_{11,k-1}^+ + a_{12}^+ p_{21,k-1}^+) + a_{12}^+ (2a_{11}^+ p_{21,k-1}^+ + a_{12}^+ p_{22,k-1}^+) + q_{11}
\end{align*}
\]
the last three additionally making use of the symmetry of the error covariance matrix. It is not clear which one of these formulations, if any, yields the tightest enclosure, and even so, it would depend on the particular interval values taken on by each variable, which vary for each iteration. However, taking the intersection of all the intervals $f_i$ yields the narrowest enclosure of $\prod_{i=1}^{4} f_i \subseteq f_i \forall i$. Using this approach for every component of each of the IKF equations for the Springer interval model (Equations 10 to 12), the improved IKF boundaries obtained are shown in Figure 6b. Also shown are the boundaries obtained using naïve IA, the KF estimate based on the nominal model (Equations 5 to 7), the simulated real trajectory, and the average of the improved IKF boundaries.

It is clear again from Figure 6b that without any special treatment, the widths of the interval bounds grow very quickly; however, in comparison to the improved boundaries it is evidenced that they encompass not just the actual set of possible KF estimates, but a much larger set that dwarfs the former, rendering the bounds meaningless in practice. The simple treatment carried out here to reduce the dependency effect has shown a significant decrease in the overestimation of the interval widths. Still further reduction may be obtained by applying more sophisticated techniques. For example, Alefeld and Mayer (2000) show that a centred first-order Taylor representation of a function provides a quadratic order of approximation of the actual range, while Neumaier (2009) discusses exploiting monotonicity properties. Another strategy may consist of splitting the input intervals into smaller subintervals, obtaining an interval-enclosure of each one, and taking the union of these, since it has been shown to reduce the dependency effect (Daumas et al., 2009). Techniques have also been proposed to reduce the aforementioned wrapping effect of IA, such as Neumaier’s (1993) use of ellipsoids to compute affine transformations of vectors in
which both the vectors and the transformation matrix contain interval elements, or Kühn’s (1998) use of zonotopes, a kind of polytope, to approximate the system states.

Of particular importance is the computation of a good enclosure for the inverse interval matrix for the Kalman gain: since an interval matrix containing any singular point-valued matrix is not invertible, overestimation of intervals may result in the computed matrix being non-invertible, preventing further application of the IKF algorithm. Owing to this, several workaround strategies to bypass this difficulty have been proposed. For instance, Chen et al. (1997) proposed using a “suboptimal IKF” in which the inverse interval matrix is replaced by its worst-case inversion, which is an ordinary (non-interval) matrix. More recently, Xiong et al. (2012) have shown how to use the set inverter via interval analysis (SIVIA) method, an algorithm used to search the pre-image of a given set under a given function (Jaulin and Walter, 1993), to provide a tighter enclosure for the interval Kalman gain, while maintaining guaranteed bounds. In the wider field of interval analysis, efficient computational schemes for interval matrix inversion are being investigated, with work having recently been carried by Nirmala et al. (2011).

Recently, in order to address the conservatism inherent to IKF, Ahn et al. (2012) have derived a new filtering scheme from the IKF in which the interval uncertainties in the system model are incorporated into the covariances of the statistical model. Though this new filtering scheme is able to take into account interval model uncertainty, it however provides a point-valued estimate rather than the rigorous bounds of the IKF.

IA is also being applied to other robust filtering techniques. Kieffer et al. (1998) presented a state estimation technique in which system and measurement uncertainty are described as random sequences but bounded by intervals, rather than by the traditional stochastic modelling assumptions of KFs. This new technique, based on IA and set inversion, returns as estimate at each time-step a set guaranteed to enclose all values of the state that are consistent with the observations. Prediction and correction phases are alternated in a way reminiscent of the KF. Whereas Reece (2000) proposed a new filter, the Biscay distribution filter, which combines IA with statistical KF estimation methods, Ashokaraj et al. (2004) demonstrated the combined use of IA and the EKF for robot navigation, in which localisation estimates from the two methods were fused to create a more robust estimator.

Soon after Chen et al. (1997) published their IKF algorithm, an extended version for non-linear systems, the extended IKF (EIKF), was derived, and its usage was demonstrated in a missile-tracking simulation problem (Siouris et al., 1997). He and Vik (1999) extended the study of the EIKF by applying it to a simulation of an integrated GPS-INS system for aircraft navigation. Additionally, the simulation of a ship navigation system using the IKF to fuse GPS readings with dead reckoning calculated from compass and velocity measurements was reported by Tiano et al. (2005).

Though the IKF computes the bounds of the KF estimates of every point-valued system model contained in the interval model, in practice a unique value of the estimate is often needed as well. Chen et al. (1997) had suggested using a weighted average of the boundaries, or more simply, the average of the boundaries. The average of the IKF boundaries for the Springer yaw estimation is shown in Figure 6b, where it can be seen to roughly coincide with the KF estimate obtained using the nominal system model. In another study, a fuzzy logic inference scheme was applied to the IKF
(Chen et al. 1998) to deliver point-valued estimates. Later on, Weng et al. (2000) went on to describe the use of evolutionary programming techniques on an IKF to firstly, reduce the interval estimate at each iteration to an “optimal interval”, and secondly, find a nominal value within that interval which best represents the state estimate for practical purposes.

6. CONCLUDING REMARKS. While most USVs have typically been remotely operated, the demand for USVs with increasing levels of autonomy is on the increase. With the large number of emerging USV platforms and the wide use of the KF in navigation and for sensor fusion in general, research is currently being undertaken in developing robust navigation techniques for USVs, as changing maritime conditions, payload, etc, characterize these vessels. These translate into uncertainties in the dynamical model that can usually be easily bounded by intervals. The IKF is a natural extension of the KF for interval system models that provides rigorous bounds to the estimate in the face of such bounded model uncertainty.

In this paper, the IKF and KF algorithms are applied to the yaw angle estimation of the Springer USV, revealing the inclusive nature of the IKF estimate and its overly pessimistic yield when naïve IA is used. Undesirable interval widening is a common problem in all applications that require IA. Techniques are being developed to reduce this effect in the broader field of interval analysis to solve problems that require rigorous computing, where these may be adapted for use on the IKF algorithm. The simulation also shows how the IKF may substitute the regular KF algorithm as point estimates can be inferred from the boundaries if needed. The paper also registers the limited use the IKF has had since its inception, even though several improvements have been proposed and studied through computer simulations and its usage as a state estimator for systems with bounded model uncertainty being regarded favourably despite the difficulties of its implementation. It is intended that this paper will provide insight into the IKF’s purpose and applicability, as well as presenting the reader with a broad picture of the current state of the art regarding improvements to the IKF developed, with proposals for further investigation. Some of these are focused on improving boundary sharpness and others on inferring reliable point-valued estimates from the intervals. The inclusion of IA libraries in programming languages makes it possible to implement these strategies. The versatile payload of many USVs allows for full-scale computers to be used on board, as well as being able to host a wide variety of sensors, making USVs ideal test-beds on which to assess the viability of implementing IKF-based solutions in real systems, providing the incentive for its widespread use, while at the same time addressing the necessity of the USV market for increasingly robust navigation systems.

REFERENCES
APPENDIX. Consider the following linear stochastic-deterministic system

\[
x(k + 1) = A x(k) + B u(k) + \omega(k)
\]

\[
y(k) = C x(k) + v(k)
\]

where \(x(k), u(k), y(k)\) are vectors of the adequate dimensions that represent, respectively, the system state, the (deterministic) system input, and the measurement vector at time-step \(k\), and \(\omega(k)\) and \(v(k)\) are white noise sequences with zero-mean Gaussian distributions with known covariances \(\text{cov}(\omega) = Q, \text{cov}(v) = R\), and that \(E[\omega(l) v^T(k)] = 0 \forall l,k, E[x(0) \omega^T(k)] = 0, E[x(0) v^T(k)] = 0 \forall k\).

Then the KF equations provide a statistically optimal (unbiased and minimum error variance) estimate of the true state vector, and can be written as a set of recursive equations (Figure A1), from which the state estimate at each time-step is obtained from the previous estimate and the new observed measurement, assuming initial estimates for \(x(0)\) and \(P^I(0)\).

Figure A1. Kalman filter recursive formulation.

Application of artificial neural networks to weighted interval Kalman filtering


© SAGE Publications Ltd. Reproduced with permission.
Application of artificial neural networks to weighted interval Kalman filtering

Amit Motwani¹, Sanjay K Sharma¹, Robert Sutton¹ and Phil Culverhouse²

Abstract
The interval Kalman filter is a variant of the traditional Kalman filter for systems with bounded parametric uncertainty. For such systems, modelled in terms of intervals, the interval Kalman filter provides estimates of the system state also in the form of intervals, guaranteed to contain the Kalman filter estimates of all point-valued systems contained in the interval model. However, for practical purposes, a single, point-valued estimate of the system state is often required. This point value can be seen as a weighted average of the interval bounds provided by the interval Kalman filter. This article proposes a methodology based on the application of artificial neural networks by which an adequate weight can be computed at each time step, whereby the weighted average of the interval bounds approximates the optimal estimate or estimate which would be obtained using a Kalman filter if no parametric uncertainty was present in the system model, even when this is not the case. The practical applicability and robustness of the method are demonstrated through its application to the navigation of an uninhabited surface vehicle.

Keywords
Interval Kalman filter, artificial neural network, robust estimation

Date received: 21 August 2013; accepted: 29 November 2013

Introduction
In a recent study by Annamalai et al.,¹ a system for determining the heading of an uninhabited surface vehicle (USV) based on an interval Kalman filter (IKF) was explored. The IKF, designed upon imprecise knowledge of the vehicle’s yaw dynamics, provided upper and lower bounds to the statistically optimal estimate of its heading angle at each time instant. However, the guidance and control of the vehicle require a point-valued estimate of its heading. Since the optimal estimate must lie within the IKF interval estimate,² it must correspond to some weighted average of the interval boundaries. The problem remains on how to infer the appropriate weighting value and is the objective of this study. This article proposes a method that provides a good approximation to the optimal weight by using appropriately simulated data and artificial neural networks (ANNs).

Problem formulation
Consider the following stochastic–deterministic state-space model of a system’s dynamics

\[ x(k + 1) = A_m x(k) + B_m u(k) + \omega(k) \]  

\[ y(k) = C_m x(k) + \nu(k) \]  

with

\[ A_m = \begin{bmatrix} 1.002 & 0 \\ 0 & 0.9945 \end{bmatrix}, \quad B_m = \begin{bmatrix} 6.354 \\ -4.699 \end{bmatrix} \times 10^{-6} \]

\[ C_m = \begin{bmatrix} \frac{180}{\pi} & 34.13 & 15.11 \end{bmatrix}, \quad \omega(k) \sim N(0, Q_m), \quad \nu(k) \sim N(0, R_m) \]

\[ Q_m = \text{cov}(\omega) = \text{diag}(1, 1) \times 10^{-10}, \quad R_m = \text{var}(\nu) = 4, \quad T_s = 1 \]  

where \( u(k) \) is the known input to the system, \( \omega(k) \) represents a random input disturbance, \( y(k) \) is the measured output and \( \nu(k) \) represents a random measurement noise. Assuming that the random processes follow

¹Marine and Industrial Dynamic Analysis Group, School of Marine Science and Engineering, Plymouth University, Plymouth, UK
²School of Computing and Mathematics, Plymouth University, Plymouth, UK

Corresponding author:
Amit Motwani, Marine and Industrial Dynamic Analysis Group, School of Marine Science and Engineering, Plymouth University, Drake Circus, Plymouth, Devon PL4 8AA, UK.
Email: amit.motwani@plymouth.ac.uk
zero-mean Gaussian distributions, then the classical Kalman filter (KF) approach based on combining model predictions with actual measurements may be used to obtain statistically optimal estimates of the system output. Let such a filter, based upon the above model, be denoted as KF-1 (the KF equations are detailed in Appendix 1). 

