

**COOPERATIVE LOCALIZATION AND
BATHYMETRY-AIDED NAVIGATION OF
AUTONOMOUS MARINE SYSTEMS**

GAO RUI

(B.Eng.(Hons), M.Eng, NUS)

**A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
DEPARTMENT OF ELECTRICAL AND COMPUTER
ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE
2019**

Supervisor:

Associate Professor Mandar Chitre

Examiners:

Associate Professor Abdullah Al Mamun

Associate Professor Chew Chee Meng

Associate Professor Nikola Mišković, University of Zagreb

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Gao Rui

July 15, 2019

Acknowledgements

This dissertation would not be possible without the assistance of many people. I am very fortunate to be associated with them and would like to take this opportunity to express my gratefulness and appreciation to them in this acknowledgement.

First and foremost, I would like to thank my supervisor Assoc. Prof. Mandar Chitre who introduced me to underwater navigation. He not only provided the patient guidance and valuable suggestions but more importantly supported and encouraged me throughout the course. He has my utmost respect and gratitude for understanding and believing in me, especially when I was overwhelmed with my family commitment.

This dissertation was built on the project - STARFISH AUV¹ at ARL², NUS. I would like to extend my gratitude to my colleagues and friends at ARL. I am grateful for the opportunity for working with and being surrounded by great people. I am honored to be part of an outstanding team. Thanks go to Ms. Ong Lee Lin for her help and guidance in my research writing.

Last but not least, I want to thank my husband, Chew Jee Loong, and my loving parents. This journey would not have been possible without their understanding and support. My deepest gratitude goes to my dear mother for her unconditional love, encourage, and all the sacrifices she has made for me. I shall also mention my two kids, Amelie and Andrew, who always make my day full of joy and happiness.

¹Small Team of Autonomous Robotic “Fish” (STARFISH)

²Acoustic Research Laboratory (ARL), Tropical Marine Science Institute (TMSI), National University of Singapore (NUS) - <http://www.arl.nus.edu.sg>

Contents

Declaration	i
Acknowledgements	ii
Summary	vii
List of Tables	ix
List of Figures	x
List of Symbols	xvii
Acronyms	xix
1 Introduction	1
1.1 Motivation	1
1.2 Thesis Contribution and List of Publications	7
1.3 Related Publications	8
1.4 Thesis Layout	9
2 Background	10
2.1 AUVs and Localization	10
2.2 Beacon-Based Localization	14
2.3 Cooperative Localization	15
2.4 Dealing with Unknown Correlation	19
2.5 Bathymetry-Aided Navigation	22
3 Distributed Localization in a Cooperative Team	25
3.1 Problem Statement	25
3.2 Decentralized Cooperation with Limited Communication	26
3.2.1 Self-Propelled Particles (SPP) and Swarm AUVs	26
3.2.2 Small Team of AUVs	31

3.3	Information Loss in Distributed Processing	46
3.4	Distributed Extended Information Filter	52
3.4.1	Illustrative Examples	53
3.4.2	Formulation and Design	56
3.4.3	Simulation Studies Using Field Experiment Data	64
3.5	Summary	70
4	When Can One Ignore the Correlation?	72
4.1	Problem Statement	72
4.2	Multi-Sensor Tracking Problem	73
4.2.1	Optimal Distributed Filtering	77
4.2.2	When Can One Ignore the Correlation?	83
4.2.3	Summary	88
4.3	Multi-Vehicle Localization	88
4.3.1	Information Flow when Ignoring Correlation	90
4.3.2	Multi-Vehicle Localization with Bathymetric Aids	105
4.4	Summary	112
5	Localization with Bathymetric Aids	115
5.1	Problem Statement	115
5.2	Probability Map Based Localization	116
5.2.1	Grid-Based Markov Localization	117
5.2.2	Particle Filtering and Multiple Hypotheses	121
5.3	Information Entropy Map	124
5.3.1	Grid-based Discrete Entropy	125
5.3.2	Particle Filter Based Entropy	128
5.3.3	Empirical Convergence and Contributing Factors	129
5.3.4	Localization Performance	132
5.3.5	Information Entropy Map Based on Different Priors	137
5.4	Summary	140
6	Navigation with Bathymetric Aids	142
6.1	Problem Formulation	142
6.2	Approximate Dynamic Programming	143
6.3	Iterative Path Planning Algorithm	145
6.3.1	Policy Generation	146
6.3.2	Policy Evaluation	148
6.3.3	The Policy Iteration Algorithm	149
6.4	Gaussian Process Regression for Path Planning	150
6.5	Simulation and Performance	153
6.5.1	Underwater Vehicle Navigation	153
6.5.2	Mission 1	153

6.5.3 Mission 2	156
6.6 Summary	159
7 Conclusions and Future Work	161
Bibliography	165

Summary

In monitoring and surveillance, localization and navigation is especially important for underwater autonomous vehicles (AUVs) since GPS and radio signals are not accessible in most of the missions. A cooperative team formed by multiple autonomous vehicles has gained increasing interests as it offers more efficiency and reliability than a single vehicle. Much work has been done in the area of underwater localization and navigation with the aids from beacons with known positions. This dissertation focuses on a team of low-cost AUVs where no one has precise position. With the ability to communicate and make relative measurements from each other, vehicles share information across the team and cooperate to complete complex missions. The information shared among vehicles helps improve the overall performance. However, unlike terrestrial communication links, underwater acoustic channel has inherent constraints such as high latency, limited bandwidth and low reliability. In such a case, a distributed processing architecture with minimum inter-vehicle communications is preferred. At the same time, the severe underwater communication issues also impose challenges on the distributed processing. We show the benefit introduced through cooperation and the issue imposed by communication constraints on the distributed processing. We propose a feasible design of the distributed localization for a team of cooperative AUVs. The algorithm is robust to packet loss, outperforms single-vehicle localization and requires minimum communications.

In the next step, we look at a simplified *naïve* assumption in underwater multi-sensor tracking and multi-vehicle localization. This assumption *naïvely* assumes information from different sources are independent of each

other. We justify the assumption and answer when the assumption is good to use. With the justification, we move on to single-vehicle bathymetry-aided navigation as it can be safely extended to cooperative multi-vehicle navigation with bathymetric aids.

Bathymetry map indicating the water depth offers an attractive tool to reduce the localization error of the submerged vehicles. With bathymetric measurements incorporated, we study the *a posteriori* description of the localization uncertainty using probabilistic methods. An information theoretical approach is used to describe the localization uncertainty. With this information theory measure, we build information entropy map and demonstrate how the bathymetry helps with different localization priors. Subsequently, a bathymetry-aided navigation is proposed to ensure a good localization at the destination. We formulate the path planning and solve it as an optimization problem. The algorithm is tested using real-world bathymetric data. Simulations show that near-optimal paths with good localization accuracy at the destination are generated within a few iterations.

List of Tables

3.1	Pairing Sequence for Ranging Update	42
-----	---	----

List of Figures

3.1	Smaller update interval and less packet loss lead to better group heading alignment.	28
3.2	AUV aggregation at update interval $\delta T = 30$ seconds: smaller packet loss results in faster convergence of the group coverage.	30
3.3	Average Error in Distance of 3 AUVs with various loss rate p_L : When packet loss is less, DKF improves localization through cooperation. When packet loss is higher, DKF improves localization at first but later deteriorates fast.	39
3.4	Estimation error of AUV 4 gets corrected once it reconnects with the other AUVs after 30 seconds.	43
3.5	Logs of past $N_p = 5$ rangings are able correct the DEKF and give similar result to CEKF.	44
3.6	Central processing architecture vs. distributed processing architecture. CKF fuses the raw measurements directly while DKF fuses the local processed data.	47

3.7	Estimation error squared versus initial correlation coefficient. DKF has larger error covariance than CKF except when initial correlation coefficient is 0.	49
3.8	Traditional distributed processing (DKF) performs poorly as compared to centralized processing (CKF), but our proposed distributed method (DEIF) is able to perform well.	54
3.9	Estimation error of AUV 1 using various filters in a 3-AUV cooperative localization example. DEIF is close to CKF.	55
3.10	Illustration of incorporating delta information in simple addition.	63
3.11	Cooperative localization results with field data.	65
3.12	Simulation results for Vehicle 3 at packet loss rate (a) $p_l = 0$ (b) $p_l = 0.3$ (c) $p_l = 0.6$ and (d) $p_l = 0.9$. The vertical arrows show the time when Vehicle 3 receives broadcast. DEIF has smaller estimation error than SKF and SCI filters, and better consistency than NKF.	67
3.13	Cooperative localization results with field data. The arrows indicate the time when the broadcasts are sent and successfully received by other vehicles. NKF estimation sometimes is worse than SKF. DEIF improves estimation accuracy of all vehicles.	70

-
- 4.1 Multi-sensor tracking: a recursive two-step flow chart. The target propagates from previous position \mathbf{x}_k to current position \mathbf{x}_{k+1} with some propagation noise ω_k . At each step, each node makes an observation ($\mathbf{z}^{(1)}$ or $\mathbf{z}^{(2)}$) about the target position. 74
- 4.2 Multi-sensor tracking: distributed filtering using weighted-sum fusion. 78
- 4.3 One-step error covariances of DF, SF, CF and SVF, against the weight λ used by DF. There is a gap between CF and optimal DF (SVF). 80
- 4.4 Ratio of error covariances - optimal DF (SVF) to CF is no larger than one. 83
- 4.5 One-step performance: the *dangerous* region of implementing NF (assuming $\mathbf{R}^{(1)} < \mathbf{R}^{(2)}$). 85
- 4.6 *Naïve* filter in stable state: (a) Actual error covariance is smaller than SF. (b) Actual error covariance eventually gets worse than SF. In both cases, NF is overconfident about its estimation. The estimated error covariance by NF is even lower than the one by CF. 86
- 4.7 Asymptotic performance: the *dangerous* region of implementing NF (assuming $\mathbf{R}^{(1)} \leq \mathbf{R}^{(2)}$). 87

-
- 4.8 Bayesian network for cooperative localization of two vehicles: shaded nodes are observations and white nodes are unobserved position state variables. 89
- 4.9 Full graph vs. Unwrapped graph with information double counting for $k = 1, 2$: the prime symbol indicates the replication of nodes. 96
- 4.10 Case 1: $\mathbf{P}_k = 5, \mathbf{Q} = 10, \mathbf{R}^{(1)} = 1, \mathbf{R}^{(2)} = 2$. When ranging error approaches 0, the two estimates are about fully correlated with similar error covariances after cooperation. . 108
- 4.11 Case 2: $\mathbf{P}_k = 5, \mathbf{Q} = 1, \mathbf{R}^{(1)} = 10, \mathbf{R}^{(2)} = 40$. When ranging error approaches 0, the two estimates are about fully correlated with similar error covariances after cooperation. . 109
- 4.12 One-step Performance: The *dangerous* region of implementing NF in multi-vehicle localization. ($\bar{\mathbf{R}}^{(2)} > \bar{\mathbf{R}}^{(1)}$) 112
- 4.13 *Naïve* filter in stable state: One can ignore the correlation in Case (b) and Case (c) but NF becomes detrimental in Case (a). 113
- 4.14 Asymptotic Performance: The *dangerous* region of implementing NF in multi-vehicle localization. The three cases in Figure 4.13 are located in the regions. 114
- 5.1 Bathymetry map near St. John Island, Singapore. Circles: Path 1 (Speed 2 m/s). Crosses: Path 2 (Speed 1.41m/s). . . 118

5.2	Grid-based Markov localization with measurement updates. Yellow crosses: top 5 possible locations. Cyan circles: true path. Green square: estimated locations with top possibility.	119
5.3	Depth measurements along two paths.	120
5.4	Particle filter localization with measurement updates for Path 1. Black circles: true path. Cyan regions: distribution of particles.	122
5.5	Gaussian mixture model estimated by EM method.	123
5.6	Gaussian random walk process: Estimation error is reduced at every measurement update.	130
5.7	Gaussian random walk process: PF-based Entropy and grid-based discrete entropy, versus theoretical entropy. Vertical red lines: time steps when measurements are available. Vertical green dashed lines: time steps when particles are re-sampled. Grid-based entropy is more sensitive to particle numbers and estimation error values.	131
5.8	PF-based entropy versus grid-based discrete entropy: Grid-based discrete entropy varies more, with respect to particle number.	133
5.9	The entropy of the particles in a bathymetric navigation particle filter depends on the path taken from source to destination.	134

5.10	Although Path B takes a detour and therefore longer time to reach the destination, the localization error of Path B is much smaller than a that of a straight line by Path A. . . .	136
5.11	Information entropy map for different areas: Different prior distributions yield different conditional entropy values. . . .	138
5.12	Information entropy map based on different Gaussian priors: It is good to have more bathymetry variation in the direction where the larger uncertainty of the Gaussian prior lies. . . .	140
6.1	Flowchart of iterative path planning algorithm.	145
6.2	Action space for policy generation: The next waypoints from action set include all possible map grids when looking at the destination.	147
6.3	Illustration of Gaussian process regression in 1-dimensional space.	152
6.4	Mission 1: As the algorithm iterates, the planned path evolves to one with more bathymetric variation.	154
6.5	Mission 1: Performance at the destination.	155
6.6	Mission 2: In the first three iterations, planned path evolves to one side of the bathymetry basin. From the fourth iteration, the planned path evolves to the other side of the basin.	157
6.7	Mission 2: Performance at the destination.	158