Now, let it be supposed that the modeller is uncertain of the precise values of the vector $B_m$ and in fact declares that

the values of $B$ could not be ascertained with complete exactitude, however, it is possible to warrant that they are not further departed from these than by an amount equating to twenty five per cent of the same.

Upon this revelation, it is apparent that the filter KF-1 will no longer supply an optimal estimate if the actual values of the state equation’s input vector, $B$, depart from those of $B_m$, and in particular, the difference between the measurements $\gamma(k)$ and the model’s predicted output $Cx(k)$ will no longer have a zero-mean value. In order to try and account for imprecisely modelled values while maintaining a KF-like structure, Chen et al. proposed describing the system dynamics using intervals. Consider the following interval model

\[
\begin{align*}
x(k + 1) &= A^I x(k) + B^I u(k) + \omega(k) \\
y(k) &= C^I x(k) + \nu(k)
\end{align*}
\]

with

\[
A^I = \begin{bmatrix} [1.002, 1.002] & [0, 0] \\ [0, 0] & [0.9945, 0.9945] \end{bmatrix}, \\
B^I = B_m - 0.25 \times \text{abs}(B_m), B_m + 0.25 \times \text{abs}(B_m) = \begin{bmatrix} [0.75 \times 6.354, 1.25 \times 6.354] \\ [1.25 \times (-4.699), 0.75 \times (-4.699)] \end{bmatrix} \times 10^{-6}, \\
C^I = \frac{1}{\pi} \begin{bmatrix} [34.13, 34.13] \\ [15.11, 15.11] \end{bmatrix}, \\
\omega(k) \sim N(0, Q), \nu(k) \sim N(0, R), \\
Q = \text{cov} (\omega) = \text{diag} \{1, 1\} \times 10^{-10}, R = \text{var}(\nu) = 4, T_s = 1
\]

in which the components of $A^I$, $B^I$ and $C^I$, as well as those of the system state vector $x^I$ and output $y$, are now given by interval values rather than ordinary point values. For simplicity but without loss of generality, the example illustrated here only contains interval model coefficients with non-zero widths in the vector $B^I$, as it is assumed that all other coefficients are modelled precisely. Based on this interval model, an IKF, which provides interval-valued estimates, can be designed.

Let it also be supposed that the true dynamics of the system, while not corresponding exactly to the values given in equation (3), are contained within the interval model (equations (4)–(6)) and are given by

\[
\begin{align*}
\dot{x}(k + 1) &= Ax(k) + Bu(k) + \omega(k) \\
\dot{y}(k) &= Cx(k) + \nu(k)
\end{align*}
\]

where the notation $\dot{u}(k)$ implies $u(kT_s)$ and is used for convenience. Then, based on simulated values of $\omega(k)$ and $\nu(k)$ and the respective filter models, the estimates of the output by the three filters can be calculated and are shown in Figure 1(b) (the KF and IKF equations are detailed in Appendix 1). It is to be noted that in all cases, the measurements are simulated using the true system’s dynamics (equations (7)–(9)) and not the

Figure 1. KF-ideal, IKF and KF-1 estimates of the output of the system to the sinusoidal input $u(k) = 15 \sin(0.01k)$: (a) $u(k) = 15 \sin(0.01k)$ and (b) system output estimates.
Figure 2. (a) Sequence of weights calculated so that the weighted average of the IKF boundaries coincides with the KF-ideal estimate and (b) innovation sequence of KF-ideal and KF-1.

**IKF: Interval Kalman filter.**

respective models since they represent the actual measurements.

Several observations ensue. First, it can be seen that the KF-1 estimate deviates from the KF-ideal estimate due to the incorrect model assumed by the former. However, both of these lie within the bounds of the IKF interval estimate, as the latter must in principle contain every single KF estimate arising from a model contained within the interval model. Finally, it can also be verified that the arithmetic average of the IKF bounds approximately coincides with the KF-1 estimate.

Let \( y^{IKF}(k) \) be the IKF estimate of the system output. If a weight \( w \in [0, 1] \) is chosen at each time step, then the weighted average of its bounds, henceforth the weighted interval Kalman filter (wIKF) estimate, is given by

\[
y^{wIKF}(k) = y^{IKF}(k) + w(k)[y^{IKF+}(k) - y^{IKF-}(k)]
\]

with

\[
y^{IKF+}(k) = \max\{y^{IKF}(k)\} \quad \text{and} \quad y^{IKF-}(k) = \min\{y^{IKF}(k)\}
\]

and lies within the boundaries of the IKF interval estimate. In addition, based upon the previous observations, there exists a particular value of \( w \) for which the wIKF estimate matches the KF-ideal estimate. (Note also that the KF-1 estimate can be computed from equation (11) with \( w(k) = 0.5 \).

Figure 2(a) depicts, at each time step, these desired weights that produce a wIKF estimate coincident with that of KF-ideal, easily calculated from equation (11) when the KF-ideal estimate is known. The key question is can these weights be calculated in practice without the knowledge of the true system dynamics, and hence, without the availability of the KF-ideal estimate? The answer, fortunately, is yes, as is explained in what follows.

It is well established that under optimal conditions, the innovations of the KF, or difference between a priori prediction and measured output, should comprise a white noise sequence. However, under erroneous modelling assumptions, the optimality of the KF estimate is lost, resulting in an innovation sequence that ceases to correspond to white noise. The innovation sequences of both the KF-ideal and KF-1 estimates of Figure 1(b) are shown in Figure 2(b).

It seems likely that there should exist a deterministic relationship between the innovation sequence and the desired weighting sequence, and as such, it should be possible to model such a relationship. It is also well established that ANNs are capable of replicating complex cause-effect relationships, enabling one to predict the output of such processes for new inputs.

### An ANN as the missing link

The recurrent multi-layer perceptron (RMLP)-type ANN shown in Figure 3 was trained using as input the innovation sequence of KF-1 and as target the desired weights (Figure 2). Due to the fact that the relationship between innovations and desired weights in most likelihood depends not just on the instantaneous values but on the trends of the innovations as well, these trends were incorporated into the ANN model by considering six consecutive values of the innovations for each desired output, consisting of the present value as well as the previous five values: \( \text{inn}(k), \text{inn}(k-1), \ldots, \text{inn}(k-5) \). Although not apparently necessary, another feature was added to the input of the network: the width of the IKF interval, \( \Delta \text{ikf}(k) = y^{IKF+}(k) - y^{IKF-}(k) \). The addition of this extra input was seen to enhance the performance of the network, the reason for which will be discussed in a later section.

It was also observed that the use of feedback from the output also helped increase the network’s accuracy, and so, five time-delayed values from the output were fed back as inputs to the network. Thus, the combined input to the network at time step \( k \) (not counting the bias unit) can be described as

\[
x(k) = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_6 \\ x_7 \\ x_8 \\ \vdots \\ x_{12} \end{bmatrix} = \begin{bmatrix} \text{inn}(k) \\ \text{inn}(k-1) \\ \vdots \\ \text{inn}(k-5) \\ \Delta \text{ikf}(k) \\ \hat{\omega}(k-1) \\ \vdots \\ \hat{\omega}(k-5) \end{bmatrix}
\]

where \( \hat{\omega} \) is the output of the network.

Details of the training process used are given in Appendix 2. The training results are shown in Figure 4(b). The virtue of the fit is evaluated by calculating the mean square error (MSE) between the predicted output \( \hat{\omega} \) and the desired one \( \omega \), and comparing it to the MSE.
between a constant weighting sequence of 0.5 (the default weighting that would be used to select a nominal value from the IKF estimate in the absence of any specific criterion) and \( w_t \). In this case, the MSE decreases from 0.061205 for the latter to 0.001762 for the ANN prediction, a decrease of over 1.5 orders of magnitude.

Figure 4 clearly shows that the trained ANN establishes a mapping between the inputs (innovation sequence of KF-1 and IKF interval width) and the desired weighting. However, it is crucial to investigate whether this model generalises well to new data.

In order to test the trained ANN on new data, two new data sets were generated from new input signals applied to the dynamic system, from which the KF and IKF estimates, desired IKF weighting and KF innovation sequences were generated. These are summarised in Figures 5 and 6. Figure 5(a) depicts a superposition of sinusoidal waveforms of various frequencies and amplitudes used as input to the dynamic system, while Figure 5(b) and (c) show the data set generated from it. Also in Figure 5(c) is the predicted output of the previously trained ANN to the signals shown in Figure 5(b). Similar graphs are shown in Figure 6 for the case in which an input consisting of a square waveform was applied to the dynamic system (equations (7)–(9)). Table 1 summarises the test performances of the trained ANN on these new test sets.
Test case 1.

As can be observed, in both test cases, the MSE of the trained ANN output is considerably lower than the mean square difference between the target weighting and the constant weight that represents the arithmetic mean of the IKF boundaries.

Thus far so good; however, the attentive reader may pose the following conundrum: in practice, the real system dynamics may differ from that which was used to generate the data on which the ANN was trained. In fact, this is most likely to be the case, and one would not know the precise values representative of the real system dynamics, for if that were so, then there would be no need for using an IKF in the first place.

Hence, let it now be supposed that the true dynamics of the system are given by

\[ x(k+1) = Ax(k) + Bu(k) + \omega(k) \]  
\[ y(k) = Cx(k) + \nu(k) \]

rather than the values given in equation (9). One should wonder whether the ANN trained under the previous (initial) assumptions would still be able to correlate innovations with desired weightings in this new context. Let an input sequence given by equation (10) (Data 1) now be applied to the system described by equations (13)–(15) and the corresponding KF and IKF estimates be calculated. Figures 7(a)–(c) show the input, the KF-1 innovation sequence together with the IKF interval widths, and a comparison of the desired weighting sequence with the output of the trained ANN, respectively. The MSE of the ANN prediction with respect to the desired weighting is 0.007434, a 72.17% reduction compared to the value of 0.026711 that results if a constant sequence of 0.5 is used.

This ascertains that the ANN trained from data generated through simulation by using some assumed system dynamics can still be applied successfully to the prediction of the desired IKF weighting sequence even when the true system dynamics differ from those assumed for training, as long as both lie within the intervals constitutive of the interval model.

A case study presenting how these concepts may be used in practice is detailed in the following section.

An ANN-guided wIKF for the navigation of a USV

In this section, the ideas described previously are applied to the problem of estimating the heading angle...
of an Uninhabited Surface Vehicle (USV). The vehicle in question consists of a twin-hull catamaran driven by two propellers, the difference in speed of which enables the vehicle to steer, and is controlled purposefully to this end. Let $n_d$ represent the difference in revolutions per minute (rpm) of the two propellers, such that the vehicle steers to the left when $n_d$ is positive and to the right when it is negative. In effect, from a control point of view, the vehicle’s yaw dynamics has been modelled using system identification (SI) techniques.\(^7\) The model obtained is precisely that described by equations (1)–(3), where $u$ is the aforementioned differential speed of the propellers in rpm, $n_d$, and $v$ models random input disturbances to the system to take into account randomly varying surface effects. Additionally, the vehicle is equipped with a magnetic compass unit that provides the instantaneous heading with a random unbiased root mean square (RMS) error of $2^\circ$ and is given by $y$ in equation (2), where $v$ represents the compass error.

In a simulation study, the vehicle is given the task of completing a way-point following mission as shown in Figure 8. The trajectory depicted is followed by the vehicle operating under an autonomous guidance and autopilot system based on line of sight (LOS) and proportional–integral–derivative (PID) control, respectively, under the constraint of maintaining a constant forward speed of 3 knot or 1.54 m s\(^{-1}\). Additionally, the value of $n_d$ though calculated by the autopilot is subsequently hard limited to within $\pm 300$ rpm, with a maximum permitted variation from one time step to the next of $\pm 20$ rpm, reflecting the physical limitations of the hardware. In this simulation, the vehicle’s actual heading was assumed available to feed back to the guidance and autopilot systems. Further details of this setup can be found in Annamalai et al.;\(^1\) as for the study undertaken herein, only aspects relative to estimation of the heading of the vehicle given a model of the same and a specified control input are needed.