6.8 Illustration of Gaussian process regression in 1-dimensional space.	159
--	-----

List of Symbols

k	time step k , used as subscript for other symbols
\mathbf{x}_k	position state at time step k
\mathbf{a}_k	action directing vehicle movement at time step k
\mathbf{F}	Jacobian matrix of evolution model
\mathbf{H}	Jacobian matrix of measurement model
\mathbf{B}_k	control matrix at time step k
\mathbf{u}_k	control input at time step k
\mathbf{z}_k	measurement or observation at time step k
$\mathbf{Z}_{1:k}$	series of measurements from time step 1 to k
r_k	range measurement at time step k
ω_k	system evolution noise
\mathbf{Q}_k	covariance of system evolution noise
ν_k	measurement noise
v_k	range measurement noise
\mathbf{R}	covariance of measurement noise
\mathbf{R}_r	covariance of ranging error
$\hat{\mathbf{x}}_k, \mathbf{y}_k$	position estimate of true position state \mathbf{x}_k
\mathbf{P}_k	error covariance of the position estimate
δT	measurement update interval

p_L	communication packet loss rate
N_p	number of past updates
Λ_k	precision matrix or information matrix
η_k	information vector
\mathbf{O}_x	State mapping matrix
\mathbf{O}_y	Observation mapping matrix
\mathbf{E}	Error matrix

Acronyms

AUV	A utonomous U nderwater V ehicle
BAN	B athymetry A ided N avigation
CKF	C entralized K alman F ilter
DF	D istributed F ilter
DoF	D egree-of- F reedom
DR	D ead R eckoning
DVL	D oppler V elocity L og
EKF	E xtended K alman F ilter
EIF	E xtended I nformation F ilter
GIB	G PS I ntelligent B uoys
GMM	G aussian M ixture M odel
INS	I ntertial N avigation S ystem
KF	K alman F ilter
LBL	L ong B ase L ine
MAP	M aximum A P osteriori
ML	M aximum L ikelihood
NEES	N ormalized E stimation E rror S quared
NKF	<i>N</i> aïve K alman F ilter
PDF	P robabilistic D ensity F unction

PF	P article F ilter
PV	P eer V ehicle
RMSE	R oot M ean S quare E rror
RV	R eceiving V ehicle
(U)SBL	(U ltra) S hort B ase L ine
SCI	S plit C ovariance I ntersection
SI	S warm I ntelligence
SPP	S elf- P ropelled P articles
SVF	S tate V ector F usion

Chapter 1

Introduction

1.1 Motivation

The past decade has seen a significant increase in interest in the use of autonomous underwater vehicles (AUVs) for maritime operations. AUVs reach shallower water than boats, and greater depths than human divers or tethered vehicles. Once deployed and submerged, AUVs are safe from bad weather and heavy traffic on the surface. Extensive research has been conducted on AUVs and several commercial AUVs have been developed and tested [21, 48, 50]. One of the challenges that all AUVs have to contend with is that of underwater localization and navigation, since GPS signals cannot be received underwater. For missions such as mapping, target identification, sensing and surveying, underwater localization and navigation

are the key components that determine the performance of an autonomous mission.

Traditionally an AUV localizes itself using the vehicle's own sensor data, such as GPS (whenever available), compass and depth sensor. In order to obtain good localization, AUVs are often equipped with high-end navigation systems such as Doppler velocity log (DVL) and inertial navigation system (INS). Packed together with other sensors for the mission (for example, sensors for monitoring and surveying), AUVs become bulky in size and high in cost. Another way to achieve the vehicle's localization accuracy is to utilize the communication with beacons with known positions. The distance from AUV to a beacon can be measured by the time-of-flight (TOF) of the acoustic signals. The relative orientation of the AUV can be computed from a set of equations consisting of distances to all available beacons [91] or the phase shift measured from the beacon array [78]. Long Baseline (LBL) uses a sea-floor baseline transponder network as navigation reference. It gives very high positioning accuracy and position stability, but incurs high effort and cost to deploy and support this substantial infrastructure. Other techniques such as Short Baseline (SBL), Ultra Short Baseline (USBL) [57] and GPS Intelligent Buoys (GIB) [2] are placed on sea surface. Localization with these setups can only be achieved in certain restricted and designated areas due to the installation of these external beacons. Some systems also assist the localization and navigation

with moving beacons such as those mounted on surface vessels [41, 64, 86] or even other AUVs [72]. The operation of additional surface vessels is not trivial; the horizontal coverage is also limited. The positions of beacon AUVs need to be calibrated precisely.

In these respects, AUVs which are able to navigate without aids from external setups or beacons and yet still maintain good localization with the use of low-grade sensors, are preferred. This has motivated multi-vehicle localization and navigation [20, 54, 72], and the use of bathymetric aids.

Multi-vehicle operation offers robustness to single-vehicle failure and reduces the overall time and cost of acquiring data over large areas. We focus on a team of low-cost AUVs where no single AUV functions as a beacon possessing accurate position. The team is able to outperform single-vehicle operation, through sharing information among the team members. However, as compared to terrestrial communication links, underwater communication links typically are encumbered by limited bandwidth, high latency and significant packet loss. It is impractical for a team of AUVs to collate all sensor data centrally, and therefore a decentralized processing architecture is preferred. However this opens up new challenges to team members working in cooperation.

The first challenge lies in the limited communication bandwidth. As

mentioned, the multi-vehicle localization problem consists of a team of vehicles cooperating and estimating their own positions. If the local sensor data are accessible to a central processing unit, a central filtering yields the optimal estimation on all vehicle positions. However, with limited bandwidth, the size of underwater transmission packets is constrained. The information shared between members is usually processed estimates rather than the raw sensor data. A distributed filtering where members only process their local sensor data and information communicated from others is commonly used. An empirical study [80] showed that the available bandwidth in underwater communication severely limits the performance of cooperative localization.

The second challenge pertains to tracking of the inter-vehicle correlation. In the presence of unstable, irregular and lossy channels, the information exchange between two vehicles may not be known to other members in the team. Therefore the inter-vehicle correlation tends to be underestimated. A common practice is to *naïvely* assume independence during information fusion between cooperative members. This assumption is not always good. Overconfidence in estimation has been reported as a result of double counting the common information [43]. This overconfidence prevents utilization of subsequent useful measurements. To prevent information double counting, author in [9] selectively incorporated information and avoided the fusion of correlated data while keeping the positions of all vehicles decorrelated. Other works assumed maximum correlation in the whole

state [15, 84, 92] or separated states [51]. These methods are conservative in handling the unknown inter-vehicle correlation, and typically yield pessimistic estimates as the independent information is not fully utilized. Based on the way that vehicles cooperate, a less conservative distributed localization could be designed to record the correlated terms and require minimum communications. Meanwhile, the *naïve* assumption which simply ignores the correlation, has been used as a common practice as it imposes minimum requirements on the processing and communication. One would be interested to know how information double counting happens and when the correlation in fusion can be ignored. The dangerous and safe regions of *naïve* filtering should be quantified.

Incorporating information sensed from the natural environment helps improve the localization without the need to establish extra physical infrastructure. Bathymetry map indicating the water depth offers an attractive tool to reduce the localization error of the submerged vehicles. Reference [80] presents a cooperative bathymetry-based localization approach for a team of low-cost AUVs. It presents empirical analysis on the factors that affect the filter performance. However, vehicle estimates are assumed to be independent of each other as the author claims that the effect of correlation is alleviated with bathymetry measurements. The amount of common information fed back through the cooperation in the team is not addressed. In our work, we justify this assumption through a careful analysis of the

effect of bathymetry aids in navigation. The advantage of bathymetric aids is path dependent. It makes sense if vehicles are able to plan a path such that a good localization can be ensured. It is generally believed that localization performance depends on the rugosity of the sea bottom. Authors in [47] show that the bathymetry variation is closely related to the localization performance. Bathymetry variation is commonly used to guide vehicles towards maximizing the bathymetric aids on localization. In [37], heuristics were used to visit salient points (locations with more bathymetric variation) for better localization. In [62], terrain dispersion, roughness and terrain entropy were evaluated. However, the waypoints along the path were selected manually. In [35], trajectories were generated by guiding vehicles to avoid areas with smooth sea floor. In order to plan a path to maximize the effect of bathymetric aids, we need to understand what kind of bathymetry make good localization first. Is the bathymetry variation the sufficient condition? How does one define a good localization? With these answers, we propose a path planning algorithm to optimize the localization performance with bathymetric aids.

1.2 Thesis Contribution and List of Publications

This thesis presents our work on cooperative localization and bathymetry-aided navigation of autonomous marine systems, with AUVs being the particular focus. The key contributions of this thesis are listed below:

1. Distributed localization is explored using a cooperating team of small-sized, low-cost, sensor-limited AUVs. In the presence of underwater communication challenges, we found that localization easily deteriorate when communication loss is high.
2. A new cooperative multi-vehicle localization algorithm using distributed extended information filter (DEIF) is proposed. The proposed method is effective in recording the correlated information in light of constrained underwater communication.
3. We answer the question as to when the correlation can be safely ignored. The safe and dangerous regions for implementation of *naïve* filtering in decentralized architecture are derived for multi-sensor tracking problem and multi-vehicle localization problem. The link between these two problems is explained with formulations.
4. We formalize a concept of an information entropy map to quantify the effectiveness of bathymetry measurements on localization.

5. A path planning algorithm is proposed for navigation with bathymetric aids. The algorithm generates near-optimal paths based on bathymetric maps, with good localization accuracy generated at the destination.

1.3 Related Publications

- [1] R. Gao and M. Chitre, “Bathymetry-aided Navigation of Autonomous Underwater Vehicles (AUVs)”, Manuscript in preparation for IEEE Journal of Oceanic Engineering
- [2] R. Gao and M. Chitre, “Path Planning for Bathymetry-aided Underwater Navigation,” in *Autonomous Underwater Vehicles (AUV 2018)*, (Porto, Portugal), November 2018.
- [3] R. Gao and M. Chitre, “On distributed processing for underwater cooperative localization,” in *14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI 2017)*, (Jeju, South Korea), July 2017. (Invited).
- [4] R. Gao and M. Chitre, “Cooperative Multi-AUV localization using distributed extended information filter,” in *Autonomous Underwater Vehicles (AUV 2016)*, (Tokyo, Japan), pp. 206–212, November 2016.

1.4 Thesis Layout

This thesis is organized as follows.

Chapter 2 provides a brief discussion of related works in cooperative localization and bathymetry-aided navigation. Chapter 3 presents the decentralized processing when vehicles cooperate with communication constraints. A novel distributed localization method is proposed. Chapter 4 answers the question of when the correlation can be safely ignored when fusing estimates from different sources. The safe and dangerous regions of implementing *naïve* filtering (ignoring correlation) are derived in two applications. The bathymetry-aided localization is justified to be safe if correlation is ignored in cooperation.

Chapter 5 illustrates localization with bathymetric aids using an information theoretic approach. In light of the entropy measure and the relation to localization introduced in Chapter 5, a path planning algorithm is proposed and tested through simulation in Chapter 6.

Chapter 7 summarizes the key findings in the research and makes suggestions for future works.

Chapter 2

Background

2.1 AUVs and Localization

The words *Positioning* and *localization* are often used interchangeably in literature. Positioning, although similar to localization, also has the connection of placing the vehicle in a particular place. In this thesis, we will use the word *localization* to mean the process of locating the vehicle, with or without the reference of a map.

The sensors for underwater localization are categorized into two groups. The first group of sensors do not require external infrastructure for localization. For example, DVL measures vehicle speed relative to the seabed. A microelectromechanical system (MEMS), usually called as compass, measures the vehicle's 3-axis orientation. An inertial measurement unit (IMU)

provides the vehicle's acceleration and angular rate with respect to the vehicle's body-frame. Other examples include depth sensor, altimeter, sidescan sonar, and forward looking sonar (FLS). Localization by these sensors usually observe the natural environment as external references and can be performed in any environment. However these sensors often do not give full information about vehicle's location. The most common localization method of localization with these sensors is dead reckoning (DR). It calculates the vehicle's position by integration of velocity over time. Accurate DR systems tend to be expensive as high-grade sensors are needed. Even though, any small bias or offset leads to an unbounded increase in error over time while AUV remains submerged. The second group are sensors which require additional infrastructure setup. Acoustic positioning systems measure vehicle's range and/or angle to the beacons and therefore provide direct estimates of the position relative to beacons. The beacons can be fixed to sea bottom, or the surface vehicles, or even other AUVs. These sensors give good localization reference but often incurs high cost in deployment and operation.

In a typical localization problem, the state to be estimated usually consists of the vehicle's position, depth, orientation (heading, pitch and roll) and velocity. During a mission, the autonomous vehicle estimates its own state and at the same time uses the estimated state for further actions. Usually the depth of an AUV is specified in a mission and measured by the

depth sensor directly. Therefore in this thesis we only focus on the vehicle's position in the 2-dimensional space, that is, the northing-easting position.

Let \mathbf{x}_k be the state vector to be estimated at time step k where $\mathbf{x}_k \in \mathbb{R}^n$, and let the estimate of \mathbf{x}_k be $\hat{\mathbf{x}}_k$ (In some illustration in this thesis, we also use \mathbf{y}_k to denote the estimate of state \mathbf{x}_k). The general system evolution is modeled as

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{a}_k) + \omega_k \quad (2.1)$$

where \mathbf{a}_k is the action that directs vehicle movement for the next step. The behavior of the system is observed through measurement $\mathbf{z}_k \in \mathbb{R}^m$. The measurement model is

$$\mathbf{z}_{k+1} = h(\mathbf{x}_{k+1}) + \nu_{k+1}. \quad (2.2)$$

The process noise ω_k and measurement noise ν_k are Gaussian white-noise sequences and are mutually independent. Their error covariances are \mathbf{Q}_k and \mathbf{R}_{k+1} respectively.

Typically the vehicle carries a belief as to where it might be, and maintains the localization as a probability distribution over the space of all such hypotheses. Knowledge of the probability density function (PDF) of the state conditioned on all available measurement data $\mathbf{Z}_{1:k} \triangleq \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_k\}$

provides the most complete possible description of the state. Given a *posteriori* density function $p(\mathbf{x}_k|\mathbf{Z}_{1:k})$, estimate $\hat{\mathbf{x}}_k$ of the state is obtained from some performance criteria, for example, maximum *a posteriori* (MAP). Bayesian estimation [6] recursively determines the a posteriori density $p(\mathbf{x}_k|\mathbf{Z}_{1:k})$ as

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{Z}_{1:k}) &= \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{Z}_{1:k})} \\ &= \alpha p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}_{1:k-1}) \end{aligned} \quad (2.3)$$

where

$$\begin{aligned} \frac{1}{\alpha} &\triangleq p(\mathbf{z}_k|\mathbf{Z}_{1:k}) \\ &= \int p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}_{1:k-1})d\mathbf{x}_k. \end{aligned} \quad (2.4)$$

The simplification is based on Markov assumption [4] which states that if one knows the vehicle location \mathbf{x}_k , future measurements are independent of the past ones, that is

$$p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{Z}_{1:k-1}) = p(\mathbf{z}_k|\mathbf{x}_k). \quad (2.5)$$

The density $p(\mathbf{z}_k|\mathbf{x}_k)$ is the *a priori* distribution describing the sensor model. The state transition model is described by the prediction as

$$p(\mathbf{x}_k|\mathbf{Z}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{Z}_{1:k-1})d\mathbf{x}_{k-1}. \quad (2.6)$$

When both evolution and measurement models are linear with additive Gaussian noises, and the *a priori* distribution is Gaussian, Kalman filter (KF) [88] can be used to derive the closed-form solution by simply using the estimated mean $\hat{\mathbf{x}}$ and estimated covariance \mathbf{P} for this system. The parametric description of the distribution is efficient in integrating the motion and updating the estimates, and also easy to evaluate the localization performance. However it is only suitable for linear systems with Gaussian noises. Extended Kalman filter (EKF) [45] is the nonlinear version of the KF which linearizes the nonlinear evolution and/or nonlinear measurements. Only unimodal distributions can be modeled by KF and its extensions. Nonparametric filters use numerical approaches to describe the PDF and are particularly suitable for nonlinear and non-Gaussian system estimation since their probability function evolves to better fit the data. The most well-known nonparametric method is the Particle filter (PF) [7].

2.2 Beacon-Based Localization

In beacon-based localization, beacons are placed in the area where AUVs navigate. For example, long baseline acoustic positioning system (LBL) places the transponders on the seafloor, while short baseline system (SBL) and ultra short baseline system (USBL) have the transponders mounted on a ship that follows the vehicle. These methods differ in the distances

between the transponders and the distance from each transponder to the vehicle [90]. The GIB system uses buoys on the surface [2]. When fixed beacons are used [63], AUVs are limited in their exploration area. The deployment and setup of the infrastructure are tedious and expensive. Some systems assist the localization with moving beacons. The moving beacons can be mounted on vessels [28, 64, 86] or even other AUVs [8, 30, 72]. Surface vessels often encounter danger of collision with other traffic on the surface. Beacon AUVs are usually assumed to have precise positioning. Generally the beacon vehicle operates at the surface and has access to GPS, or is equipped with high-accuracy sensors which enables it to estimate its own position with minimum errors. A setup, which consists of beacon AUVs with precise positioning, can form a cooperative team of heterogeneous AUVs.

2.3 Cooperative Localization

The idea of cooperative localization with beacon AUV is not new. Authors in [72, 81] presented a single-beacon vehicle providing range-only measurement to support the localization of other AUVs. The supported AUVs are survey AUVs equipped with sensors for mission purposes. For example, LEDIF sensor [59] was installed on STARFISH AUVs [49] for chemical sensing. With these sensing units, survey AUVs are often equipped with

poor navigation sensors. The distance between the survey AUV and the beacon vehicle is measured to impose a limit on the position drift of the survey AUV. This approach has been explored by several works which use observability analysis [5, 76], and position determination algorithms [3, 36]. However, these works pay little attention to the path planning of the beacon vehicle. For example, [3] assumed a circular path for the beacon vehicle, [33] used a zigzag path during experiments and [85] adopted a diamond shaped path.

It is acknowledged that the relative motion between the beacon vehicle and the survey AUVs is key to having single beacon range-only navigation perform well. The path of the beacon vehicle should be planned in such a way that it improves the position estimate of the survey AUVs. In [25], the path planning of the beacon vehicle aimed at minimizing the accumulated localization errors in the supported survey AUVs. In [10], the optimal beacon point targeted at minimizing the position uncertainty of the survey AUV, but it was determined by brute-force searching approach. Later [81] proposed the cooperative path planning algorithm using dynamic programming and Markov decision process formulations.

In the waters of Singapore, heavy maritime traffic makes it dangerous and inconvenient for AUVs to surface for GPS fix. Consequently, even beacon AUVs with high-grade sensors suffer from position drift. We look at this problem and propose a solution whereby no single AUV functions as

a beacon possessing accurate position information in the team. In terms of localization capability, the team of AUVs is a homogeneous team, whereas a heterogeneous team includes a beacon AUV with precise position. The relative position or range between these AUVs can be considered as a relative geometry constraint in localization. This cooperative localization is easier to be understood if we treat the group of AUVs as one identity. The individual AUVs are the multiple “limbs” while the distances between AUVs are the multiple virtual “joints”. Each limb moves on its own path, and from time to time, when the distance between two AUVs is measured, the position estimates of all the AUVs are adjusted accordingly. Cooperative multi-AUV localization has the potential to outperform single-AUV localization, by taking advantage of data sharing among the team members. However, it should be noted that many factors affect the communication and subsequently the cooperation performance. Water temperature, salinity, underwater noise, Doppler phase shift, reflection and scattering of seabed and sea surface are relevant contributing factors to be considered. Underwater communications have limited bandwidth and are less reliable compared with terrestrial communication links. Full communication with every member of the team at any one time of instant is often not practical.

In such a case, decentralization in processing and navigation becomes necessary. One example of decentralized processing comes from the self-organized behavior, namely *Swarm Intelligence (SI)*. SI system consists of

a population of simple agents, which interact locally with one another following simple rules. There is no centralized unit or individual who knows the full details or dictates how each member should behave. Interactions between agents lead to an ordered or “intelligent” global behavior. The phenomenon is often referred to as ‘emergence’. The inspiration often comes from nature, especially biological system behavior like ant colonies, bird flocking, animal herding and fish schooling. The application of SI to robotics is called ‘swarm robotics’, a method of coordinating large numbers of simple robots, which interact with one another to give rise to a desired collective behavior [58, 60, 61]. In ocean studies, the use of swarm AUVs for data gathering purposes has emerged as an attractive and alternative solution to the tedious and manual process of deploying sensor probes, for example beacons installed on surface vessels. Swarm AUVs are able to gather more data than the traditional approach, operating at lower overall cost, and can be deployed to function in harsh environment. It is also robust to individual failure, compared with the single-AUV system. Swarm AUVs coordinate their behavior in a distributed way to achieve a particular goal, such as resource sharing, synchronized motion [74], localization [53], a specific swarm pattern or coverage [77], etc.

The concept of SI can be visualized in the tutorial about self-propelled particles (SPP) in [74]. In this tutorial, each particle has its own randomness in movement and follows the average heading when it meets other

particles within its vicinity. It introduces order parameters and other critical values to visualize how the order or phase change with respect to the variation of randomness, number of particles, vicinity range etc. In SI, the communication and coordination among members is usually minimal. The observation on other members also comes with more randomness.

2.4 Dealing with Unknown Correlation

In this thesis, we focus on a small team of AUVs. These AUVs locally estimate their own positions, and gather observations to update the position estimates. The distributed processing architecture has many advantages over centralized architecture. It is reliable in the sense that the loss of any individual AUV or links does not necessarily prevent the rest of the team from completing the mission. It is flexible in the sense that AUVs can be added or deleted from the team by making only local changes.

The most challenge arising from distributed processing in a cooperative team is the effect of redundant information [39]. To be specific, the information from different vehicles cannot be combined straightforward unless they are independent or have a known degree of correlation. Many years ago authors in [13] recognized that local estimates have correlated errors. If between local estimates the correlation is *naïvely* assumed to be zero, estimation overconfidence comes with the fused estimate, and may lead

to filter divergence [42]. Therefore dealing with unknown correlation in the cooperative localization becomes important. With smaller number of AUVs in a group, common information flowing around the team is more influential. This is because the common information easily cycles with less independent information input from other AUVs.

A simple parametric localization filter like Kalman filter [46] and its extensions [38, 45, 46] gives estimated position $\hat{\mathbf{x}}$ and the corresponding estimated error covariance \mathbf{P} . At each time step, a vehicle predicts its own position and updates the estimate if local measurement is available. They keep their local estimate $\hat{\mathbf{x}}^{(\cdot)}, \mathbf{P}^{(\cdot)}$ where (\cdot) denotes the vehicle's identity. There is a time when two vehicles (Vehicle i and Vehicle j) in the group communicate, exchange their information (for example, the estimated position and estimated error covariance, $\hat{\mathbf{x}}^{(i)}, \mathbf{P}^{(i)}$ and $\hat{\mathbf{x}}^{(j)}, \mathbf{P}^{(j)}$), and update their respective position estimates. The other members in the group may not know this cooperation due to the loss of transmission packets. Therefore, to those members, the correlation between information provided by vehicles i and j is unknown. This is essentially a data fusion problem dealing with unknown correlation in distributed network. The estimation can be visualized as an estimation network where the nodes are the vehicles and the edges connecting the nodes denote the information flow within the network.

For one-step cooperation, we drop the time step k for simplicity. Assuming both estimates are consistent, that is that $\mathbb{E}[(\hat{\mathbf{x}}^{(i)} - \mathbf{x}^{(i)})(\hat{\mathbf{x}}^{(i)} - \mathbf{x}^{(i)})^\top] \preceq \mathbf{P}^{(i)}$ and $\mathbb{E}[(\hat{\mathbf{x}}^{(j)} - \mathbf{x}^{(j)})(\hat{\mathbf{x}}^{(j)} - \mathbf{x}^{(j)})^\top] \preceq \mathbf{P}^{(j)}$, the fused estimate $(\hat{\mathbf{x}}^{(f)}, \mathbf{P}^{(f)})$ at Vehicle i should satisfy the following *fusion principles* [67, 84]:

1. Performance improvement: $\mathbf{P}^{(f)} \preceq \mathbf{P}^{(i)}$,
2. Estimation consistency: $\mathbb{E}[(\hat{\mathbf{x}}^{(f)} - \mathbf{x}^{(i)})(\hat{\mathbf{x}}^{(f)} - \mathbf{x}^{(i)})^\top] \preceq \mathbf{P}^{(f)}$.

An optimal fusion is derived in [23] if correlation is known. To fuse estimates with unknown correlation, [43] proposed a Covariance Intersection (CI) method, which essentially provides an upper bound on the error covariance of the fused estimate. It leads to various approaches for determining the weight w [24, 40]. Although the estimation consistency is maintained, the estimated error covariance is no smaller than the individual error covariance at any direction. The expression of the fused estimate is derived by assuming independent errors between $\hat{\mathbf{x}}^{(i)} - \mathbf{x}^{(i)}$ and $\hat{\mathbf{x}}^{(j)} - \mathbf{x}^{(j)}$, which is contradictory to the problem formulation. Reference [75] proposed Ellipsoid Intersection (EI), which essentially picks the estimate with smaller error covariance. It fulfills the first fusion principle but the consistency is not proved. Along the same line as CI, the author in [52] and his related work proposed Split Covariance Intersection (SCI), which uses split form to represent the dependent and independent parts in the estimate. However, in the state update with relative position estimates and states of

other vehicles, only the relative measurement is separated as independent information.

Instead of overestimating the intersection region, authors in [15] proposed a largest ellipsoid algorithm which leads to a tighter estimate but the fused estimate is slightly underestimated. In place of CI, [67] proposed Bounded Covariance Inflation (BCInf), a mechanism for creating conservative covariance matrices for which the bounds on the cross correlations are known. The key is the book keeping messages (the coupling scalar) to interpret the correlated part and uncorrelated part. The decentralization was demonstrated by using only two-vehicle SLAM.

2.5 Bathymetry-Aided Navigation

Bathymetry is the submerged equivalent of an above-water topographic map. It is obtained by prior survey and recorded in a geographical map. Bathymetry-based localization and navigation, is also known as terrain relative navigation (TRN) [55], terrain based navigation (TBN) [22, 56], terrain-aided navigation (TAN) [22], and bathymetry-aided navigation (BAN) [47]. The key idea is to match the local bathymetry as seen by an underwater vehicle against the reference map, and estimate the location of the vehicle on that map. Bathymetric SLAM is a well-recognized concept of navigation and map building [14, 18] using a multibeam sonar in high-end

AUVs. Multibeam sonar measures the bathymetry in a small patch of area. The measured information is rich as it consists of multiple measurements in a geographical formation. In terms of localization, rich bathymetry information from multibeam sonar gives high localization accuracy. However, multibeam sonar is costly and take up space in AUVs. As we focus on small-sized and low-cost AUVs, bathymetry measurements can be simply managed with a single echo-sounder [73] or altimeter. Observations from the depth sensor and altimeter lead to the observed bathymetry where the vehicle is located. Although the bathymetry is a single-point measurement, works in [47, 80] have showed the feasibility. It is fused with the odometric estimation to get the best possible update about the vehicle position. Therefore bathymetry map indicating the water depth offers an attractive tool to reduce the localization error of the submerged AUVs, without additional cost on hardware or external infrastructure.

In [47], the authors showed a strong correlation between localization accuracy and variation of bottom topography. The conclusion was that a reasonable localization could be made with a single beam altimeter, as long as sufficient bathymetric variation was available along the AUV path. In [35], bathymetry variation was characterized by the depth gradient and a path was generated to avoid regions where the terrain has small variations. In [37], salient points (locations with more bathymetry variations) are visited when the localization uncertainty surpasses a user-provided threshold.