In order to estimate the heading of the vehicle from the noisy measurements of the compass, a KF may be used. However, as discussed previously, under incorrect modelling assumptions, the KF estimate may become biased. Let it be assumed that although the exact model of the vehicle’s dynamics is not known, it is certain to be contained within the interval model described by equations (4)–(6). In this scenario, the technique

<table>
<thead>
<tr>
<th>Test case 1</th>
<th>Test case 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE_0.5</td>
<td>MSE_ANN</td>
</tr>
<tr>
<td>0.044161</td>
<td>0.022765</td>
</tr>
</tbody>
</table>

MSE: mean square error; ANN: artificial neural network.

![Figure 7. Performance of trained ANN on test set generated from modified system (equations (13)–(15)).](image)

(a) System input: $u(k) = 15 \sin(0.01k)$. (b) KF-1 innovation sequence and $\Delta k_f$ and (c) prediction performance: comparison of desired weighting sequence and ANN output.

IKF: interval Kalman filter; ANN: artificial neural network; MSE: mean square error.

![Figure 8. Way-point following mission showing the trajectory followed by the USV.](image)

USV: uninhabited surface vehicle.
described previously may be put into practice: a set of dynamics contained within the interval model may be assumed as the ‘true’ vehicle dynamics, and the estimates of both a KF based on it and a KF based on the nominal model (equations (1)–(3)) can be simulated, together with the interval estimates from an IKF founded on the interval model. The desired weighting sequence can then be obtained as that which is necessary for the $w_{\text{IKF}}$ values to match those of the KF estimates that were obtained using the assumed ‘true’ dynamics. Finally, an ANN can be trained to obtain these desired weights from the innovation sequence of biased KF estimates (those obtained from the KF that uses the nominal system model).

In order to train the ANN, rather than use the waypoint mission described earlier to generate the required input and target data, a different mission was used. This allows the mission described previously to be used to test the method in order to evaluate its performance. The training mission chosen consists of 14 way-points which are shown in Figure 9(a). These way-points were chosen to provide a variety of turning angle requirements assuming that the vehicle reaches the way-point along the ideal trajectory. For instance, if the vehicle reaches way-point 1 facing east, then it is initially required to turn 20° towards the left to head towards the following way-point. When way-point 2 is reached, it is required to turn left an additional 40°, and so on until 100° from way-point 5 to way-point 6, before heading back towards the initial coordinates. It then has to repeat the trajectory but this time in a clockwise direction. This forces a large range of the vehicle’s dynamics to be used for generating training data for the ANN and thus should favour its capability of accurate prediction for any standard trajectory.

Another point for consideration is what model to choose from the interval model to represent the ‘true’ dynamics of the vehicle for simulation. Instead of choosing a single model for the whole mission, different sets of models were chosen during different time intervals. The values of $B$ for simulating the vehicle’s dynamics during the course of the training mission

![Figure 9](image-url)

Figure 9. Way-point following mission for training the ANN. (a) Way-point specification and trajectory followed by the USV for the training mission, (b) controller input: differential thrust calculated by the autopilot, (c) heading estimates, (d) KF-1 innovation sequence and $\Delta k_f$ and (e) comparison of desired weighting sequence and trained ANN output.

USV: uninhabited surface vehicle; ANN: artificial neural network; KF: Kalman filter; IKF: interval Kalman filter; MSE: mean square error.
Table 2. Models for generating test data.

<table>
<thead>
<tr>
<th>Test case 1</th>
<th>Test case 2</th>
<th>Test case 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>(B(1) = B_m(1) + 0.25B_m(1))</td>
<td>(B(1) = B_m(1) - 0.25B_m(1))</td>
<td>(B(1) = B_m(1) + 0.25B_m(1))</td>
</tr>
<tr>
<td>(B(2) = B_m(2) - 0.25B_m(2))</td>
<td>(B(2) = B_m(2) + 0.25B_m(2))</td>
<td>(B(2) = B_m(2) + 0.25B_m(2))</td>
</tr>
</tbody>
</table>

were set to initially coincide with those of \(B_m\) up until reaching the first way-point and subsequently to vary as follows

\[
B(1) = B_m(1) + 0.25B_m(1)\text{sign}\left[\sin\left(\frac{2\pi}{T_1} k\right)\right], \quad T_1 = 400 \text{ s}
\]

(16)

\[
B(2) = B_m(2) + 0.25B_m(2)\text{sign}\left[\sin\left(\frac{2\pi}{T_2} k\right)\right], \quad T_2 = 450 \text{ s}
\]

(17)

The training data set will thus be generated from multiple dynamic systems covering several combinations of values selected from the extremes of the interval model and provides a richer training set than would be generated using a constant set of dynamics for simulating the vehicle for the entire duration of the mission.

Based on these varying dynamics, Figure 9(a) shows the actual path taken by the vehicle using LOS guidance and a PID autopilot, while the PID-generated differential propeller speed is shown in Figure 9(b). For the trajectory followed, both guidance and control systems were given the actual heading of the vehicle. Additionally, a KF was used to estimate the heading based on simulated noisy compass measurements, using the actual dynamics of the vehicle for prediction. The estimates of this KF are labelled as KF-ideal in Figure 9(c). A second KF, one that used the nominal system model instead, was used to generate the estimates labelled as KF-1 in Figure 9(c). Finally, an IKF was also implemented and the interval estimates therefrom are plotted on the same figure.

Figure 9(d) shows the innovation sequence of the biased KF along with the IKF interval widths, both of which are used as inputs to the ANN. Finally, Figure 9(e) shows the desired (target) weighting sequence (calculated so that the wIKF estimate matches the unbiased KF estimate (KF-ideal)). This input and target data set were used to train the ANN of Figure 3. The trained network’s output is plotted alongside the target data and is generally the case due to the nature of IKF interval computations where the initially narrow intervals tend to grow. When the IKF widths are small, errors in the weights become less significant, as both IKF bounds are themselves already close to the optimal estimate, and hence, so is any weighted average of these (with weight between 0 and 1). Therefore, a performance measure that takes this into account would flaunt even better numbers than those presented in the table.

It should also be said that the ANN architecture presented here (Figure 3), and its particular characteristics (layer sizes, etc.), was found to provide a good balance between prediction accuracy and number of neurons employed, but is by no means the only valid architecture that can be used, and furthermore, for each system, the most suited type and size of ANN should be explored.

Discussion and conclusion

Although the KF has been used extensively and successfully in numerous applications, when the model used by the KF is deficient, then as demonstrated in the preceding sections, the estimates tend to become biased. For applications that increasingly require robust state variable estimation at reduced costs, carrying out precise modelling of the system becomes prohibitive. Such
is the case, for example, with low-cost USV systems which have seen a recent surge in the number of applications, and for which ever-increasing levels of autonomy are desired at reduced costs.8 SI is a popular modelling strategy to this end, but because it is based on empirical data, the model values obtained must be understood to be precise to within a certain tolerance. Even if accurate models could be developed, tolerance to changes in system dynamics is also a quality that is increasingly required; for example, in the case of USVs, varying mission objectives could mean being able to operate under varying payload. For this reason, robust navigation systems that can handle imprecisely modelled or varying dynamics are needed.

The IKF was developed to extend the KF to systems described in terms of intervals rather than precise point-valued quantities. However, the main problem with interval filtering is that due to the conservative nature of interval computation, the estimates tend to be over-conservative, limiting their practical usage.2 In practice, a single estimate is often required, and several studies have been aimed at inferring point-valued estimates from the interval estimates of the IKF. Chui and Chen9 suggested using a weighted average of the IKF boundaries, and, in the absence of any weighting criteria, to take the arithmetic average of the boundaries. As demonstrated in this article, the wIKF methodology developed provides estimates that are much improved over taking the simple arithmetic average of the interval bounds. Another method was proposed by Weng et al.10 in which evolutionary programming is used as a global search method to find the point estimate that minimises the maximum estimation error covariance. However, on the one hand, this method requires running an iterative search algorithm at each time step, and on the other hand, it does not use actual measurement data to infer the desired point-valued estimate, being based on statistical principles alone. In the approach used here, the training of the network is done offline, so that it is only used for prediction during an actual mission. This only requires forward propagation of information through the network, which can be computed efficiently using a vectorised implementation.

Other robust filtering approaches have been proposed which alter the basic hypotheses or structure of the IKF algorithm,11,12 resulting in either loss of the optimality quality of the IKF’s interval estimate or rigour in the sense of guaranteeing to contain any optimal estimate of a particular realisation of the interval system. In the method presented here, the IKF original estimate is maintained, while its boundaries are simply used to infer a point-valued estimate for practical purposes. Thus, while this latter estimate can be used, referring for example to the case study presented here, as input to a guidance and control system, the IKF intervals themselves can be used to compute guaranteed bounds to the trajectory followed by the vehicle and to trigger an alarm should these bounds permeate into an undesired region.

A pending comment with regard to the particular example used in this article to demonstrate the technique developed is the use of Δk/σ(k) as additional input to the ANN. Its use was seen to increase the predictive accuracy of the network, especially to correct the scaling of the prediction. A heuristic explanation for this phenomenon is the following. The IKF intervals themselves inevitably tend to widen over-conservatively as mentioned earlier, and in fact, depending on the sharpness of the interval computation, the width may vary significantly. However, they all represent the same ‘optimal interval’ that would be obtained if interval computation could be carried out with infinite sharpness. In other words, the ANN developed here should be immune to the exact width of the IKF interval and sensitive only to the innovations of the biased KF.
estimate. It thus requires information of the former, which should somehow be incorporated into the ANN prediction process since its target output, the optimal wIKF estimate, is computed from the IKF bounds themselves.

To conclude, it has been demonstrated how an ANN can be trained successfully to use residual KF data (the innovation sequence) to infer advantageous weightings for obtaining point-valued estimates from IKF boundaries, as compared to simply using, for example, the arithmetic mean of the boundaries (which provides similar estimates to that of the KF that uses the incorrect nominal model). The test results for the case study presented here (Table 3) show that the trained ANN is capable of generalising well to new situations. Depending on the application, it is always possible to develop an adequate training set that will enable effective prediction for new missions within the scope of the application. This required analysis, the training process and evaluation of the trained network on a set of new missions can all be done beforehand and via simulation alone, rendering this method cost-effective, reliable and practically realisable.

### Declaration of conflicting interests

The authors declare that there is no conflict of interest.

### Funding

This work was supported by the Engineering and Physical Sciences Research Council (grant no. EP/I012923/1).

### References


### Appendix 1

Consider the following linear interval stochastic–deterministic system

\[
x^I(k+1) = A^I x(k) + B^I u(k) + \omega(k)
\]

\[
y^I(k) = C^I x(k) + \nu(k)
\]

where the components of \(A^I, B^I, C^I\), the state vector \(x^I(k)\) and the output vector \(y^I(k)\) are interval values; \(u(k)\) is the (deterministic) system input and \(\omega(k)\) and \(\nu(k)\) are white noise sequences with zero-mean Gaussian distributions with known covariances \(\text{cov}(\omega) = Q, \text{cov}(\nu) = R\), and \(E[\omega(l)\omega^T(l)] = 0\forall l, k, E[y^I(0)\omega^T(k)] = 0, E[x^I(0)\nu^T(k)] = 0\forall k\).

Then, given successive measurements of the output, the IKF equations (equations (20)–(24)) provide an interval enclosure of the statistically optimal (unbiased and minimum error variance) estimates of the system state vector, for every point-valued system contained in the interval model. The state estimate at each time step is obtained from the previous estimate and the new observed measurement, \(y(k)\), assuming initial estimates for \(x^I(0)\) and the error covariance matrix \(P^{I+}(0)\). Note that the measurement vector is a realisation of the uncertain interval vector \(y^I(k)\) and is an ordinary vector (with point-valued elements).
Prediction
\[ \hat{x}^t(k + 1) = A^1 \hat{x}^t(k) + B^1 u(k) \]  
\[ p^t(k + 1) = A^1 p^t(k) A^T + Q \]  
Kalman gain
\[ K^t(k) = p^t(k) C^T \left( C^T p^t(k) C^T + R \right)^{-1} \]
Correction
\[ \hat{x}^t(k) = \hat{x}^t(k) + K^t(k) [y(k) - C^t \hat{x}^t(k)] \]  
\[ p^t(k) = [I - K^t(k) C^T] p^t(k) \]
and the output estimate at time \( k \) is simply
\[ y^{\text{ext}}(k) = C^t \hat{x}^t(k) \]
When the elements of equations (18) and (19) are all point-valued so that \( A^t, B^t, C^t, \hat{x}^t, \) and \( y^t \) can be replaced by \( A, B, C, \hat{x} \) and \( y \), respectively, then the recursive equations (20)–(25) describe the ordinary KF algorithm by replacing \( A^1, B^1, C^1, \hat{x}^t, \hat{x}^t, p^t, \) and \( y^t \) with \( A, B, C, \hat{x}, \hat{x}, P, P \) and \( y^{\text{ext}} \), respectively.

Appendix 2
The RMLP of Figure 3 has a hidden layer with five units, all of which incorporate hyperbolic tangent activation functions. Information is propagated forward through the network at each time step according to equation (26).

Forward propagation
\[ a^{(1)} = \begin{bmatrix} 1 \\ x \end{bmatrix}; \quad a^{(2)} = \begin{bmatrix} \tanh((\Theta^{(1)} a^{(1)})^i) \end{bmatrix}; \quad \hat{w} = a^{(3)} = \Theta^{(2)} a^{(2)} \]
where \( a^{(d)} \) are the outputs of the nodes of layer \( (d) \), and \( \Theta^{(1)} \in \mathbb{R}^{2 \times 15} \) and \( \Theta^{(2)} \in \mathbb{R}^{1 \times 6} \) are the matrices of parameters of the network such that \( \Theta^{(2)}_{ij} \) represents the strength of the connection between node \( a^{(2)}_i \) and \( a^{(1)}_j \) (Figure 3).

Training the network consists of finding the parameters \( \Theta^{(2)}_{ij} \) that minimise the cost function
\[ J = \frac{1}{m} \sum_{k=0}^{m-1} (w_t(k) - \hat{x}(k))^2 \]  
where \( m \) being the number of training samples. This process was carried out recursively via the gradient descent (GD) algorithm: after assigning random initial values to \( \Theta^{(2)}_{ij} \), the parameters are updated as
\[ \Theta^{(2)}_{ij} = \Theta^{(2)}_{ij} - \alpha \frac{\partial J}{\partial \Theta^{(2)}_{ij}} \]  
until convergence is reached, where \( \alpha \) is the learning rate, chosen adequately based on trial and error. The gradient of the cost function with respect to the network’s parameters was computed using the back-propagation (BP) method.6

BP. For each training pattern \( x(k) \) (equation (12)) and target \( w_t(k) \)
1. Compute \( \hat{w} \) (equation (26));
2. \( \delta^{(2)} = \left[ 1 - \tanh^2 \left( \sum_j (\Theta^{(1)} a^{(1)})^j \right) \right] \Theta^{(2)} \delta^{(3)} \); \( i = 1, \ldots, 5 \)
3. \( \delta^{(3)} = \delta^{(2)} - \delta^{(1)} a^{(2)} \); \( i = 0, \ldots, 5 \)
4. \( \delta^{(1)} = \delta^{(1)} - \delta^{(2)} a^{(1)} \); \( i = 1, \ldots, 5; j = 0, \ldots, 12 \)
end
\[ \frac{\partial J}{\partial \Theta^{(2)}_{ij}} = \frac{1}{m} \delta^{(2)} \]

The GD process was applied in two stages, depending on how the gradient was calculated. During the first set of iterations, the (delayed) target values \( w_t \) were used to construct \( x(k) \) for computation of \( \hat{w} \) in the first step of the BP process, effectively training a network without feedback. During a second stage, (past) predictions of the network \( \hat{w} \) were used to construct \( x(k) \) in accordance with the true feedback architecture of the network. Although the gradients computed with the BP algorithm in this case are approximations to the true gradients, the errors are small as after the first set of iterations, the network is sufficiently trained to output predictions close to the target values.

Computation of stable interval Kalman filter bounds for their use in robust state estimation for an uninhabited surface vehicle with bounded indeterminate system dynamics

Proceedings of the IEEE Intelligent Vehicles Symposium, 8-11 June, Dearborn, MI, USA, pp 356 -361

© 2014 IEEE. Reprinted with permission.
Computation of stable interval Kalman filter bounds for their use in robust state estimation for an uninhabited surface vehicle with bounded indeterminate system dynamics*

Amit Motwani, Sanjay Sharma, Robert Sutton and Phil Culverhouse

Abstract—This paper implements an interval Kalman filter (IKF) for the navigation of the Springer uninhabited surface vehicle. Interval filters become necessary when the system dynamics are not known precisely or vary unpredictably, but can nevertheless be described in terms of bounded intervals. Such filters based on interval systems require the use of interval arithmetic (IA) for their operation. One of the main limitations to such techniques is that the interval bounds of the computed filter estimates often diverge due to the overly conservative nature of IA. In this paper, ellipsoidal rather than direct IA is used to operate the IKF and obtain bounds of the interval estimates that do not diverge due to the so-called wrapping effect. From these bounds, a weighted average is computed at each time-step that is close to the true system state. To obtain this weighting, an artificial neural network (ANN) is previously trained to map residuals of an ordinary Kalman filter to the optimal weights, and this trained network is then used online in new tracking missions.

I. INTRODUCTION

There is a growing need for low-cost autonomous navigation systems for uninhabited surface vehicles (USVs), and to be truly useful, such systems need to be robust in terms of being able to function under varying operating conditions. The Springer USV was constructed as a test-bed on which to develop such systems. It is a medium waterplane twin hull vessel measuring 4.2m long and 2.3m wide, designed to undertake autonomous environmental surveillance in shallow waters. The vehicle itself is equipped with a range of electronic navigational sensors and on-board PCs housed within watertight Pelican cases situated on each hull. Full details regarding the vehicle’s hardware can be found in [1], and only the propulsion system will be further detailed in the next section.

The present study proposes a navigation system for the Springer that does not require precise knowledge of the vehicle’s dynamics and can maintain a good operational capability through changes in these dynamics.

II. WAYPOINT TRACKING MISSION

A. The Springer USV yaw dynamics

Springer is propelled by two electric trolling motors. Steering is based on differential propeller revolution rates, whereas if the average of these is maintained constant, then the forward speed of the vehicle remains largely unchanged. Hence, for a constant forward speed (constant average rpm of the motors), the steering can be considered as a single-input, single-output system, with the differential revolution rate being the input and the rate of change of the vehicle’s heading the output. The governing dynamics of said system may be approximately described in terms of a linear model. Such a model can easily be obtained using system identification (SI) techniques [2].

For this study, consider the steering dynamics of Springer given by the following discrete state-space model for a constant forward speed of 1.5 ms⁻¹,

\[
x(k + 1) = Ax(k) + Bu(k) + \omega(k)
\]

\[
y(k) = Cx(k) + Du(k)
\]

\[
A = \begin{bmatrix} 0.8 & -0.2225 \\ 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix},
\]

\[
C = \begin{bmatrix} 0 & 0.004225 \end{bmatrix}, \quad D = 0; \quad T_s = 1 \text{s},
\]

\[
\omega \sim N(0,Q); \quad Q = \text{cov}(\omega) = 0.01 \times \text{diag}[1,1]
\]

where \(u(k)\) is the differential motor speed input and \(y(k)\) the rate of change of heading of the vehicle, \(\omega\) represents a random input disturbance, and \(T_s\) is the sampling period. It will also be considered that \(|u(k)| \leq 300\) rpm and \(|\Delta u(k)| = |u(k)-u(k-1)| \leq 40\text{rpm}\) due to actuator limitations.

B. Way-point tracking mission

Consider the autonomous tracking mission shown in Fig.1, consisting of a series of way-points with known coordinates through which the vehicle must traverse. In order to achieve this, at each sampling instant a reference heading angle is determined according to line-of-sight:

\[
r(k) = \arctan \left( \frac{y_d(k) - y_{USV}(k)}{x_d(k) - x_{USV}(k)} \right)
\]

where \((x_{USV}, y_{USV})\) represents the current location of the vehicle, assumed to be known at each sampling time (e.g. via

* Research supported by EPSRC, UK.

The authors are with the Marine and Industrial Dynamic Analysis Group (MIDAS), School of Marine Science and Engineering, Plymouth University, Plymouth, UK.

Corresponding author: A. Motwani; phone: +44(0)1752-586147; fax: +44(0)1752-586147; (e-mail: amit.motwani@plymouth.ac.uk).
a GPS receiver), and \((x_r, y_r)\) the target coordinates, or coordinates of the next way point. The autopilot in turn determines the control action, i.e. the differential motor speed that must be applied.

Feedback controllers generally compute a control action based on comparison of the output of the system with the reference or desired output. In order to obtain the heading angle of the vehicle to compare with the reference angle \(r(k)\), the output of (2) can be integrated according to (5). Then, with the heading as the output, the system can be described by the augmented state-space equations (6) and (7).

\[
\theta(k+1) = \theta(k) + y(k) = \theta(k) + C \cdot x(k) + Du(k)
\]  
\[
\begin{bmatrix}
  x_1(k+1) \\
  x_2(k+1) \\
  \theta(k+1)
\end{bmatrix} =
\begin{bmatrix}
  A & 0 & \theta(k) \\
  0 & B & \omega_1(k) \\
  -C & D & \omega_2(k)
\end{bmatrix}
\begin{bmatrix}
  x_1(k) \\
  x_2(k) \\
  \theta(k)
\end{bmatrix} +
\begin{bmatrix}
  u(k) \\
  \omega_1(k) \\
  0
\end{bmatrix}
\]  
\[
\begin{array}{c}
  x_1(k) \\
  x_2(k) \\
  \theta(k)
\end{array} =
\begin{bmatrix}
  0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  x_1(k) \\
  x_2(k) \\
  \theta(k)
\end{bmatrix}  
\]  

or with the appropriate definitions, expressed compactly as

\[
\ddot{x}(k+1) = \ddot{\dot{x}}(k) + \ddot{\omega}(k)
\]

\[
\dot{\theta}(k) = \ddot{\theta}(k) = \dddot{x}(k)
\]

For this study a simple state-feedback control law is implemented by:

\[
u(k) = -K \dddot{x}(k) + K_s r(k)
\]

The values of \(K\) are chosen so that the closed loop system dynamics (given by (11)) has a rise time of 10s, deemed sufficient taking into account that physical constraints would constantly lead to actuator saturation if higher feedback gains were used. The corresponding gain values are \(K = [-0.1500 0.0950 10.1487]\). \(K_s\) is a scaling gain selected \textit{a posteriori} to ensure that the steady-state gain of the closed-loop system (given by (11) and (9)) is unity.

\[
\dddot{x}(k+1) = (\dddot{\dot{x}} - BK)\dddot{x}(k) + B K_s r(k) + \dddot{\omega}(k)
\]

The simulation of the trajectory undertaken by the vehicle, shown in Fig.1, was obtained using the aforementioned guidance and control laws, the position of the vehicle being updated at each sampling instant according to

\[
x_{USV}(k+1) = x_{USV}(k) + v T_s \cos\left[\sqrt{2} (\theta(k) + \theta(k+1))\right]
\]

\[
y_{USV}(k+1) = y_{USV}(k) + v T_s \sin\left[\sqrt{2} (\theta(k) + \theta(k+1))\right]
\]

with \(v = 1.5 \text{ ms}^{-1}\) and initial conditions of position, heading angle, and state vector given by:

\[
x_{USV}(0) = 0, y_{USV}(0) = 0, \theta(0) = 0, x(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}
\]

Advertently during this process, a change in the steering dynamics of the vehicle was prescribed upon its arrival at the first way-point as follows:

\[
A = A + 0.05 |A|, B = B + 0.05 |B|, C = C + 0.05 |C|
\]

where it is understood that the absolute value is taken element-wise; that is, all the coefficients of the deterministic model were increased by 5%. It is to be noted nevertheless that the autopilot was furnished with the true (augmented) state-vector of the system at all time, and that on occurrence of (15) the controller gains were recalculated in accordance with the changed vehicle dynamics. Hence, no detrimental effect is apparent in the tracking capability of the vehicle due to the alteration of its steering dynamics.