Various terrain statistic information was listed in [62], including standard deviation, roughness, correlation coefficient and entropy. However the authors did not demonstrate these terrain information metrics and the set-points (planned points along the path) were manually selected. The relation between the bathymetry information and localization performance has not been systematically studied. This relation first requires a quantification of localization uncertainty, which is important in relating localization performance to the bathymetry information. Generally, the undulating topology of the underwater terrain yields non-Gaussian or multi-modal distributions in localization. Traditional parametric filters such as Kalman filter are not suitable in describing such distributions or quantifying the localization performance. Secondly, the bathymetry information need to be defined properly. Information entropy map turns out to be a suitable idea matching the localization uncertainty and bathymetry information. The idea of entropy-based map has been used as an efficient method of portraying terrain data [32]. Author in [89] also provided an information theory framework for the analysis of spatial uncertainties. Particle filter based entropy was derived in [17] to characterize the uncertainty of a running particle filter. Up to what we know, no one has used the information theoretic measure to evaluate the navigation with bathymetric aids.

Chapter 3

Distributed Localization in a Cooperative Team

3.1 Problem Statement

In presence of limited bandwidth and lossy channels, the information communicated between vehicles may not be raw information; cooperation in a team may not be acknowledged by all the members as well. We show that the cooperation under constrained communication still helps improve individual performance. However the cooperation may yield worse localization than single-vehicle operation without cooperation. This is mainly due to the lossy channel which makes vehicles treat correlated information as independent. We illustrate these problems with simple examples and

propose a decentralized localization algorithm. This algorithm is able to handle unknown correlation, requires small transmission packets and provides consistent position estimates when fusing correlated data.

The work in this chapter was published in [69].

3.2 Decentralized Cooperation with Limited Communication

Many natural behaviors have shown that simple interactions among a population of simple agents improve the overall performance with decentralized processing. Examples are birds flocking, ant colonies, animal herding, bacterial growth, fish schooling, etc. The collective behavior of decentralized, self-organized systems is called *Swarm Intelligence (SI)*. We show the improvement through simulations of the two common SI behaviors. Next we simulate a small team of AUVs, cooperating at various loss rate of the communication packets. We highlight the problems and challenges of the distributed localization with underwater communication constraints.

3.2.1 Self-Propelled Particles (SPP) and Swarm AUVs

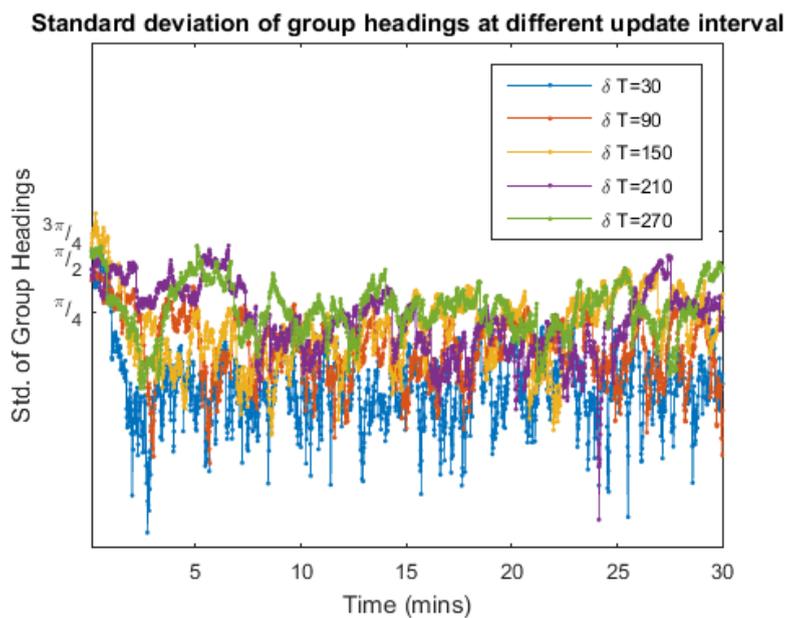
Self-propelled particles (SPP) is a swarm modelled by a collection of particles that move with a constant speed but respond to a random perturbation

[29]. The simple interactions with each other lead to the emergence of collective behavior. Two examples of SPP models - group heading alignment and aggregation, are applied to a simulation with 7 AUVs. We show that at various loss rate of the communication, the AUV swarm still outperforms the individual.

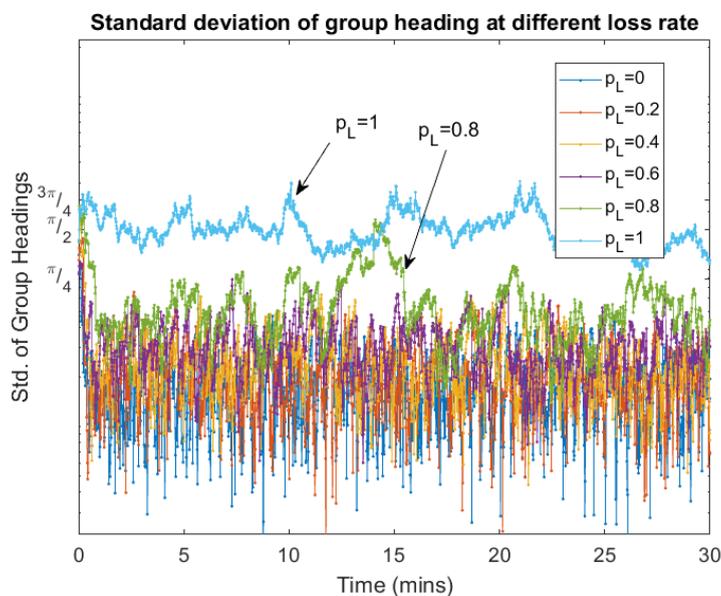
3.2.1.1 Group Heading Alignment

The group of AUVs are simulated with random noises added to the vehicle heading direction and heading speed. Each vehicle shares its heading quasi-periodically, that is, periodically with some noises. Other vehicles have a probability to hear the broadcast and turn to the angle bisector with the broadcaster's heading. Vehicles are deployed randomly in an area of 1000×1000 meters squared with random initial headings.

The standard deviation of all AUV headings indicates the alignment of the group. We plot the alignment with different the update interval δT at a loss rate 50%. In Figure 3.1(a), the update interval δT varies 1 minute apart from 30 seconds to 4 and half minutes, with a varying noise $\mathcal{N}(0, 2^2)$ in seconds. It can be seen that the standard deviation of vehicle headings is smallest for $\delta T = 30$ seconds. With shorter update interval, the heading information spreads faster and the group headings get aligned within 3 minutes. In the other hand, longer update interval gives slower



(a) At 50% packet loss: Smaller update interval yields faster convergence and smaller variation in group heading.



(b) With update interval $\delta T = 30$ seconds: Smaller packet loss yield better group heading alignment.

FIGURE 3.1: Smaller update interval and less packet loss lead to better group heading alignment.

heading alignment. Update intervals of 210 seconds and 270 seconds show little convergence in vehicle headings. The standard deviation is near $\frac{\pi}{2}$ radius, which means vehicles are heading in almost all directions.

Figure 3.1(b) shows the group heading alignment at update interval $\delta T = 30$ seconds, with different loss rate. With higher loss rate, the group headings align to a lesser degree.

The group heading alignment can be used to lead the group heading with one vehicle sticking to its planned heading. Similar application is the group search and tracking, where team members follow the heading combined between target-driven and group-coherent rules.

3.2.1.2 AUV Aggregation

Another commonly seen swarm intelligence is the aggregation, where entities, particularly animals, of similar size which aggregate together, perhaps milling about the same spot or moving or migrating in some direction. We model this aggregation using AUVs. In the aggregation, at each time, one vehicle broadcasts its own position estimates. Other vehicles that successfully pick up the broadcast will head towards the broadcaster, or the center of the broadcasting vehicles if they received two broadcasts within 1 second. An aggregation circle is drawn to cover all the vehicle locations. We use the maximum pairwise distance to evaluate the group coverage. Vehicles

are initialized to head straight away from each other. They only update their headings if they receive the broadcasted position from other team members. We show that AUVs aggregation performance highly depends on the communication rate.

Figure 3.2 shows that if the communication breaks down totally ($p_L = 1$), vehicles move further and further from each other and group coverage increases linearly. When the communication has less packet loss, the group coverage converges faster, to a minimum coverage. Group coverage converges to around 100 meters with loss rate 0 and 0.2.

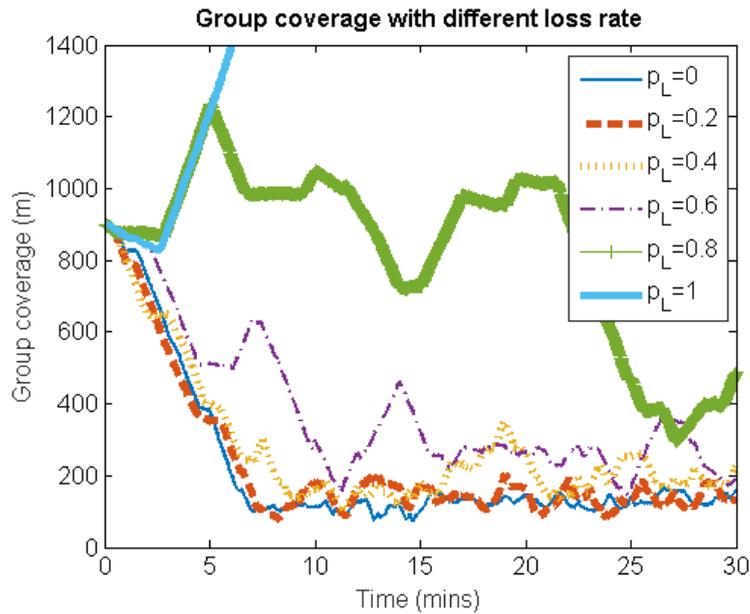


FIGURE 3.2: AUV aggregation at update interval $\delta T = 30$ seconds: smaller packet loss results in faster convergence of the group coverage.

3.2.2 Small Team of AUVs

We look at 3 AUVs cooperating through acoustic ranging. 3 AUVs is considered the smallest number. The reason is that 2-AUV group has common information directly returned; it is not complete to understand the information flow and the effect of unknown correlation in a third member with a 2-AUV group.

The distance between vehicles is calculated based on the time-of-flight of the acoustic signals. Compared with swarm AUVs, the cooperative network is smaller and the cooperation is specified to localization with aids of acoustic ranging.

From time step k to the next time step $k + 1$, vehicle's position x propagates in a way such that

$$\mathbf{x}_{k+1}^{(i)} = \mathbf{F}^{(i)}\bar{\mathbf{x}}_k^{(i)} + \mathbf{B}_k^{(i)}\mathbf{u}_k^i + \omega^{(i)} \quad (3.1)$$

where i is the vehicle number and $\mathbf{w}^{(i)}$ is propagation additive noise. The propagation noises are independent zero-mean Gaussian processes with covariances $\mathbf{Q}^{(i)}$. $\mathbf{B}^{(i)}$ and $\mathbf{u}^{(i)}$ are the control matrix and control input.

When AUVs send acoustic signals for ranging, they also encapsulate the information in the acoustic signals during the exchange. With a non-linear range measurement, Extended Kalman filter (EKF) [45] is firstly

formulated for the centralized system with 3 vehicles (Vehicle i , j and m). The centralized system assumes all the local propagation, local measurements and relative measurement are known to a central unit. Then a simple decentralization is used to show the problem of communication issues. A decentralized system is closer to the realistic: each vehicle only knows what happens locally; the local propagation, local measurement, and relative measurement when it happens to this vehicle. Therefore, the filter has to be designed as to what are kept locally and what are communicated. We discuss the possibility and simulate scenarios using the decentralized EKF.

The centralized EKF predicts the positions of 3 vehicles as

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{F}_k \hat{\mathbf{x}}_{k|k} + \mathbf{B}_k \mathbf{u}_k$$

$$\begin{bmatrix} \hat{\mathbf{x}}^{(i)} \\ \hat{\mathbf{x}}^{(j)} \\ \hat{\mathbf{x}}^{(m)} \end{bmatrix}_{k+1|k} = \begin{bmatrix} \mathbf{F}^{(i)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{F}^{(j)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{F}^{(m)} \end{bmatrix}_k \begin{bmatrix} \hat{\mathbf{x}}^{(i)} \\ \hat{\mathbf{x}}^{(j)} \\ \hat{\mathbf{x}}^{(k)} \end{bmatrix}_{k|k} + \begin{bmatrix} \mathbf{B}^{(i)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}^{(j)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}^{(m)} \end{bmatrix}_k \begin{bmatrix} \mathbf{u}^{(i)} \\ \mathbf{u}^{(j)} \\ \mathbf{u}^{(k)} \end{bmatrix}_{k|k} \quad (3.2)$$

where $\mathbf{0}$ is zero matrix in proper size. The centralized control input \mathbf{u} and control matrix \mathbf{B} are formulated in the same way as the centralized state vector \mathbf{x} and propagation matrix \mathbf{F} . The propagation of i th AUV error covariance is

$$\mathbf{P}_{k+1|k}^{(i)} = \mathbf{F}_k^{(i)} \mathbf{P}_{k|k}^{(i)} \mathbf{F}_k^{(i)\top} + \mathbf{Q}_k^{(i)} \quad (3.3)$$

while the error propagation of the cross-correlation term between AUV i and j is

$$\mathbf{P}_{k+1|k}^{(ij)} = \mathbf{F}_k^{(i)} \mathbf{P}_{k|k}^{(ij)} \mathbf{F}_k^{(j)\top}. \quad (3.4)$$

If the propagation model of every AUV is known to all, each AUV can keep its respective row of the centralized covariance matrix and make full propagation for the cross-correlation term. However this only applies for missions with predefined control input. In [68], authors split the cross-correlation term such that

$$\begin{aligned} \mathbf{P}_{k+1|k}^{(ij)} &= \sqrt{\mathbf{P}_{k+1|k}^{(ij)}} \sqrt{\mathbf{P}_{k+1|k}^{(ji)\top}} \\ &= \mathbf{F}_k^{(i)} \sqrt{\mathbf{P}_{k|k}^{(ij)}} (\mathbf{F}_k^{(j)} \sqrt{\mathbf{P}_{k|k}^{(ji)\top}})^\top. \end{aligned} \quad (3.5)$$

According to [68], when AUV i and AUV j meet, they exchange their distributed cross-correlation terms and get the full picture by multiplying them. Therefore vehicles do not need communicate about their respective propagation model. However the local measurement should update the correlation as well as the correlated terms of other members. Meanwhile, due to the communication packet loss, for example, AUV i may not know the information exchange happening between Vehicle j and Vehicle m .

The range measured between AUV i and m at time step $k + 1$ is r_{k+1} and is observed as

$$\begin{aligned} r_{k+1} &= h(\mathbf{x}_{k+1}) + v_{k+1} \\ &= \|\mathbf{x}_{k+1}^{(i)} - \mathbf{x}_{k+1}^{(m)}\| + v_{k+1} \end{aligned} \quad (3.6)$$

where $\|\bullet\|$ is a norm operation and $v_{k+1} \sim \mathcal{N}(0, \mathbf{R}_{r,k+1})$ is the ranging measurement error. \mathbf{R}_r is the error covariance. The observation matrix for range between AUV i and m is the Jacobian

$$\begin{aligned} \mathbf{H}_{k+1} &= \left. \frac{\partial h}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k+1|k}} \\ &= \begin{bmatrix} \mathbf{H}^{(im)} & \mathbf{0}_{1 \times 2} & -\mathbf{H}^{(im)} \end{bmatrix}_{k+1|k} \end{aligned} \quad (3.7)$$

where position state is in size 2×1 and $\mathbf{H}^{(im)} = \frac{(\hat{\mathbf{x}}^{(i)} - \hat{\mathbf{x}}^{(m)})^\top}{\|\hat{\mathbf{x}}^{(i)} - \hat{\mathbf{x}}^{(m)}\|}$. The residual covariance matrix is

$$\begin{aligned} \mathbf{S}_{k+1} &= \mathbf{H}_{k+1} \mathbf{P}_{k+1|k} \mathbf{H}_{k+1}^\top + \mathbf{R}_{r,k+1} \\ &= [\mathbf{H}_{k+1}^{(im)} (\mathbf{P}_{k+1|k}^{(i)} - \mathbf{P}_{k+1|k}^{(mi)} - \mathbf{P}_{k+1|k}^{(im)} + \mathbf{P}_{k+1|k}^{(m)}) \mathbf{H}_{k+1}^{(im)\top}] + \mathbf{R}_{r,k+1}. \end{aligned} \quad (3.8)$$

We can see that the observation matrix and the residual covariance matrix are derived solely from the information related to the two vehicles. The

Kalman gain is

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1|k} \mathbf{H}_{k+1}^\top \mathbf{S}_{k+1}^{-1}$$

$$\begin{bmatrix} \mathbf{K}^{(i)} \\ \mathbf{K}^{(j)} \\ \mathbf{K}^{(m)} \end{bmatrix}_{k+1} = \begin{bmatrix} (\mathbf{P}_{k+1|k}^{(i)} - \mathbf{P}_{k+1|k}^{(im)}) \mathbf{H}_{k+1}^\top \\ (\mathbf{P}_{k+1|k}^{(ji)} - \mathbf{P}_{k+1|k}^{(jm)}) \mathbf{H}_{k+1}^\top \\ (\mathbf{P}_{k+1|k}^{(mi)} - \mathbf{P}_{k+1|k}^{(m)}) \mathbf{H}_{k+1}^\top \end{bmatrix} \mathbf{S}_{k+1}^{-1}. \quad (3.9)$$

The state estimate is updated as

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1} \tilde{\mathbf{y}}_{k+1}$$

$$= \begin{bmatrix} \hat{\mathbf{x}}^{(i)} \\ \hat{\mathbf{x}}^{(j)} \\ \hat{\mathbf{x}}^{(m)} \end{bmatrix}_{k+1|k} + \begin{bmatrix} \mathbf{K}^{(i)} \\ \mathbf{K}^{(j)} \\ \mathbf{K}^{(m)} \end{bmatrix}_{k+1} (r_{k+1} - \|\hat{\mathbf{x}}_{k+1}^{(i)} - \hat{\mathbf{x}}_{k+1}^{(m)}\|). \quad (3.10)$$

We can see that the Kalman gain for each AUV is proportional to the difference between the cross-correlations with the exchanging pair. If vehicle j picks up the ranging and communicated information from the exchanging pairs, it can also update locally.

The covariance update is

$$\begin{aligned}
& \mathbf{P}_{k+1|k+1} \\
&= (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}_{k+1}) \mathbf{P}_{k+1|k} \\
&= \mathbf{P}_{k+1|k} - \Delta \mathbf{P}_{k+1} \\
& \quad \Delta \mathbf{P}_{k+1} \\
&= \mathbf{K}_{k+1} \mathbf{H}_{k+1} \mathbf{P}_{k+1|k} \\
&= \begin{bmatrix} \mathbf{K}^{(i)} \mathbf{H}_{k+1} (\mathbf{P}^{(i)} - \mathbf{P}^{(mi)}) & \mathbf{K}^{(i)} \mathbf{H}_{k+1} (\mathbf{P}^{(ij)} - \mathbf{P}^{(mj)}) & \mathbf{K}^{(i)} \mathbf{H}_{k+1} (\mathbf{P}^{(im)} - \mathbf{P}^{(m)}) \\ \mathbf{K}^{(j)} \mathbf{H}_{k+1} (\mathbf{P}^{(i)} - \mathbf{P}^{(mi)}) & \mathbf{K}^{(j)} \mathbf{H}_{k+1} (\mathbf{P}^{(ij)} - \mathbf{P}^{(mj)}) & \mathbf{K}^{(j)} \mathbf{H}_{k+1} (\mathbf{P}^{(im)} - \mathbf{P}^{(m)}) \\ \mathbf{K}^{(m)} \mathbf{H}_{k+1} (\mathbf{P}^{(i)} - \mathbf{P}^{(mi)}) & \mathbf{K}^{(m)} \mathbf{H}_{k+1} (\mathbf{P}^{(ij)} - \mathbf{P}^{(mj)}) & \mathbf{K}^{(m)} \mathbf{H}_{k+1} (\mathbf{P}^{(im)} - \mathbf{P}^{(m)}) \end{bmatrix}.
\end{aligned} \tag{3.11}$$

With a ranging between Vehicle i and Vehicle m , the terms highlighted in red and blue are the updates on the cross-correlations with AUV j . If vehicles keep each row locally, the cross-correlation terms need to be updated together such that they are the same (in transpose) as the one kept at the counterpart. If all the exchanged information is not acknowledged by AUV j and yet the exchanging AUVs still update their correlation with AUV j , the terms highlighted in red and blue are not same (in transpose).

There are two options when exchanging packet is lost to AUV j :

- Total ignorance: AUV j does not pick up the communications between AUV i and j and has no idea about this cooperation afterwards.

The packet gets lost completely.

- Delay and relay: The communications between AUV i and m at time $k + 1$ is logged by the exchanging AUVs and other AUVs (if there are more vehicles which pick up the cooperation). It will be used to update AUV j later when they meet. It is an ‘*Out Of Sequence Measurement*’ (*OOSM*) problem.

Three filters are tested in the simulation of cooperative localization.

They are:

- DR - Dead reckoning without any ranging and cooperation among the AUVs.
- CEKF - the centralized EKF with ranging. It tracks the full error covariance matrix of the team, gives the optimal estimation and therefore serves as a baseline for comparison.
- DEKF - the decentralized EKF with some packet loss rate to other AUVs. Each vehicle keeps its respective row in the CEKF. The centralized filter is represented as

$$\mathbf{P}_{k+1} = \begin{bmatrix} \mathbf{P}_{row,k+1}^{(i)} \\ \mathbf{P}_{row,k+1}^{(j)} \\ \mathbf{P}_{row,k+1}^{(m)} \end{bmatrix}. \quad (3.12)$$

In the simulation, for each AUV, the heading direction and heading speed (between 0.5 and 2 m/s) are randomly generated, with a low probability (1.4%) to change to a new direction and speed at each time step. The propagation noise comes from the zero-mean Gaussian noise of the velocity with 0.1 m/s standard deviation for all AUVs. At every 10 seconds, a pair of AUVs exchange their information for ranging. No local measurement is made.

3.2.2.1 Distributed EKF with Packet Loss

The advantage of cooperative localization using ranging over the group of AUVs is shown by CEKF in Figure 3.3. When DR has drifting error, the aid received from ranging information during the first 100 seconds is significant. After 100 seconds, ranging still helps by making the overall drifting slower.

When the loss rate $p_L = 0$ (Figure 3.3(a)), DEKF has the same performance as CEKF. When the loss rate increases to 40% (Figure 3.3(b)), we see that the average error by DEKF is larger than the average error given by CEKF. When p_L goes up to 50% (Figure 3.3(c)), there is a jump in estimation error observed. When p_L is even larger (Figure 3.3(d) and Figure 3.3(e)), the positioning error grows rapidly and the performance is much worse than DR. This result agree with the statement in [42]: when

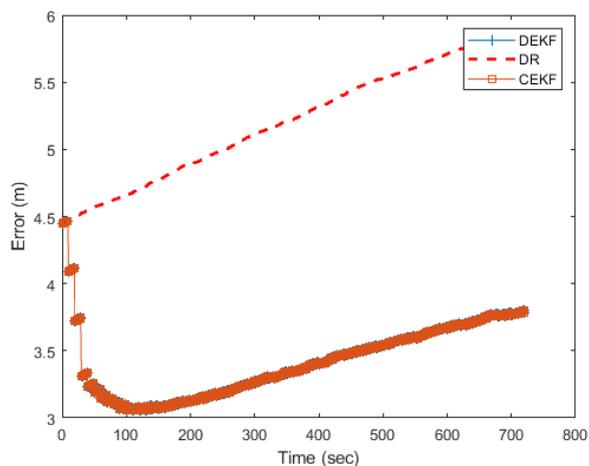
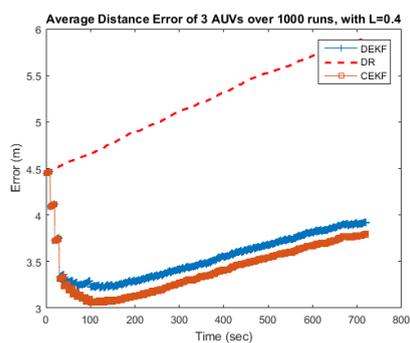
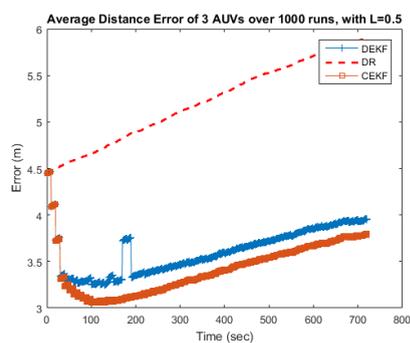
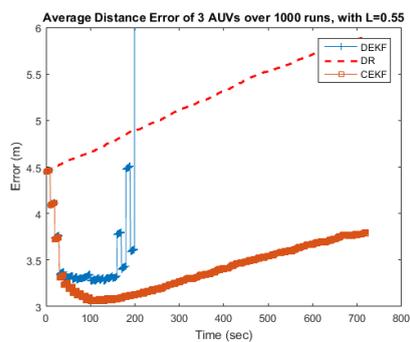
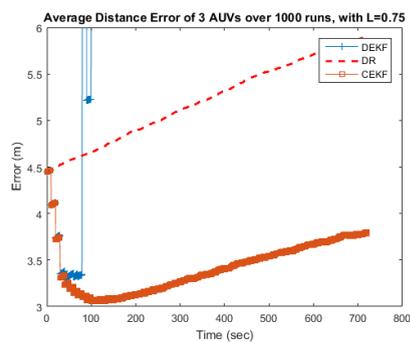
(a) $p_L = 0$ (b) $p_L = 0.4$ (c) $p_L = 0.5$ (d) $p_L = 0.55$ (e) $p_L = 0.75$

FIGURE 3.3: Average Error in Distance of 3 AUVs with various loss rate p_L : When packet loss is less, DKF improves localization through cooperation. When packet loss is higher, DKF improves localization at first but later deteriorates fast.

the correlation is ignored, estimation overconfidence arises with the fused estimate, and may lead to filter divergence.

3.2.2.2 Delay and Relay with a Simplified Model

Equations (3.10) and (3.11) show that the update of the estimate and error covariance are additive. If AUV j misses the update from the ranging between AUV i and m at time step $k+1$, it will continue predicting its estimate without the additive terms. If the propagation matrix \mathbf{F} is an identity matrix, the error covariance matrix propagates with additive process noise \mathbf{Q} only; the correction term for the missing update can be simply added to the current state estimate. If the propagation matrix is not an identity matrix, the propagation in the delayed duration has to be re-calculated to obtain the current state (this is called *retrodiction* or *backward prediction*). This becomes especially hard for nonlinear propagation model as the inverse model depends on all the past status.

We test on a simplified model with the following assumptions:

Assumption 1. *The propagation model of every AUV has identity matrix \mathbf{F} and is known to all.*

Assumption 2. *Ranging is the only available measurement.*

Let each AUV keep the details of a limited number N_p of the past updates (let $N_p = 5$ for example). The procedures and required information are as follows:

1. **Check:** When 2 AUVs communicate for ranging update, they compare and check the logs of the counterpart for the past N_p exchanges.
2. **Delayed measurement:** If any missing logs in the past are found, the current state estimate $\hat{\mathbf{x}}_{row}$ and error estimate \mathbf{P}_{row} are updated with the delayed measurement(s).
3. **Ranging:** The two AUVs then exchange information for ranging, and log the current update. At the same time, the other AUVs who successfully pick up the ranging update will also get updated and log the update.