In practice however, the change in the system’s dynamics may not be known or predictable, nor the state-vector measurable. For the system (1) to (3), the components of the state-vector have no particular correspondence with physical quantities since the model was obtained from input-output data alone.

C. Kalman filtering

The vehicle is equipped with a low-cost microelectromechanical gyroscope that measures the yaw-rate at 1Hz with an rms error of 0.1 deg∙s⁻¹, i.e. according to the model

\[
z(k) = y(k) + \nu(k)\quad \nu \sim N(0, R);
\]

\[
R = \text{cov}(\nu) = 0.1^2 \text{deg}^2 \text{s}^{-2}
\]

Based on the vehicle dynamic model (1) to (3) and the gyro measurement, the Kalman estimate of the state-vector can be calculated using (17) to (21) given initial estimates of the state-vector and error covariance [3].

KF equations:

\[
\hat{x}(k | k - 1) = A \hat{x}(k - 1 | k - 1) + Bu(k - 1)
\]

\[
P(k | k - 1) = A P(k - 1 | k - 1) A^T + Q
\]

Kalman gain:

\[
K(k) = P(k | k - 1) C^T \left( C P(k | k - 1) C^T + R \right)^{-1}
\]

Correction:
\[
\dot{x}(k | k) = \dot{x}(k | k-1) + K(k)[z(k) - C \dot{x}(k | k-1)]
\]  
\[
P(k | k) = [I - K(k)C]P(k | k-1)
\]  
(20)
(21)

Estimates of the augmented state-vector, yaw-rate and heading angle then ensue straightforwardly ((22) to (24)).

\[
\dot{x}(k) = \dot{x}(k)
\]

\[
\hat{\theta}(k) = \hat{\theta}(k-1) + C \dot{x}(k-1) + Du(k-1)
\]

\[
\hat{y}(k) = C \dot{x}(k) + Du(k)
\]

(22)
(23)
(24)

For the previously generated trajectory (Fig.1), the KF estimates (assuming known initial system state) of the yaw-rate and heading angle are shown in Fig.2, along with the corresponding true values of these, and simulated measurements of the yaw-rate. The controller action is also shown. It can be observed how after an initial period of ~70s (by which time the vehicle has just crossed the first way-point), the KF heading estimate starts to drift away from the real heading, due to mismatches between the estimated and real yaw-rates as the KF continues to predict based upon the initial model.

If in fact the KF state-estimate rather than the true state-vector is fed-back to the controller, the path travelled by the vehicle is a different one altogether. It is shown in Fig.3, wherefrom it is apparent that the autonomous system struggles significantly during the latter, more demanding stage of the course, failing to deliver the vehicle to the last three way-points (it should be said that the guidance system has additional incorporated logic that decides when a way-point is considered to have been missed, whereupon it shifts target to the following way-point).

\[
M^* \in M^1 = M \pm 0.05|M| = [M - 0.05|M|, M + 0.05|M|]
\]

(25)

D. Interval Kalman filter

Assume that even though the initial vehicle steering dynamics may vary in an unforeseeable manner, bounds to these variations can be determined. For example, that the dynamics will always be representable by a model whose coefficients lie within 5% of the initial, nominal model, that is,

\[
M^* \in M^1 = M \pm 0.05|M| = [M - 0.05|M|, M + 0.05|M|]
\]

for \( M \in \{A, B, C, D\} \). The model given by

\[
x(k+1) = A^1 x(k) + B^1 u(k) + \omega(k)
\]

\[
y(k) = C^1 x(k) + D^1 u(k)
\]

(26)
(27)

is then an interval model whose coefficients are intervals, and which contains all the point-valued models centred around the nominal model (1) to (3).

It has been proposed that for such interval systems the interval Kalman filter (IKF) can be used to obtain estimates of the state-vector [4]. The IKF equations are identical to those of the ordinary KF ((17) to (21)) but with interval-valued elements. The state-vector and error covariance estimates are now interval-valued, as are all derived quantities, although the measurement \( z(k) \) retains its point-valued nature as it is obtained directly from the sensor at each sample-time rather than through (16) and (27).

The IKF equations require that the same operations carried out using ordinary point-valued arithmetic be defined for interval quantities. Interval arithmetic (IA) operations were defined by Moore [5] in such a way that calculations on interval values yield further intervals which guarantee to enclose (though not necessarily be equal to) the range of possible results, a fundamental quality known as the inclusion property. As a consequence, the IKF estimate is guaranteed to contain the estimate of all ordinary KFs based upon point-valued models contained within the interval system.

Using the initial values

\[
\hat{x}^1(0) = [-0.005, 0.005]; P^1(0) = [0.0, 0.0]
\]

(28)

and the rules of IA as defined in [5], the computed IKF yaw-rate estimates \( \hat{y}^1(k) \) for the way-point simulation of Fig.1.
Fig. 4. IKF yaw-rate estimates and gyro measurements for the way-point tracking mission with state-feedback control based upon knowledge of true state-vector and system dynamics.

are shown in Fig. 4. (Note that only the upper and lower boundaries of the intervals $\dot{y}^i(k)$ are shown). The yaw-rate measurements, shown in Fig. 2, are replicated as well for reference.

It is quite apparent that the widths of the IKF estimates expand rapidly, resulting in bounds that surpass the maximum and minimum machine representable numbers, from which point on the simulation cannot continue (approximately at 60s in Fig. 4).

III. ELLIPTOIDAL INTERVAL ARITHMETIC

A. The Wrapping Effect

The overly conservative nature of direct IA is easily exposed. Consider for example the product $\alpha(\beta - \beta)$ where $\alpha = [-1,1]$ is an interval, and $\beta$ a real variable; then expanding the brackets leads to

$$\alpha(\beta - \beta) = [-1,1] \times (\beta - 1) \times \beta - [-1,1] \times \beta = [-\beta, \beta] - [-\beta, \beta] = [-2\beta, 2\beta]$$

(29)

whereas clearly the sharp result is zero. The cause of this overestimation can be attributed to the fact that after the expansion, the arithmetic did not remember that both operands represent the same variable. This memory-less nature of IA is further exacerbated for vector operations as what is known as the wrapping effect.

To understand this effect, consider the repeated anticlockwise rotation of the interval vector $x^i = [-2, 2]$ by $20^\circ$, i.e.,

$$x^i := Ax^i, \quad A = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}, \quad \alpha = 20 \times \frac{\pi}{180}$$

(30)

The left column of Fig. 5 depicts this process. The initial $x^i$ is represented by a box (rectangle, in the 2-D case), which after being rotated must then be enclosed by another rectangle with sides parallel to the coordinate axes for it to be representable as an interval vector, thus adding points to the enclosure that do not strictly belong to the rotated set. With each new rotation, this process is repeated, resulting in an ever larger rectangle. By contrast, if the initial rectangle is enclosed within an ellipsoid (or ellipse, in the 2-D case), as shown in the right column of Fig. 5, then the successive rotation of this ellipse maintains a constant enclosed area, since $\det(A) = 1$. Note that this technique can be applied to any affine transformation of a vector, since affinities transform ellipsoids into ellipsoids.

Extending this idea, Neumaier [6] has developed a methodology by which it is possible to compute ellipsoidal bounds of a general interval affine transformation of an interval vector $x^i$,

$$x^i := A x^i + b^i$$

(31)

given an ellipsoidal enclosure of the initial interval vector. Fig. 6 illustrates the application of this method to the case of an interval rotation given by

$$A^i = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

(32)

Fig. 6. Illustration of repeated interval rotation of a 2-D interval vector, using direct and elliptical IA (single rotation and ten successive rotations).
where $\gamma \pm \phi$ is used to indicate the interval $[\gamma - \phi, \gamma + \phi]$. It is apparent that although the elliptical enclosure does increase in size with repeated iterations of the transformation (as it must, since $A^r$ contains matrices with determinant larger than unity) it does so to a much lesser degree than the expansion of the box enclosures obtained via direct IA, hence reducing the wrapping effect.

B. Application of Ellipsoidal Arithmetic to the IKF equations

The IKF equations propagate the state and error covariance estimates, and it is possible to view these as iterative interval affine transformations. For the propagation of the state estimate, (17) is already in the form of an affine transformation of $\hat{x}(k\mid k-1)$, and (20) can be written as

$$\hat{x}(k \mid k) = [I - K(k)C] \hat{x}(k \mid k-1) + K(k)z(k)$$  \hspace{1cm} (33)

which is again an affinity, considering that $K(k)$ has been previously obtained from (19).

As far as the error covariance matrices, these can be written out as vectors. For a 2x2 system, (18) can be written using indicial notation as:

$$p_{i,j}(k \mid k-1) = a_{i,j}p_{i,m}(k \mid k-1) + q_{i,j}; \ i, j, m = 1, 2$$  \hspace{1cm} (34)

Define $\tilde{p}(k \mid k-1), \tilde{p}(k \mid k-1), \tilde{q} \in \mathbb{R}^4$ as

$$\tilde{p}_{2(i-1)+j}(k \mid k-1) = p_{i,j}(k \mid k-1)$$

$$\tilde{p}_{2(i-1)+m}(k \mid k-1) = p_{l,m}(k \mid k-1)$$

$$\tilde{q}_{2(i-1)+j} = q_{i,j}$$

Then the components of the error covariance are propagated by the affine transformation of the vector $\tilde{p}$ according to

$$\tilde{p}_r(k \mid k-1) = g_{r,s} \tilde{p}_s(k \mid k-1) + \tilde{q}_r,$$ with $r = 2(i-1) + j, s = 2(l-1) + m,$ $g_{2(i-1)+j,2(l-1)+m} = a_{i,j} a_{j,m}; \ r, s = 1,..,4; \ i, j, l, m = 1, 2$  \hspace{1cm} (38)

Similarly for (21), define $\tilde{p}(k \mid k) \in \mathbb{R}^4$ as

$$\tilde{p}_{2(i-1)+j}(k \mid k) = p_{i,j}(k \mid k); \ i, j = 1, 2$$

(\text{with} \ \tilde{p}(k \mid k-1) \ \text{defined as in (38)}); \text{then,}

$$\tilde{p}_r(k \mid k) = h_{r,s} \tilde{p}_s(k \mid k-1)$$ with $r = 2(i-1) + j, s = 2(l-1) + j,$ $\tilde{p}_{2(i-1)+j,2(l-1)+j} = [I - K(k)C]k_i; \ r, s = 1,..,4; \ i, j, l, m = 1, 2$  \hspace{1cm} (40)

Hence, the ellipsoidal arithmetic method may be applied to the propagation of the state estimate and error covariance of the IKF. Let an IKF thus implemented be denoted as eIKF. For the way-point tracking mission depicted in Fig.1, simulated using state-feedback control based on the true state-vector and knowledge of the updated system dynamics, the eIKF estimate of the yaw-rate is given in Fig.7, along with the IKF estimate previously plotted in Fig.4. Unlike the latter, the former provides interval bounds that do not diverge over the length of the simulation.

IV. WEIGHTED AVERAGE OF INTERVAL BOUNDS

Using the technique described in [7], an artificial neural network (ANN) is trained to predict a good weighted average of the eIKF bounds at each time sample from residual data of a KF.

Firstly, a training data-set is constructed by simulating another way-point following mission. This mission, shown in Fig.8, is selected to contain representative USV manoeuvre requirements. The initial steering dynamics given by (1) to (3) is altered several times during the course of the mission, but always within the interval model (25) to (27). The control law is based on feedback of the actual state vector, and is updated in accordance with each new modified dynamics to ensure that the path is adequately tracked.

An eIKF, together with a KF based upon the initial dynamics, and an ideal KF (always based upon the updated dynamical model) are then simulated and their estimates of the yaw-rate are shown in Fig.9. The innovations sequence of the former, non-ideal KF, is plotted as well, together with the desired weights, computed so that the weighted average of the eIKF bounds coincides, at each time sample, with the ideal KF estimate, i.e.

$$w_d(k) = \frac{\hat{y}_{\text{ideal-KF}}(k) - \inf(\hat{y}_{\text{eIKF}}(k))}{\sup(\hat{y}_{\text{eIKF}}(k)) - \inf(\hat{y}_{\text{eIKF}}(k))}$$  \hspace{1cm} (41)

where $\inf$ and $\sup$ denote the infimum and supremum respectively.

The objective is basically to construct a mapping from the innovations to the desired weight. This mapping is realised by training an ANN. Moreover, it was shown in [7] how the incorporation of the IKF widths as inputs could aid

![Fig.8: Way-point following mission for generating training data.](image-url)
this mapping. Hence, in this case, the eIKF widths are computed as well and used as input to the network along with the KF innovations, the target values for the training process being the desired weights. A multi-layer perceptron type network with input time-delays was found to achieve a good fit to the target, and its output is shown too in Fig.9.

V. RESULTS AND CONCLUSIONS

Once trained, the ANN can be used in any new mission to predict the desired weight with which to compute a weighted average of the bounds of the eIKF state-vector and output estimates; let these weighted estimates be referred to as wIKF estimates. For the original way-point tracking mission (Section II), simulation results in which the state-feedback controller uses the wIKF state estimate rather than the true system dynamics are shown in Fig.10 and Fig.11. The control gains however are not updated to take into account the modified steering dynamics after the first way-point. Nevertheless, as shown in Fig.10, the mission is completed with relative ease. The ANN is effectively conferring a degree of intelligence to the navigation system which is responding to changes in system dynamics, even though the interval filter has no knowledge of the precise system dynamics. As seen in Fig.11, the wIKF estimates deviate a lot less from the true values than do the ordinary KF estimates due to biased predictions. Table 1 collates a few statistics to compare mission performance, benchmarked against the hypothetical ideal scenario in which the system states are available and the control law is based always on the true system dynamics. It clearly shows a distinct improvement when using the wIKF estimates rather than those of an ordinary KF.

The methodology developed in this study for designing a robust navigation system for the Springer USV can of course be followed to design such systems for other USVs, and indeed can be applied to other systems as well, such as ground or underwater vehicles.

<table>
<thead>
<tr>
<th>TABLE I. COMPARISON OF MISSION PERFORMANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of way-points reached</td>
</tr>
<tr>
<td>Total distance travelled (m)</td>
</tr>
<tr>
<td>Maximum deviation (m)</td>
</tr>
</tbody>
</table>

a Control based on true state-vector and knowledge of true system dynamics

REFERENCES


Subsea cable tracking by an unmanned surface vehicle.

Subsea cable tracking by an unmanned surface vehicle

Tomasz Szyrowski*1, Amit Motwani1, Sanjay K Sharma1, Robert Sutton1 and Gareth A Kennedy2
1School of Marine Science and Engineering, Plymouth University, Plymouth, UK
2Camborne School of Mines, Exeter University, Cornwall Campus, Penryn, UK

Received 29 July 2014; Accepted 4 December 2014

Abstract

Subsea cable localisation is a demanding task that requires a lot of time, effort and expense. In the present paper the authors propose a methodology that is automated and inexpensive, based on magnetic detection from a small unmanned surface vehicle (USV) and the use of a batch particle filter (BPF) algorithm. A dynamic path planning algorithm for the USV is also developed so that adequate samples of the magnetic field readings can be gathered for processing by the BPF. All of these elements work together online as the cable is tracked, which was demonstrated in a simulated mission.

Keywords: batch particle filtering, subsea cable tracking, weighted interval kalman filtering, fuzzy logic

1. Introduction

Technological advancement has brought with it an increasing number of maritime transportation links and offshore structures, and these are set to increase. For example, the UK government has recently announced plans to develop what will be the world’s largest offshore wind farm off the coast of Suffolk, UK (Guardian, 2014). Such farms require arrays of subsea cables to link each of the turbines together as well as to offshore substations. Detection of such cables is important for periodical inspection and maintenance.

Cable detection methods that are presently used have come a long way, but their scope is still fairly limited. Optical inspection, hydroacoustic localisation and magnetic detection are the three main methods for subsea surveys. The main limitation of the first two are that they fail to detect cables that are buried or hidden under plant growth, as is often the case in coastal and shallow waters (Szyrowski et al., 2013a).

Detection of subsea ferromagnetic objects is mainly based on mathematical inversion methods (Cowls and Jordan, 2002; Won, 2003). Two or more magnetic detectors separated some distance from one another are used to measure the magnetic field (MF) emitted by the ferromagnetic object or induced current, whereby its location can be determined through triangulation. However, this method is only accurate up to a range of 3m (Takagi et al., 1996; Kojima et al., 1997; Szyrowski et al., 2013b). This reduced range translates into requiring a diver or expensive underwater vehicle to perform the survey.

In the present paper, a solution to this problem is proposed whereby the detection of a subsea cable is carried out by an autonomous vehicle from the surface. To achieve this, the authors have developed a precise vehicle navigation system, as well as a guidance algorithm that directs the vehicle along required pathways that enable a stream of measurements of the MF to be collected. From this data, a novel localisation algorithm based on particle filtering (PF) is applied to determine the location of the source of the MF.

Instead of using a single measurement from each of the distance-separated magnetic detectors and then trying to determine the source by inversion methods, the batch particle filter (BPF) uses several measurements of the MF taken in the vicinity of the source. In order to acquire these, the unmanned surface vehicle (USV) inspects the area where the cable is thought to lie. Once a meaningful reading of an MF is obtained, a specific survey
path that crosses the cable is dynamically planned and further measurements are gathered.

A fuzzy logic algorithm is used to distinguish between meaningful readings that originate from the MF induced by the cable and the surrounding magnetic noise. When the cable has been crossed and no more meaningful readings are obtained, the set of readings is given to the BPF, which then estimates the exact crossover point of the cable based on all the readings obtained. Using this information, the position of the cable some distance downstream is estimated, and the vehicle’s survey path is re-planned.

The USV must be able to trace the planned survey path and, above all, localise itself and accurately determine its heading at every instant. The precise localisation not only enables smooth autonomous navigation, but also gives the MF readings taken along directions relative to the vehicle’s own heading. In order to compare the successive readings, they must all be described in a common global reference frame before using them to determine the magnetic source. Hence, a novel, robust heading estimation technique based on interval Kalman filtering has been applied to estimate the USV’s heading.

The BPF algorithm has been tested offline on real subsea cable survey data gathered during a manned expedition. Although the MF in this case was induced by an alternating current flowing through the cable, the method can be applied to localising any ferromagnetic object by equipping the USV with an eddy current inducing coil for generating a magnetic response in the object. The integration of the BPF source estimation method with autonomous USV navigation and path planning has been tested on computer simulations, with sea trials being planned for the future.

This approach offers advantages with respect to conventional methods, for example, enabling operations in hazardous environments without risking divers’ safety or needing to employ manned ships. Moreover, autonomous tracking of the dynamically planned path is carried out accurately and efficiently, avoiding delays in the control loop that would inevitably exist in relaying information continuously regarding the path updates to the crew of a manned vessel, saving both time and costs.

The rest of the present paper is organised as follows: the BPF algorithm and its motivation as an efficient tool for subsea ferromagnetic object localisation is presented in section 2. The fuzzy logic algorithm used for discriminating meaningful readings from noise is also described in section 2. Section 3 then discusses the autonomous operation of the USV, with emphasis on the robust heading estimation procedure using the so called weighted interval Kalman filter (WIKF). Section 4 details the simulation of a cable-tracking mission, and finally conclusions and future objectives are discussed in section 5.

2. Selective batch particle filtering for ferromagnetic object localisation

In order to detect ferromagnetic objects, these must emit an MF. An MF in ferrous objects originates from electric currents flowing within them. In objects such as cables or pipes, an electric current can be injected, thereby producing an MF. Where this is not possible, an alternating MF generated by an external agent can induce eddy currents in the body of the object, which then emits a secondary MF that can be detected (Cowls and Jordan, 2002; Tumanski, 2007).

In the maritime subsea environment, the propagation of the MF is highly affected by the conductivity of the sea water, which depends on its salinity (Bogie, 1972; King, 1989). Currently used methods for subsea cable localisation, which are short range and applicable only within distances of up to 3m from the source, assume that the conductivity of the water is uniform and can be neglected (Szyrowski et al., 2013b). They also assume the MF decay, as a function of the distance from the source, follows a simple decay function, and hence the difference between two readings separated by a known distance allows inference of the distance of the readings from the source. However, Cowls and Jordan (2002) have pointed out that this assumption is not always true: although the signal strength generally decays as $r^{-3}$, it can include some variation and can be difficult to calculate precisely.

The strength of the MF generated by a subsea cable on the water surface can be modelled as a scalar field, as shown in Fig 1. In the case of a cable with small curvature, the MF generated by the cable at any point in space $(P_i)$ can be approximated by that owing solely to the point of the cable $(S_i)$ that is closest to $P_i$, i.e.:

$$B_i = \frac{\mu_0 I}{4 \pi r_i^3} \left( d_i \times [P_i - S_i] \right) \quad (1)$$

where $d_i$ is the direction of the cable at point $S_i$; $r_i$ is the shortest distance to the cable; $\mu_0$ is a attenuation parameter; $I$ is the electric current. In a small area, the parameter $(a)$ can be considered to represent an averaged attenuation; hence, for a short section of cable, the term $\frac{\mu_0 I}{4 \pi r_i^3}$ can be grouped into a single constant $(c)$. Note that the contribution to the MF at $P_i$ is mostly owing to the section of cable around...
$S_k$, and is not much affected by portions of the cable further away where the attenuation might be different. Hence, MF can be described as:

$$B_k = c \frac{(d_k \times [P_k - S_k])}{r_k^3}$$  \hspace{1cm} (2)

The problem of determining the location of the source ($S_k$) from measurements of the MF at various sample points also must include estimating the local value of the parameter $c$. This problem can be thought of as a regression of the various measurements of the MF vector at various sample points on the sea surface onto the model described by Equation 2 parameterised by source points ($S_k$), direction vectors ($d_k$) and the averaged attenuation ($c$) (Fig 2).

If it can be considered that the sample points all correspond to the same source (i.e. that they are obtained from crossing the cable perpendicularly), then the problem is simplified to determining one source (in addition, to the average attenuation ($c$)). This scenario is depicted in Fig 3, in which the plane ($\Pi_k$) is perpendicular to the cable at the intersection point ($S_k$). Then for each sample point ($P_k$) on the surface along the line ($l_k = W \cap \Pi_k$), $S_k$ is the closest point of the cable to it. In this case, the MF vector $B_k$ at each $P_k$ is contained within the plane $\Pi_k$.