The information exchange and update is kept at the communicating AUV i and m and other AUVs if they successfully pick up the communication. They are:

- Ranging time k_e .
- Exchangers' identity (for example, AUV i and m).
- Exchangers' position estimates $\hat{\mathbf{x}}_{k_e}^{(i)}$ and $\hat{\mathbf{x}}_{k_e}^{(m)}$.
- The row of exchangers' error covariance $\mathbf{P}_{row,k_e}^{(i)}$ and $\mathbf{P}_{row,k_e}^{(m)}$.

- The acoustic ranging r_{k_e} .
- The error covariance \mathbf{R}_{r,k_e} of the acoustic ranging.

We simulate 4 AUVs with vehicle ID as 1, 2, 3 and 4. We schedule the communication of the AUV pairs in Table 3.1. Figure 3.4 shows a special case where only AUV 4 has a loss rate $L_4 = 1$. This means AUV 4 gets the delayed ranging update of other pairs of AUVs only when it communicates with others for ranging. The DEKF of AUV 1 to 3 are the same as CEKF and is not shown here. We are only showing the RMSE of AUV 4 over 1000 runs. It can be seen that the estimation of AUV 4 gets corrected after 30 seconds once it reconnects with the other AUVs.

TABLE 3.1: Pairing Sequence for Ranging Update

Time ($\delta T = 10$ seconds)	Pair	
	i	j
10	1	2
20	2	3
30	3	4
40	4	1
50	1	2
60	2	3
\vdots	\vdots	\vdots

Figure 3.5 shows the result when all AUVs have loss rate $p_L = 0.4$. It can be seen that the logs of past $N_p = 5$ rangings are able to correct the DEKF and give an estimate which is very close to the one given by CEKF.

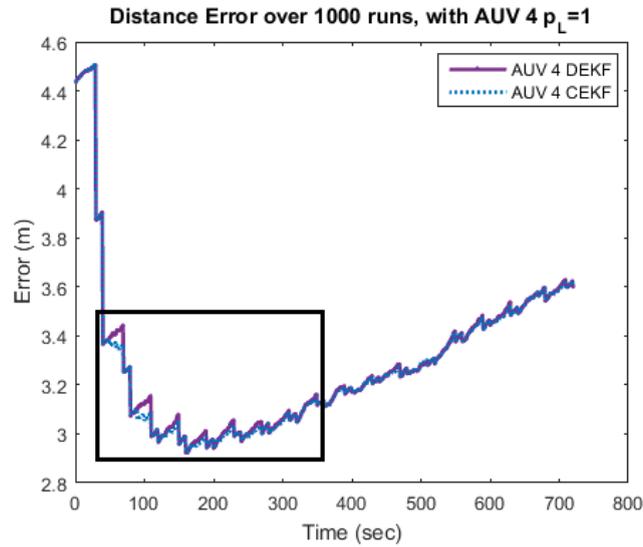
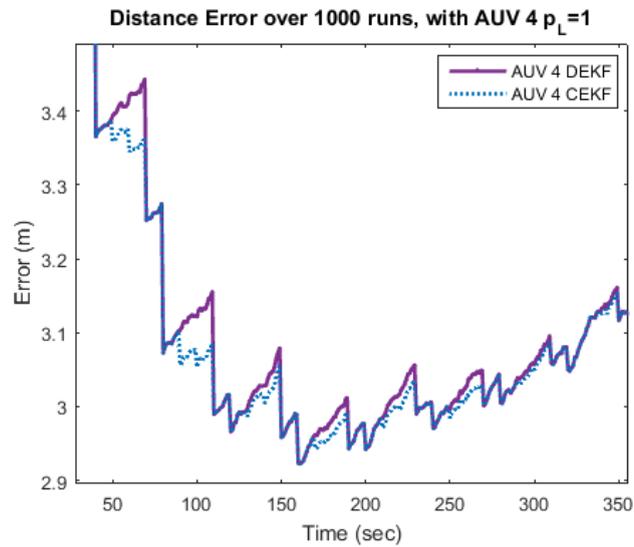
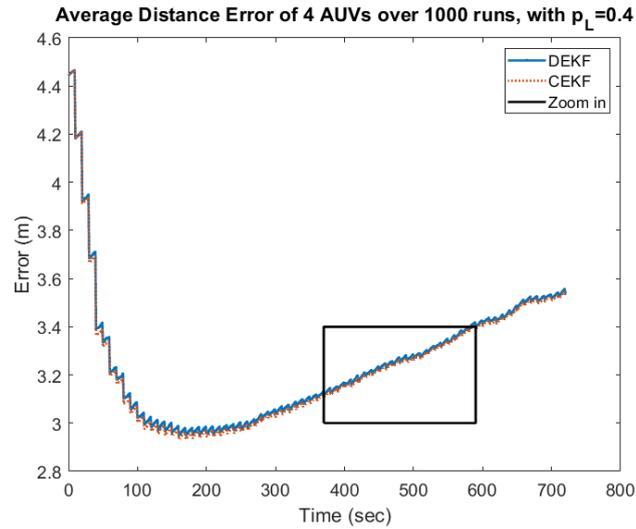
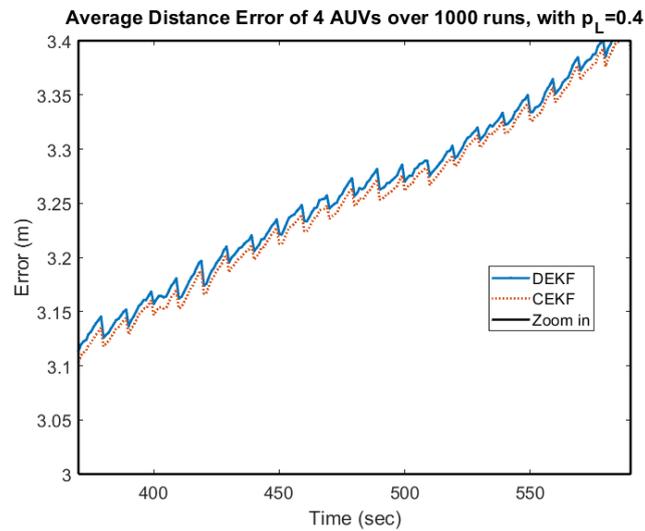
(a) RMSE of AUV 4 with $p_L = 1$ (b) Zoom-in: RMSE of AUV 4 with $p_L = 1$

FIGURE 3.4: Estimation error of AUV 4 gets corrected once it reconnects with the other AUVs after 30 seconds.

(a) RMSE of 4 AUVs with $p_L = 0.4$ (b) Zoom-in: RMSE of 4 AUVs with $p_L = 0.4$ FIGURE 3.5: Logs of past $N_p = 5$ rangings are able correct the DEKF and give similar result to CEKF.

In fact, the pairing sequence is critical in the delay and relay. As long as $N_p \geq N - 2$ where N is the number of AUVs in the team, the controlled pairing sequence can guarantee the circulation of the information among the team over a cycle.

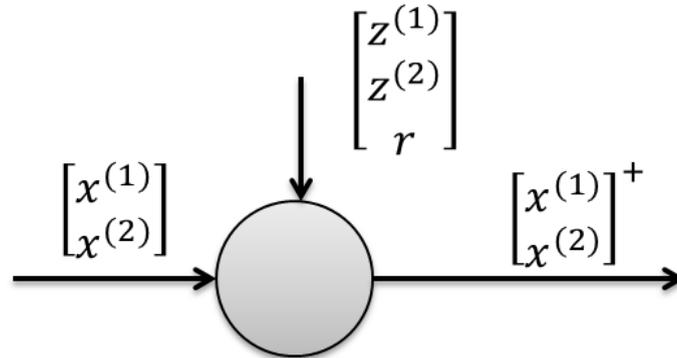
Compared to terrestrial communication, underwater communication uses acoustic waves instead of electromagnetic waves. It has problems such as multi-path propagation, time variations of the channel, small available bandwidth and strong signal attenuation especially over long ranges. Therefore the communication has low data rates. In such a case, successful ranging has irregular time interval and random pairing sequence. It is possible that AUVs miss the past ranging update without any relay. It is also possible that an AUV gets ‘out of sequence measurement’ (OOSM). The whole communication scheme (time and sequence) becomes complicated and unpredictable.

Meanwhile, only the pre-planned missions are known to each vehicle. The actual paths and activities change with respect to the situation and may not be updated to every other member. Vehicles can also make local measurement to update their position estimates. All these possibilities make the correlation untrackable in the distributed processing .

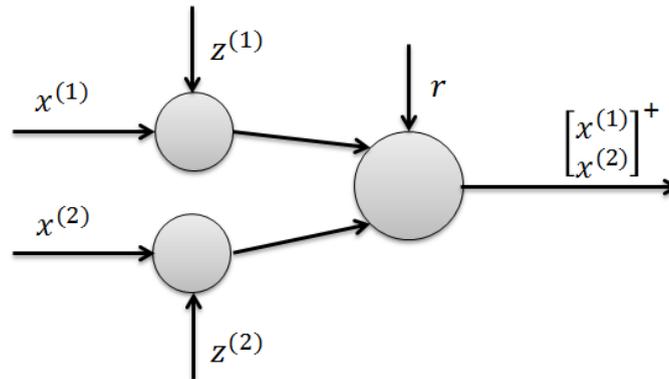
3.3 Information Loss in Distributed Processing

In the previous section, each member keeps a row of the centralized state vector and error covariance. Vehicles may still lose track of the exact correlation with each other. In this section, each vehicle only keeps estimates about itself and we assume the inter-vehicle correlation is known exactly. Examples show that compared with centralized processing, the distributed processing has some information loss even when the correlation is tracked precisely.

The simplest example consists of two vehicles estimating their locations in 1-dimensional space. Their initial correlation coefficient ranges from $[0 : 0.1 : 0.9]$. Both vehicles follow the 3 steps: 1) propagate with noise, 2) make local measurements about their positions, and 3) communicate their positions along with range measurements. Figure 3.6 shows the centralized Kalman filter (CKF) and decentralized Kalman filter (DKF) estimation architectures. The positions of Vehicle 1 and 2 are considered as random variables as they evolve at each succeeding moments with random process noise. Each vehicle makes a local measurement, followed by a relative



(a) Central processing (CKF)



(b) Decentralized processing (DKF)

FIGURE 3.6: Central processing architecture vs. distributed processing architecture. CKF fuses the raw measurements directly while DKF fuses the local processed data.

measurement

$$\begin{aligned}
 \mathbf{z}^{(1)} &= \mathbf{x}^{(1)} + \nu^{(1)}, \\
 \mathbf{z}^{(2)} &= \mathbf{x}^{(2)} + \nu^{(2)}, \\
 \mathbf{r} &= h(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) + v.
 \end{aligned}
 \tag{3.13}$$

A central estimation (Figure 3.6(a)) uses the aggregated state and aggregated measurements. The centralized error covariance is

$$\mathbf{P}_{\text{CKF}} = \left(\begin{bmatrix} \mathbf{P}^{(1)} & \mathbf{P}_{12} \\ \mathbf{P}_{12}^\top & \mathbf{P}^{(2)} \end{bmatrix}^{-1} + \mathbf{H}_{\text{CKF}}^\top \begin{bmatrix} \mathbf{R}^{(1)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^{(2)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R}_r \end{bmatrix}^{-1} \mathbf{H}_{\text{CKF}} \right)^{-1} \quad (3.14)$$

where $\mathbf{H}_{\text{CKF}} = \frac{\partial \mathbf{z}}{\partial \mathbf{x}}$. DKF (Figure 3.6(b)) only sends over the processed estimate and error covariance after local updates. We assume the initial correlation is exactly known. Therefore, both filters trace the correlation and therefore estimation are strictly consistent. Figure 3.7 shows the estimated error covariance after the two processing architectures in Figure 3.6. The initial error covariances are $\mathbf{P}^{(1)} = 4$, $\mathbf{P}^{(2)} = 4$ respectively. The relative measurement has an error covariance $\mathbf{R} = \mathbf{1}$. We can see that DKF has larger error covariance except when the initial correlation coefficient is 0. With a larger initial correlation, the processing of the local measurements before fusion results in a larger gap in the estimation error compared with CKF.

We use information entropy to explain the performance of DKF. Entropy refers to the uncertainty associated with a probability distribution, and is a measure of the descriptive complexity of a PDF. Mathematically

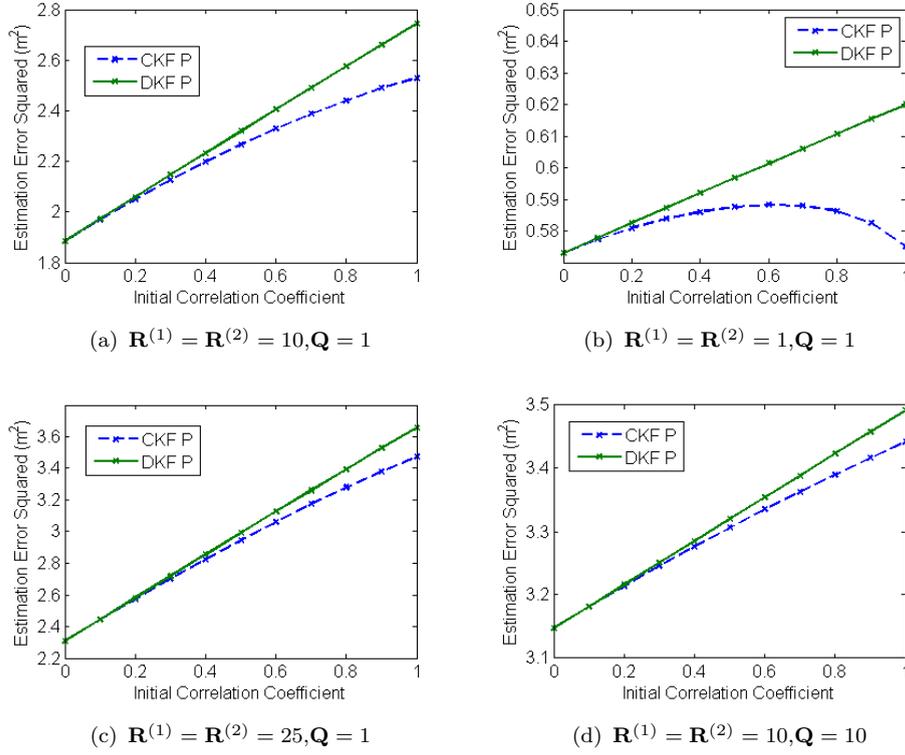


FIGURE 3.7: Estimation error squared versus initial correlation coefficient. DKF has larger error covariance than CKF except when initial correlation coefficient is 0.

it is expressed as

$$h\{F(\mathbf{x})\} \triangleq \mathbb{E}\{-\ln p(\mathbf{x})\} \quad (3.15)$$

where \mathbf{x} is a random variable. With the measurement \mathbf{z} , the entropy of the conditional probability is

$$\mathbb{E}\{-\ln[p(\mathbf{x}|\mathbf{z})]\} = \mathbb{E}\{-\ln[p(\mathbf{x})]\} - \mathbb{E}\{\ln\left[\frac{p(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})}\right]\} \quad (3.16)$$

$$h(\mathbf{x}|\mathbf{z}) = h(\mathbf{x}) - I(\mathbf{z}; \mathbf{x}).$$

It states that the entropy following an observation is reduced by an amount equal to the information inherent in the observation [1].