**Fig 1:** Distribution of magnetic field from long ferromagnetic wire

**Fig 2:** MF vector field in sample points
The successive MF measurements along \( l_k \) from a horizontal coil whose axis is aligned with the direction of \( l_k \) are then established. The readings are generated according to Equation 2 for simulation purposes, using values of current and attenuation measured from previously conducted trials on the Baltic Sea (Szyrowski et al., 2014), and generated at regular intervals along \( l_k \). This is consistent with taking samples at regular time intervals from the USV moving along \( l_k \) at constant speed.

The coil readings can be expressed as:

\[
z_k = h(x_k) + v_k = \begin{bmatrix} C_H + v^H_k \\ C_V + v^V_k \end{bmatrix} = \begin{bmatrix} B_k \cdot h_k + v^H_k \\ B_k \cdot z + v^V_k \end{bmatrix}
\]  \tag{3}

where \((C_H + v^H_k)\) corresponds to the output from the horizontal coil, and \((C_V + v^V_k)\) corresponds to the output from the vertical coil; \(v^H_k\) and \(v^V_k\) are random measurement noises associated with the readings; \(h_k\) represents a unit-length vector in the direction of the horizontal coil axis, which is assumed to be the same as the direction of the line \( l_k \); and \(z\) is a vertical unit-length vector. The total MF is the vector sum of these two components.

The coil measurements in terms of the locations of the sampling point, cable source and the average attenuation parameter can be obtained simply by substituting Equation 2 into Equation 3, and is given by:

\[
z_k = \begin{bmatrix} \frac{d_{k+1} \times |P_k - S_k|}{l_s} \cdot h_k + v^H_k \\ \frac{d_{k+1} \times |P_k - S_k|}{l_s} \cdot [0 0 1] + v^V_k \end{bmatrix}
\]  \tag{4}

Having assumed that \( d_k \) is perpendicular to \( \Pi_k \) – i.e. that the vehicle moves along a path perpendicular to the direction of the cable – Equation 4 reduces to:

\[
z_k = \begin{bmatrix} |B_k| \cos(\theta) + v^H_k \\ |B_k| \sin(\theta) + v^V_k \end{bmatrix}
\]  \tag{5}

where \(\theta\) is the angle between \(B_k\) and \( l_k \). Equation 5 is used in the implementation of a BPF to estimate the source \( S_k \) for a single crossover line \( l_k \).

Particle filters (PF), introduced by Gordon (1993), are a tool to estimate the posterior probability density function (PDF) of state variables from observations (Crisan and Obanubi, 2012; Fallon and Godsill, 2010). PFs use a recursive scheme to approximate the PDF by a set of random samples called particles, which tend to concentrate in regions of high probability density, serving to approximate the true PDF. The operation of the PF algorithm is schematically represented in Fig 4. An initialisation step generates random particles representing a hypothetical MF they represent and the experimental data. The particles are resampled and regenerated until they converge to the area which guarantees a satisfactory error. (The reader interested in a more detailed description of PF may consult Szyrowski et al. (2014).)

In the BPF algorithm, each particle, which is representative of a possible location of the cable source, has an associated weight or probability, and the most plausible location is given by the particle with the largest weight. The rest of the particles in the set tend to cumulate around this solution. In a regeneration stage, only a quarter of the particles, those with the highest weights, are maintained. Each one

---

**Fig 3:** Basic cable-tracking parameters

**Fig 4:** Particle filter algorithm
of these particles then generates three new particles randomly in the space surrounding it. The higher the weight of the mother particle, the smaller is the surrounding area in which the new particles are allowed to be generated. Thus, particles with smaller weights will have their children cover a wider area, whilst those particles with high weights will conceive particles that are very close to themselves. After the regeneration step, the algorithm starts a new iteration where for each particle, the MF and the resulting theoretical coil readings are generated. According to how well each of these fit the actual readings, their weights are then computed.

In order to start the algorithm, a set of initial particles must be chosen. First, a region of interest (ROI) is established. Then, the horizontal coil readings of the MF should, in theory, reach a peak value when the reading is taken directly above the cable – that is, when the path \( l_k \) apparently intersects the cable when viewed from above. At this point, the theoretical reading from the vertical coil should be zero. Based on this observation, the ROI is chosen as a rectangular area below the sample point with the maximum difference between horizontal and vertical coil reading. In practice, this reading might not necessarily correspond to a measurement taken directly above the cable, as the readings incorporate a stochastic measurement noise. Thus, the ROI is extended so as to cover a horizontal distance corresponding to the abscissas of the five previous and five posterior measurement locations as well.

During an actual survey, the depth of the water column is measured with an echo-sounder at each sample point. It is conceivable that the cable could be buried up to 3m under the seabed, and that certain sections may be suspended in the water above it. Hence, the height of the ROI is taken from 3m below the seabed to 2m above it. In the simulation carried out herein, the seabed was assumed to lie at a constant 10m below the surface. For one crossover line \( (l_k) \), the simulated coil readings of the MF and the ROI established from these are depicted in Fig 5.

Once the ROI has been established, \( N \) particles are generated randomly within it. In the simulation carried out, the number of particles was set to \( N = 100 \). The locations of the initial particles are shown in Fig 6. Each of these represents a hypothetical location of the cable source. In addition, to each of these particles an averaged attenuation parameter \( c \) is also assigned. For the BPF algorithm, each particle is assigned a value of \( c \) randomly chosen within a range of 10% of the nominal value set experimentally.

Each of the particles is then assigned a weight. In order to do this, the theoretical coil readings for each particle (hypothetical cable source location and parameter \( c \)) are generated according to Equations 2 and 5. The mean square error between these and the actual readings is computed, normalised and assigned to the corresponding particle. This completes the initialisation of the BPF algorithm.

The BPF then takes on an iterative character. In each iteration, 75% of the particles are regenerated from the 25% with the largest weights. For each of these progenitor particles, three new particles are generated by adding Gaussian noise to their position and value of \( c \). The amplitude of the noise added is inversely proportional to the engendering particle’s weight. The particle with the largest weight (and hence whose MF distribution along the sampling points coincides most closely to the measured one) generates particles closer to itself than do those particles with a smaller weight that are not quite so close to the true source, and hence spread their seeds further afar.
The new set of particles then undergoes the same procedure of weight assignment, and the whole process is repeated. During this process, the particles tend to cumulate in the region of highest probability of the location of the true source. It should be recalled that each of the particles also represents a certain value of the parameter ($c$). The particle with the highest weight represents not just the location of the true source, but also the actual value of the parameter ($c$). Hence, the BPF algorithm gives not only an estimate of the most likely location of the true cable source, but also of the locally averaged attenuation rate. One of the key advantages of this procedure is that it is able to locate the source even without a correct prior estimate of the parameter ($c$), which is not possible using other well-known techniques such as those based on Kalman filtering.

Observation of Figs 5 to 8 reveals that much of the coil readings actually correspond to background magnetic noise rather than to any meaningful measurements of the MF emitted by the cable. In order to both reduce the computational burden on the BPF and increase its efficiency, it becomes necessary to filter out the unwanted readings. This is accomplished with discriminatory filter based on fuzzy logic decision-making. It is based on observing that the measurements are meaningful when the magnitudes of the readings are consistently larger than some threshold value. Based on this, in order to filter out some of the noise from the readings, a simple moving average (SMA) of both the horizontal and vertical coil readings is computed. The average of these two SMAs is then obtained and used as input to the membership functions depicted in Fig 9.

The fuzzy decision-making is based on the following rules:
- Rule 1: If SMA negative, then $ff$ is negative;
- Rule 2: If SMA is zero, then $ff$ is zero; and
- Rule 3: If SMA is positive, then $ff$ is positive;

where the variable $ff$ is the output of the fuzzy classifier, and its crisp value is obtained from the output membership functions shown in Fig 10.
The value of $ff$, or fuzzy flag, is used as a flag to indicate whether the coil readings are meaningful. Only those readings for which $ff$ is greater than zero are considered by the PF. The two vertical lines in Fig 5 represent the range in which the fuzzy flag is positive for the readings shown in the graph.

The value of the fuzzy flag $ff$ is used to indicate when the coil readings are meaningful. It is initially set to zero, as the first readings are taken at the start of a crossover line ($l_k$). When the vehicle is within a certain distance of the cable and the readings start to increase, the flag changes to 1, indicating that these readings should be stored and later passed on to the BPF. Eventually, after the vehicle has crossed the cable, the flag returns to zero, and the stored readings are sent to be processed by the BPF. The instants at which the flag is raised and then lowered again are shown as two vertical red lines in Fig 5. As will be described in the next section, the flag also serves a dual purpose, which is to indicate when the USV may move onto the next cable crossover line.

3. Autonomous surface vehicle navigation

The number and type of potential applications for autonomous USVs has increased dramatically in recent times with the availability of low-cost sensing devices (Motwani, 2012). The Springer USV, shown in Fig 11 (Naeem et al., 2006), is a 4m-long twin-hull catamaran built at Plymouth University in the UK and is one such vehicle that is being used as a test-bed for developing robust intelligent navigation, guidance and autopilot systems. In particular, a novel heading estimation technique developed for Springer is described and used herein for underwater cable localisation and tracking.

Accurate heading estimation is, of course, important for autonomous navigation, since the vehicle’s autopilot acts on it to steer the vehicle onto the desired course. However, it is of particular importance for the cable localisation method described in the
present paper. This is because the measurement of the MF obtained from the horizontal coil depends on the direction of its axis, and so this direction (which could be chosen to coincide with that of the vehicle’s heading if the coil is mounted parallel to the vehicle’s longitudinal axis) needs to be known.

For localisation, the Springer uses a GPS receiver, while the heading estimation system described here uses a low-cost microelectromechanical system (MEMS) gyroscope and a dynamic steering model of the vehicle. The gyroscope provides a measurement of the vehicle’s turning rate, which is subject to some measurement noise and can be used on its own to determine the heading of the vehicle. However, successive integration of the gyroscope output eventually leads to integration drift.

As mentioned, a dynamic steering model of the vehicle was also used. The propulsion of the vehicle is based on two battery-driven trolling motors, one on each hull. Steering of the vehicle is controlled by applying a difference in motor speeds, whereas the overall speed of the vehicle can be changed by varying the average speed of the two motors. Trials have been conducted wherein, maintaining a constant speed, the vehicle was made to carry out various turning manoeuvres. Data of the applied differential motor speed and vehicle turning rate were recorded, and through system identification (SI) models of the form in Equations 6 to 8 were obtained. Basically, the model allows one to predict the turning rate of the vehicle based on the applied differential motor speed over time. In particular, for a constant vehicle speed of 1 ms⁻¹, the model obtained is characterised by the values given in Equation 8:

\[
\begin{align*}
x(k + 1) &= A x(k) + B u(k) + \omega(k) \\
y(k) &= C x(k) + D u(k)
\end{align*}
\]

\[
A = \begin{bmatrix} 0.8 & -0.2225 \\ 1 & 0 \end{bmatrix}, \\
B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \\
C = \begin{bmatrix} 0 & 0.004225 \end{bmatrix}, \\
D = 0; \\
T_s = 1 \text{s}, \\
\omega \sim N(0, Q); \\
Q = \text{cov} (\omega) = 0.01 \times \text{diag} [1, 1]
\]

where \( u(k) \) is the differential motor speed input in rpm, and \( y(k) \) is the rate of change of heading of the vehicle; \( \omega \) represents a random input disturbance; and \( T_s \) is the sampling period. In addition, actuator limitations impose the following constraints on the above model: \( |u(k)| \leq 1200 \text{rpm} \) and \( |\Delta u(k)| \equiv |u(k) - u(k-1)| \leq 500 \text{rpm} \).