CKF gives

$$\begin{aligned}
h_{\text{CKF}} &= h(\mathbf{x}|\mathbf{z}) \\
&= h(\mathbf{x}) - I(\mathbf{z}; \mathbf{x}) \\
&= [h(\mathbf{x}^{(1)}) + h(\mathbf{x}^{(2)}) - I(\mathbf{x}^{(1)}; \mathbf{x}^{(2)})] - [I(\mathbf{z}^{(1)}; \mathbf{x}^{(1)}) + I(\mathbf{z}^{(2)}; \mathbf{x}^{(2)}) + I(\mathbf{r}; \mathbf{x})].
\end{aligned} \tag{3.17}$$

DKF gives

$$\begin{aligned}
h_{\text{DKF}} &= h(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}|\mathbf{z}^{(1)}, \mathbf{z}^{(2)}) - I(\mathbf{r}; \mathbf{x}^{(1)}, \mathbf{x}^{(2)}) \\
&= [h(\mathbf{x}^{(1)}|\mathbf{z}^{(1)}, \mathbf{z}^{(2)}) + h(\mathbf{x}^{(2)}|\mathbf{z}^{(1)}, \mathbf{z}^{(2)}) - I((\mathbf{x}^{(1)}; \mathbf{x}^{(2)}|\mathbf{z}^{(1)}, \mathbf{z}^{(2)})] - I(\mathbf{r}; \mathbf{x}) \\
&= [h(\mathbf{x}^{(1)}|\mathbf{z}^{(1)}) + h(\mathbf{x}^{(2)}|\mathbf{z}^{(2)}) - I((\mathbf{x}^{(1)}; \mathbf{x}^{(2)}|\mathbf{z}^{(1)}, \mathbf{z}^{(2)})] - I(\mathbf{r}; \mathbf{x}) \\
&= [h(\mathbf{x}^{(1)}) - I(\mathbf{z}^{(1)}; \mathbf{x}^{(1)})] + [h(\mathbf{x}^{(2)}) - I(\mathbf{z}^{(2)}; \mathbf{x}^{(2)})] - I((\mathbf{x}^{(1)}; \mathbf{x}^{(2)}|\mathbf{z}^{(1)}, \mathbf{z}^{(2)})) - I(\mathbf{r}; \mathbf{x}).
\end{aligned} \tag{3.18}$$

The difference between these two expressions is $I(\mathbf{x}^{(1)}; \mathbf{x}^{(2)}) \geq I(\mathbf{x}^{(1)}; \mathbf{x}^{(2)}|\mathbf{z}^{(1)}, \mathbf{z}^{(2)})$,

and makes $h_{\text{CKF}} \leq h_{\text{DKF}}$. The equality happens only when $I(\mathbf{x}^{(1)}; \mathbf{x}^{(2)}) = 0$.

The mutual information between the states is reduced by conditioning on local measurements. There is information loss in only transmitting the processed up-to-date estimates, instead of raw sensor data.

In another way, we elaborate this in the context of multivariate distribution. Assuming the cross-correlation can be tracked, the mutual information $I(\mathbf{x}^{(1)}; \mathbf{x}^{(2)})$ is expressed as

$$I(\mathbf{x}^{(1)}; \mathbf{x}^{(2)}) = -\frac{1}{2} \ln |\Sigma_\rho| \quad (3.19)$$

where Σ_ρ is the correlation matrix constructed from the covariance matrix $\mathbf{P} = \begin{bmatrix} \mathbf{P}^{(1)} & \mathbf{P}^{(12)} \\ \mathbf{P}^{(12)\top} & \mathbf{P}^{(2)} \end{bmatrix}$. The entries of the correlation matrix records the Pearson product-moment correlation coefficients between the random variables. In 1-dimensional space, $I(\mathbf{x}^{(1)}; \mathbf{x}^{(2)}) = -\frac{1}{2} \ln(1 - \rho^2)$ where ρ is the correlation coefficient. After local measurement update, the new central error covariance is given as

$$\bar{\mathbf{P}} = \left(\begin{bmatrix} \mathbf{P}^{(1)} & \rho\sqrt{\mathbf{P}^{(1)}\mathbf{P}^{(2)}} \\ \rho\sqrt{\mathbf{P}^{(1)}\mathbf{P}^{(2)}} & \mathbf{P}^{(2)} \end{bmatrix}^{-1} + \begin{bmatrix} \mathbf{R}^{(1)} & 0 \\ 0 & \mathbf{R}^{(2)} \end{bmatrix}^{-1} \right)^{-1}. \quad (3.20)$$

The new correlation coefficient for $\bar{\mathbf{P}}$ is

$$\begin{aligned} \bar{\rho} &= \frac{\rho}{\sqrt{\left(1 + \frac{\mathbf{P}^{(2)}}{\mathbf{R}^{(2)}}(1 - \rho^2)\right)\left(1 + \frac{\mathbf{P}^{(1)}}{\mathbf{R}^{(1)}}(1 - \rho^2)\right)}} \\ &\leq \rho. \end{aligned} \quad (3.21)$$

Therefore we have

$$I(\mathbf{x}^{(1)}; \mathbf{x}^{(2)} | \mathbf{z}^{(1)}, \mathbf{z}^{(2)}) \leq I(\mathbf{x}^{(1)}; \mathbf{x}^{(2)}) \quad (3.22)$$

$$h_{\text{CKF}} \leq h_{\text{DKF}}.$$

It means CKF ends with less uncertainty in the state $[\mathbf{x}^{(1)\top}, \mathbf{x}^{(2)\top}]^\top$, or equivalently smaller estimation error. We can understand this problem as information loss in pre-processing of the raw data. If we want DKF to achieve CKF performance, we need to transmit all the past local propagation and measurement data.

3.4 Distributed Extended Information Filter

In Section 2.4, we have defined the fusion principles. It means that information fusion through cooperation must improve the individual performance and consistent estimation is preferred. Existing method either ignores the correlation from different sources or overestimate the correlation. The former is called *naïve* filter. The later such as covariance intersection method [44] and its related [52, 67], ellipsoid intersection method [75] and largest ellipsoid algorithm [15] assume maximum correlation when dealing with information from unknown sources.

We use extended information filter (EIF) - an inverse covariance form of the Kalman filter, to separate the independent part of the local information. The reason is that the local prediction and measurement information can be encapsulated into a single message and acoustically transmitted with bounded size. The decentralized EIF was first implemented in a single-beacon cooperative localization in [31] where the server (an surface vehicle) sends the encapsulated information while performing ranging with the client (underwater vehicle). It can handle asynchronous broadcasts from the server but the information flows in one direction only, that is, from server to clients. To accommodate more AUVs operating over very large operational areas without surface beacons, we propose a new cooperative multi-AUV localization algorithm using distributed EIF (DEIF). We describe the detailed design and implementation for a team of cooperative AUVs, where no single AUV functions as a beacon possessing accurate position information. The proposed method is designed to record the correlated information from the most recent cooperation, providing consistent position estimates in event of packet loss.

3.4.1 Illustrative Examples

In this section, we use some simple examples to illustrate the problems with traditional approaches and show how our proposed DEIF overcomes these problems.

3.4.1.1 Limited Bandwidth

The estimation result of EIF for the 3-step problem (Figure 3.6) is shown in Figure 3.8.

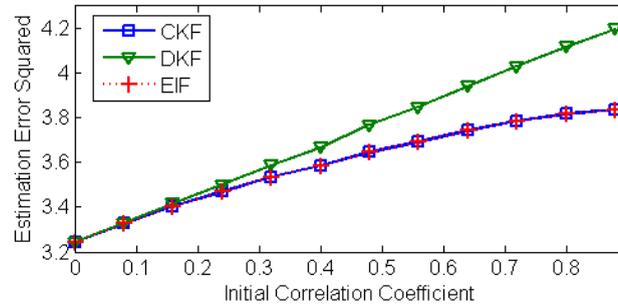


FIGURE 3.8: Traditional distributed processing (DKF) performs poorly as compared to centralized processing (CKF), but our proposed distributed method (DEIF) is able to perform well.

As discussed, there is information loss in the traditional DKF even if the correlation is precisely known. To maintain the same performance as the centralized estimation, traditional DKF requires a full storage and transmission of the historical information. If there are n steps of local propagation and measurements before the cooperation, the packet size for transmission will increase as $\mathcal{O}(n)$. Our proposed method using DEIF is able to avoid the information loss that DKF suffers, and perform as well as the CKF using transmissions of fixed, small-sized packets.

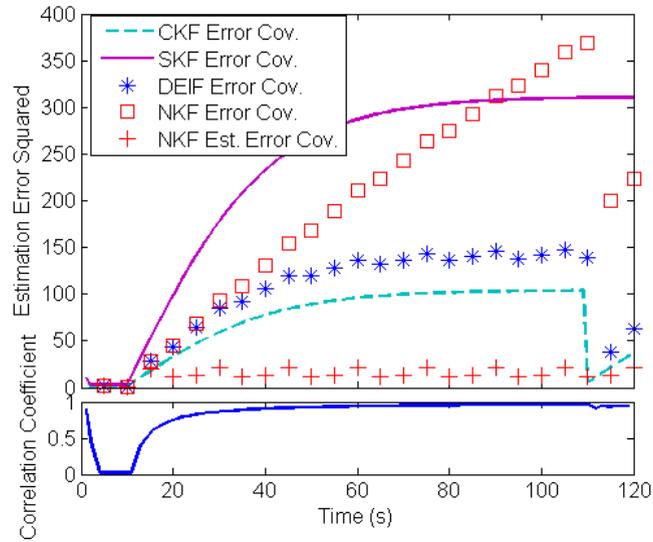


FIGURE 3.9: Estimation error of AUV 1 using various filters in a 3-AUV cooperative localization example. DEIF is close to CKF.

3.4.1.2 Inter-Vehicle Correlation

We demonstrate the danger of ignoring the correlation in a 3-AUV cooperative localization. Let AUVs broadcast their state estimates in a round-robin fashion. The estimation error of AUV 1 is shown in Figure 3.9. At 10 seconds, all AUVs submerge and lose GPS position measurements, but continue to communicate with each other. At 90 seconds, AUV 2 obtains a high-quality position measurement (say, by surfacing and obtaining a GPS fix). AUV 1’s localization is improved by fusing estimates from AUV 2. A *naïve* Kalman filter (NKF) simply ignores the correlation among vehicles; it assumes an improvement in estimate by fusing data from ‘independent’ sources, while in fact there is no improvement from double counting the shared information. The estimated error covariance of NKF appears to be

very low, but the actual estimation error diverges quickly. Its performance can be even worse than single vehicle localization (SKF). Our proposed method (DEIF) performs well, and produces results that are quite close to the ideal (but impossible) CKF.

3.4.2 Formulation and Design

The vehicle propagation and measurement models follow Equation (2.1) and Equation (2.2).

In the multi-vehicle cooperation, a vehicle encodes its information into acoustic packets and broadcasts as a Peer Vehicle (PV). Other vehicles in the team receive the packets as Receiving Vehicles (RVs). When the team is synchronized, the one-way travel time of the acoustic signals is simply the difference between the time-of-launch and time-of-arrival. The distance from the PV with position \mathbf{x}_P to an RV with position \mathbf{x}_R is obtained, given the propagation speed of underwater signals. The observation model is

$$r_k = \|\mathbf{x}_{k,R} - \mathbf{x}_{k,P}\| + v_k \quad (3.23)$$

where the operator $\|\cdot\|$ denotes the Euclidean norm, and v_k is the zero-mean Gaussian noise with covariance \mathbf{R}_k . To minimize the effect of nonlinearity, we assume the vehicles are far away from each other so that the error in position estimation can be modeled as a 2-dimensional zero-mean Gaussian

random variable, after the relative measurement update. We assume all the noises are independent Gaussian noise.

The Gaussian-distributed position state vector \mathbf{x} with error covariance \mathbf{P} has the associated information matrix and information vector in an EIF as

$$\begin{aligned}\Lambda &= \mathbf{P}^{-1} \\ \eta &= \Lambda \mathbf{x}.\end{aligned}\tag{3.24}$$

At time step k , each vehicle keeps an information set $(\mathbf{x}_k, \mathbf{P}_k, \Lambda_p, \eta_p)$. \mathbf{x}_k is the combined 3-state vector $\mathbf{x}_k = [x_k^\top, x_t^\top, x_c^\top]^\top = [x_k^\top, \mathbf{x}_p^\top]^\top$ where $\mathbf{x}_p = [x_t^\top, x_c^\top]^\top$. x_k is the current position state. t denotes the time step when the most recent cooperation is made. During this cooperation, if a broadcast from a PV with position state x_c is received, this vehicle's position is updated as x_t . If this vehicle broadcasts as a PV, x_c is dummy and x_t is considered fully correlated with the team. (Λ_p, η_p) is the corresponding information pair (information matrix and vector) of \mathbf{x}_p by Equation (3.24).