Both the steering model and gyroscope readings can be used to determine the current heading of the vehicle. However, they can be combined in a Kalman filter (KF) for improved accuracy. Based on the vehicle dynamic model Equations 6 to 8 and the gyro measurement, precise to 0.1 deg-s⁻¹ root mean square (RMS) at 1Hz sampling, i.e. with measurement \( z(k) \) according to:

\[
z(k) = y(k) + v(k); \\
v \sim N(0, R); \\
R = \text{cov}(v) = 0.1^2 \text{ deg}^2 \text{s}^{-2}
\]

The KF estimate of the state vector \( \hat{x}(k) \) is obtained by applying Equations 10 to 14 given initial estimates of \( \hat{x} \) and error covariance \( \text{P}(k) = \text{var} \{x(k) - \hat{x}(k)\} \) (Simon, 2006):

**Prediction:**

\[
\hat{x}(k|k-1) = A \hat{x}(k-1|k-1) + B u(k-1)
\]

\[
P(k|k-1) = AP(k-1|k-1)A^T + Q
\]

**Kalman gain:**

\[
K(k) = P(k|k-1)C^T \left\{ C P(k|k-1)C^T + R \right\}^{-1}
\]

**Correction:**

\[
\hat{x}(k|k) = \hat{x}(k|k-1) + K(k)[z(k) - C \hat{x}(k|k-1)]
\]

\[
P(k|k) = [I - K(k)C]P(k|k-1)
\]

The heading of the vehicle can then be obtained from the KF estimate of the state vector as follows:

\[
\hat{\theta}(k) = \hat{\theta}(k-1) + C \hat{x}(k-1|k) + D u(k-1)
\]

The estimated heading is used by the vehicle’s autopilot to generate the required differential propeller speed to steer the vehicle along the desired path. The vehicle’s guidance system, in turn, generates and updates the reference path of the vehicle via a series of waypoints. A detailed description of the waypoint tracking and autopilot systems used for the autonomous operation of Springer can be found in Annamalai et al. (2014).

Although the use of a KF avoids gyro integration drift and provides a theoretically optimal estimate of the heading (in a statistical sense), it fails if the vehicle model does not accurately reflect the true dynamics of the vehicle (Motwani et al., 2013). In the case presented here, accurate modelling of the steering dynamics via SI is utopian, and at best small modelling errors will arise owing to inhomogeneous sea and wind conditions, variations in payload, etc.

This difficulty is addressed by applying what is known as a weighted interval KF (WIKF), developed for systems with finite modelling uncertainty (Motwani et al., 2014b). Consider a system model, such as Equations 6 to 8, in which the model coefficients (elements of the matrices A, B, C and D) are not known precisely,
but are known to lie within certain bounds. Then, describing these coefficients by intervals rather than point-values, the resulting model is called an interval model. Based on this concept, the interval KF (IKF) was proposed as an extension to the KF for interval systems (Chen et al., 1997). The IKF equations mirror those of the standard KF, but operate on interval values instead using interval arithmetic. The state estimates provided by the IKF are also in the form of intervals rather than point-values.

Although the IKF provides optimal state estimates of interval systems, in practice, a single value is required that most closely matches the state of the true system. The technique used by the WIKF is to obtain this estimate as a weighted average of the IKF bounds. This weight in turn is predicted at each time-step by an adequately trained artificial neural network (ANN) from the sequence of residual data of a standard KF (Motwani et al., 2014a).

Firstly, one or several training missions are devised in which the USV dynamics used to simulate the vehicle’s motion is made to vary within certain bounds. The bounds are those of the interval model being proposed. Two KFs are simulated to obtain estimates during these missions. The first is a KF that uses a nominal point-valued model contained in the interval model. The second is a KF that uses the true vehicle dynamics at each instant (i.e. the dynamics used to simulate the vehicle’s motion). Because the latter is an ideal KF, its innovations comprise a white noise sequence. However, those of the former digress from being white insofar as the model used differs from the true vehicle dynamics.

In addition, an IKF is also simulated based on the interval model. It can be shown that the ideal KF estimate can be retrieved as a weighted average of the IKF estimate bounds, and this desired weight is calculated and stored at each time-step. Finally, an ANN is trained to match the innovations of the first KF during the mission with the desired weighting sequence. It has been shown that such a trained network can be used to predict adequate weights independently of the true vehicle dynamics and KF nominal model selected for generating the training data, as long as they lie within the interval model that describes the bounded uncertainty. A detailed account of the WIKF can be found in Motwani et al. (2014a).

The navigational effectiveness of the WIKF over the use of a standard KF for uncertain systems will be shown in the next section on the cable-tracking mission.

4. Cable source tracking

To track a subsea cable from the USV, the objective is to criss-cross it at right angles. In each crossover line, the MF is measured from the coils, and the BPF described in section 2 is applied to determine the location of the source below that crossover line. The vehicle then advances to the next crossover line, and again the source location below that line is determined. From a navigational point of view, the problem of projecting these crossover lines needs to be addressed.

From an initial estimate of a point of the cable ($\hat{S}_1$) and an initial estimate of the cable’s direction at that point ($\hat{d}_1$), the first crossover line ($l_1 = P_1Q_1$) is projected as a segment perpendicular to $\hat{d}_1$ from start point ($P_1$) to end point ($Q_1$) 50m on either side of $\hat{S}_1$ (Fig 12). These two waypoints are sent to the USV’s guidance system, and the vehicle advances at a speed of 1ms⁻¹ from $P_1$ to $Q_1$. Every 1s, the vehicle updates its location and heading estimate, and the coil readings are taken.

Although ideally the vehicle follows the straight line from $P_1$ to $Q_1$ with its heading aligned along $P_1Q_1$, this is not always the case as there may be surface currents and other environmental effects that distort the vehicle’s path. The estimate of the vehicle’s

![Fig 12: USV trajectory showing cable estimation process](image)

![Fig 13: Correction of horizontal coil reading](image)
heading is thus used to correct the horizontal coil reading of the MF for use by the BPF algorithm, which assumes the horizontal coil to be perpendicular to the cable’s direction at the source point $\hat{S}_i$. If $\alpha$ is the error between the heading angle of the USV and the direction of $P_iQ_i$ at some instant $k$ (Fig 13), then the measured horizontal coil reading, $HmV(k)$, is corrected according to Equation 16.

$$HmV(k) = \frac{HmV'(k)}{\cos(\alpha)}$$ \hspace{1cm} (16)

While initially the USV targets the end point $Q_i$ of $l_i$, when the fuzzy inference system (see section 2) lowers its flag, the coil readings along this line will no longer be meaningful. At this point, the BPF determines an a posteriori or corrected cable source point on $l_i$, namely $\hat{S}_i$, based on which an a priori estimate of the position of the source on the next crossover line is obtained as $\hat{S}_i = S_i + \rho \hat{d}_i$, where $\rho$ is a prescribed distance (30m in this case). The estimated direction of the cable at $\hat{S}_i$, $\hat{d}_i$, is set equal to $\hat{d}_i$. Having estimated $\hat{S}_i$ and $\hat{d}_i$, 30m downstream, the next crossover line ($l'_i = P_iQ'_i$) is projected as a segment perpendicular to $\hat{d}_i$ from start point $P_i$ to end point $Q_i$, spanning 50m either side of $\hat{S}_i$. There is now no further interest in the vehicle reaching $Q_i$, and therefore a new target $Q'_i$ on $l'_i$ is established just 10m ahead of the vehicle. From this, a path is generated between this new final point on $l_i$ and the initial point on $l'_i (P_i)$ via a Hermite spline, on which several intermediate waypoints are generated to provide a smooth turning path for the USV.

For a general line $l_i$, after the a posteriori estimate of the source location, $\hat{S}_i$, has been determined by the BPF algorithm, the a priori estimate of the position of the source on the next crossover line ($\hat{S}_{i+1}$) is obtained by extrapolating the last three source estimates ($\hat{S}_{i-2}, \hat{S}_{i-1}, \hat{S}_i$) with a parabolic function a distance of $\rho$ further downstream. The cable’s direction at $\hat{S}_{i+1}$, $\hat{d}_{i+1}$, is estimated to be that of the tangent to the parabola at that point. A simulation of the cable localisation and tracking process is shown in Fig 14.

Navigational data for said simulation are given in Fig 15. The USV heading estimation was based on the WIKF technique described in the previous section. In order to illustrate the robustness of this technique, an interval model centred around Equations 6 to 8 with $\pm 5\%$ uncertainty on all of the values was considered. It was assumed that the vehicle’s true dynamics was given by the upper boundaries of all the intervals. Estimates of the turning rate and heading angle obtained from the WIKF, an ideal KF (based on the true vehicle dynamics), and a nominal KF (based on the nominal model Equations 6 to 8) are shown in Fig 15a,b. Also shown are the IKF bounds, the true values of turning rate and heading, and the gyro measurements. For completeness, the differential thrust applied by the autopilot is shown in Fig 15c, the innovation sequence of the nominal KF that is fed to the trained ANN in Fig 15d, and the desired and ANN-predicted weightings for the IKF bounds in Fig 15e.

A quantitative comparison of the turning rate and heading errors of each of these three filters is given in Table 1. Both from the figure and the table, it can be observed how the nominal KF heading estimates are biased because of the incorrect vehicle model assumed. The average heading estimate error of the KF is almost four times that of the WIKF.
Fig 16 shows what the actual trajectory of the USV would be like if these headings were used for navigation, highlighting the importance of accurate heading estimates to minimise deviation from the desired path. Moreover, because of the incorrect heading data, the readings of the MF in the direction of the crossover line would not be obtained correctly, leading to an incorrect estimation of the cable source by the BPF.

5. Conclusion

The BPF algorithm for estimating the source of the cable was initially developed for surveys from manned platforms that attempted to estimate the cable’s position by successively criss-crossing its assumed path. This technique proved successful in practice as long as the platform was guided adequately. On the other hand, the WIKF was developed to provide the Springer USV with an accurate heading estimation system to enhance autonomous operation of the vehicle.

The present paper has proposed a methodology for autonomously carrying out subsea cable localisation from a USV using the BPF algorithm. To make this possible, an effective dynamic path planning algorithm to guide the USV has been proposed. It was also aided by a fuzzy-logic-based data discrimination
procedure that indicates when meaningful coil readings are being obtained. This allows the USV to move on to the next projected crossover line before completing the initially projected current line, saving time and operation costs.

In addition, the uncertainty and decision-making delays of manned navigation and guidance, which were observed as one of the main practical drawbacks in the cable localisation surveys that were carried out using the BPF, are greatly reduced by the degree of automation proposed in the present paper. From the point of view of accuracy, the robust USV heading estimation based on the WIKF means that the horizontal coil readings are more accurately projected onto the global reference frame. By providing only those coil readings that are actually meaningful to the BPF algorithm, it is able to converge more rapidly and reliably.

The method proposed here does not replicate the cable localisation method carried out from a manned boat, but improves on its effectiveness, reducing time and costs, and of course, without the safety concerns of manned operation. In the simulation shown here, the USV was able to track the cable even though the initial assumption of the source was over 20m away from the true source, and the initial assumed cable direction was over 10 degrees off. A limitation to the approach described here is the assumption that the object being tracked does not contain pronounced curvatures, since the projected crossover lines are assumed to be perpendicular to the cable based on a priori estimates of the cable’s direction. Nevertheless, this assumption is mostly true for cables and pipelines.

It is envisaged to test this approach in a real cable-tracking mission off the Cornish coast in the southwest of England. Furthermore, the method will be extended for the localisation of small ferromagnetic objects, for which MFs will be generated from eddy currents induced by alternating MFs of the searching coils themselves. The resulting induced MF distribution of smaller objects will be shaped in the form of a single peak above the object, for which different path planning routines will need to be developed, as well as the necessary intelligent discrimination methods to identify the type of object localised.

References


Mottani A, Sharma SK, Sutton R and Culverhouse P. (2013). Interval Kalman filtering in navigation system design for...