3.4.2.1 Initialization

At time step $k = 0$, the initial position x_0 is assigned to x_t . We assume that all vehicles are deployed independently, the initial position has no correlation with the team and x_c is dummy.

3.4.2.2 Local Prediction

Let (Λ_k, η_k) be the information pair for $\mathbf{x}_k = [x_k^\top, \mathbf{x}_p^\top]^\top$, and x_{k+1} be the predicted state from x_k according to the propagation model. We augment the state vector with the predicted state and we have $\tilde{\mathbf{x}}_{k+1} = [x_{k+1}^\top, x_k^\top, \mathbf{x}_p^\top]^\top$.

The augmented state vector has an associated information pair given by

$$\tilde{\Lambda}_{k+1} = \begin{bmatrix} \mathbf{Q}_k^{-1} & -\mathbf{Q}_k^{-1}\mathbf{F}_k & \mathbf{0} \\ -\mathbf{F}_k^\top \mathbf{Q}_k^{-1} & \mathbf{F}_k^\top \mathbf{Q}_k^{-1} \mathbf{F}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & & \Lambda_k \\ \mathbf{0} & & \end{bmatrix}$$

$$\tilde{\eta}_{k+1} = \begin{bmatrix} \mathbf{Q}_k^{-1}(f(\hat{\mathbf{x}}_k, \mathbf{a}_k) - \mathbf{F}_k \hat{\mathbf{x}}_k) \\ -\mathbf{F}_k^\top \mathbf{Q}_k^{-1}(f(\hat{\mathbf{x}}_k, \mathbf{a}_k) - \mathbf{F}_k \hat{\mathbf{x}}_k) \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \eta_k \end{bmatrix}.$$

We can see that in an EIF, the prediction information is contained in the first term of the addition, with zeros padded accordingly. Meanwhile, we only see non-zero entries in the off-diagonal blocks between states at consecutive time steps. This agrees with Markov process assumption stating that prediction for the future state solely depends on the most recent state.

The 3-state vector $\mathbf{x}_{k+1} = [x_{k+1}^\top, x_t^\top, x_c^\top]^\top$ and associated covariance \mathbf{P}_{k+1} are predicted as

$$\begin{aligned} \hat{x}_{k+1} &= f(\hat{x}_k, \mathbf{a}_k) \\ \mathbf{P}_{k+1} &= \begin{bmatrix} \mathbf{F}_k & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \mathbf{P}_k + \begin{bmatrix} \mathbf{Q}_k & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \end{aligned} \quad (3.25)$$

where \mathbf{I} is the identity matrix in proper size.

3.4.2.3 Local Measurement Update

The measurement matrix \mathbf{H}_k is sparse as it only affects a few subblocks within the corresponding entry for x_k in the information pair. It allows an additive update to the delta information as

$$\begin{aligned} \Lambda_k^+ &= \Lambda_k + \mathbf{H}_k^\top \mathbf{R}_k^{-1} \mathbf{H}_k \\ \eta_k^+ &= \eta_k + \mathbf{H}_k^\top \mathbf{R}_k^{-1} (z_k - h(\hat{\mathbf{x}}_k) - \mathbf{H}_k \hat{\mathbf{x}}_k) \end{aligned} \quad (3.26)$$

where the superscript “+” means the observation is given up to and including time step k . It is noted that the local measurement may not be available at every time step.

The local measurement updates \mathbf{x}_k and \mathbf{P}_k in a standard Kalman filter way.

3.4.2.4 Cooperative Localization with Relative Measurement

We denote the information set kept at the PV as $(\mathbf{x}_k, \mathbf{P}_k, \Lambda_p, \eta_p)_P$. Similarly, the subscript to the information set is R for the RV. It should be noted that in a team containing more than two vehicles, the time step t recorded at PV and RV may not be the same.

Delta information for cooperation

Both PV and RV form their delta information such that

$$\begin{aligned} \Delta\Lambda_k &= \Lambda_k - \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \Lambda_p \end{bmatrix} \\ \Delta\eta_k &= \eta_k - \begin{bmatrix} \mathbf{0} \\ \eta_p \end{bmatrix} \end{aligned} \tag{3.27}$$

The PV broadcasts the delta information $(\Delta\Lambda_k, \Delta\eta_k)_P$ together with $(\mathbf{x}_k, \mathbf{P}_k)_P$ in a packet. The RV receives the packet and obtains the acoustic ranging as well.

Incorporating the delta information

When the broadcast from PV is received, the RV firstly forms a information pair corresponding to the combined state vector $[x_{k,R}^\top, x_{t,R}^\top, x_{t,P}^\top, x_{k,P}^\top]^\top$. The

information matrix consists of three parts: the delta information from the PV $\Delta\Lambda_{k,R}$, the delta information from the RV $\Delta\Lambda_{k,P}$, and the information matrix $\bar{\Lambda}_t$ corresponding to $[x_{t,R}^\top, x_{t,P}^\top]^\top$. $\bar{\Lambda}_t$ is corresponding to $[x_{t,R}^\top, x_{t,P}^\top]^\top$.

Assuming the states $x_{c,R}$ and $x_{c,P}$ are from the same source and fully correlated, a bounding joint covariance for states $x_{t,R}$ and $x_{t,P}$ can be derived. We use split covariance intersection (SCI) [52] to form the bounding covariance. (In a comparative study later, we show the benefit introduced by the delta information in DEIF over a pure SCI filter.) At both PV and RV sides, given the covariance matrix $\begin{bmatrix} \mathbf{P}_t & \mathbf{P}_{tc} \\ \mathbf{P}_{ct} & \mathbf{P}_c \end{bmatrix}$ (corresponding to $[x_t^\top, x_c^\top]^\top$), the split form for x_t is

$$\begin{aligned} \mathbf{P}_t &= \mathbf{P}_{\text{IND.}} + \mathbf{P}_{\text{DEP.}} \\ \mathbf{P}_{\text{IND.}} &= \mathbf{P}_t - \mathbf{P}_{tc}\mathbf{P}_c^{-1}\mathbf{P}_{ct} \\ \mathbf{P}_{\text{DEP.}} &= \mathbf{P}_{tc}\mathbf{P}_c^{-1}\mathbf{P}_{ct}. \end{aligned} \tag{3.28}$$

The reason is that $x_t - \mathbf{P}_{tc}\mathbf{P}_c^{-1}x_c$ and x_c are independent, and x_c represents the source of information shared from the team.

The split covariance matrix for $[x_{t,R}^\top, x_{t,P}^\top]^\top$ is therefore

$$\begin{bmatrix} \mathbf{P}_{\text{IND.,R}} + \frac{\mathbf{P}_{\text{DEP.,R}}}{\kappa} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{\text{IND.,P}} + \frac{\mathbf{P}_{\text{DEP.,P}}}{1-\kappa} \end{bmatrix}_t, \tag{3.29}$$

and $\kappa \in [0, 1]$. The value of κ is obtained by minimizing the error covariance of RV after the relative measurement update. The corresponding information pair for $[x_{t,R}^\top, x_{t,P}^\top]^\top$ is formed as

$$\bar{\Lambda}_t = \begin{bmatrix} \mathbf{P}_{\text{IND.,R}} + \frac{\mathbf{P}_{\text{DEP.,R}}}{\kappa} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{\text{IND.,P}} + \frac{\mathbf{P}_{\text{DEP.,P}}}{1-\kappa} \end{bmatrix}_t^{-1} \quad (3.30)$$

$$\bar{\eta}_t = \bar{\Lambda}_t \mathbb{E}([x_{t,R}^\top, x_{t,P}^\top]^\top).$$

The information pair $(\bar{\Lambda}_k, \bar{\eta}_k)$ corresponding to the combined state vector $[x_{k,R}^\top, x_{t,R}^\top, x_{t,P}^\top, x_{k,P}^\top]^\top$ is formed in Equation (3.31), where $\Delta\eta_{k,P}^*$ and $\Delta\Lambda_{k,P}^*$ are the rearranged $\Delta\eta_{k,P}$ and $\Delta\Lambda_{k,P}$, according to the reversed sequence $[x_{t,P}^\top, x_{k,P}^\top]^\top$. Zeros are padded before and after where needed. The zero padding and addition are illustrated in Figure 3.10.

$$\begin{aligned}
\bar{\Lambda}_k &= \begin{bmatrix} \Delta\Lambda_{k,R} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \cdots & \bar{\Lambda}_t & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \Delta\Lambda_{k,P}^* \end{bmatrix} \\
\bar{\eta}_k &= \begin{bmatrix} \Delta\eta_{k,R} \\ \vdots \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \vdots \\ \bar{\eta}_t \\ \vdots \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \vdots \\ \Delta\eta_{k,P}^* \end{bmatrix}
\end{aligned}
\tag{3.31}$$

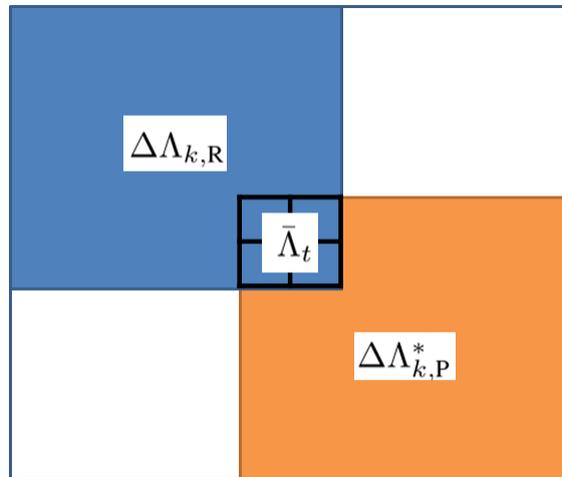


FIGURE 3.10: Illustration of incorporating delta information in simple addition.

Relative measurement update

With the information pair $(\bar{\Lambda}_k, \bar{\eta}_k)$, a measurement update can be made in the same way as Equation (3.26).

Information set update

After broadcasting out its information set, PV considers its state x_k as fully correlated with the team, and assigns it to x_t . States at time steps prior to k are discarded. The RV assigns the updated $x_{k,R}$ to x_t . The received $x_{k,P}$ is recorded as x_c . At both sides, the most recent cooperation time t is assigned the value of k . The corresponding information pair (Λ_p, η_p) is recorded.

3.4.3 Simulation Studies Using Field Experiment Data

We compare the performance of the proposed DEIF with several other methods using both simulated data and experimented data for a team of three vehicles. The experimented data was collected from a team of three vehicles executing lawnmower surveys in Singapore waters (Figure 3.11).

While executing the planned path, vehicles experience propagation noise

introduced by choppy water, system hardware and so on. Local measurements such as GPS positions (if the vehicle is on the surface), or bathymetric measurements (for submerged vehicles in a known terrain), are fused to improve localization. Vehicles may not get good local measurements about their positions all the time. For example, vehicles may submerge throughout the mission, or the sea bottom is smooth without much variation to provide rich bathymetry information. The cooperation happens when one vehicle broadcasts and the other two vehicles receive the information set.

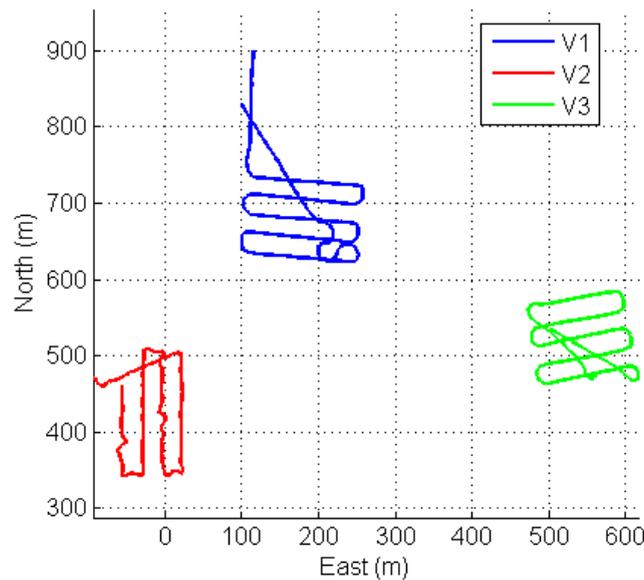


FIGURE 3.11: Cooperative localization results with field data.

When the inter-vehicle correlation is unknown due to packet loss, we show that DEIF performs better than the filter ignoring the correlation or overestimating the correlation. The effectiveness and advantages will be demonstrated by illustrative examples and comparative results from simulated and experimented data as follows.

3.4.3.1 Simulated Data

The simulated data mimics the experimented data, using identical sensor characteristics and the same trajectories. In this simulation, vehicles take turns to broadcast their information every 10 seconds. The transmission packets are lost at a rate of p_L . All vehicles cruise on surface and have GPS fixes in the first 100 seconds. Only Vehicle 2 re-surfaces at 420 seconds for 50 seconds. The results are evaluated in two metrics: the normalized estimation error squared (NEES) and root mean square error (RMSE). The NEES provides a measurement of estimation consistency [12]. Under ideal conditions, the NEES has a degree-of-freedom (DoF) equal to the dimension of the state (in our case, DoF is 2). The RMSE records the estimated error in distance, compared with the true position. Figure 3.12 shows the NEES and RMSE over 10 runs for Vehicle 3 at different packet loss rate. The arrows indicate the successful reception of the broadcast.

SKF stands single Kalman filter. NKF stands for *naïve* Kalman filter. It claims to have the lowest RMSE but has severe problem with estimation consistency. This is especially the case when vehicles communicate at high frequency (with lower packet loss rate). The estimation error given by NKF is in fact much higher than the error it claims to be. When vehicles seldom communicate and are mostly independent of each (Figure 3.12(d)), the *naïve* assumption by NKF is almost met, and therefore a consistent

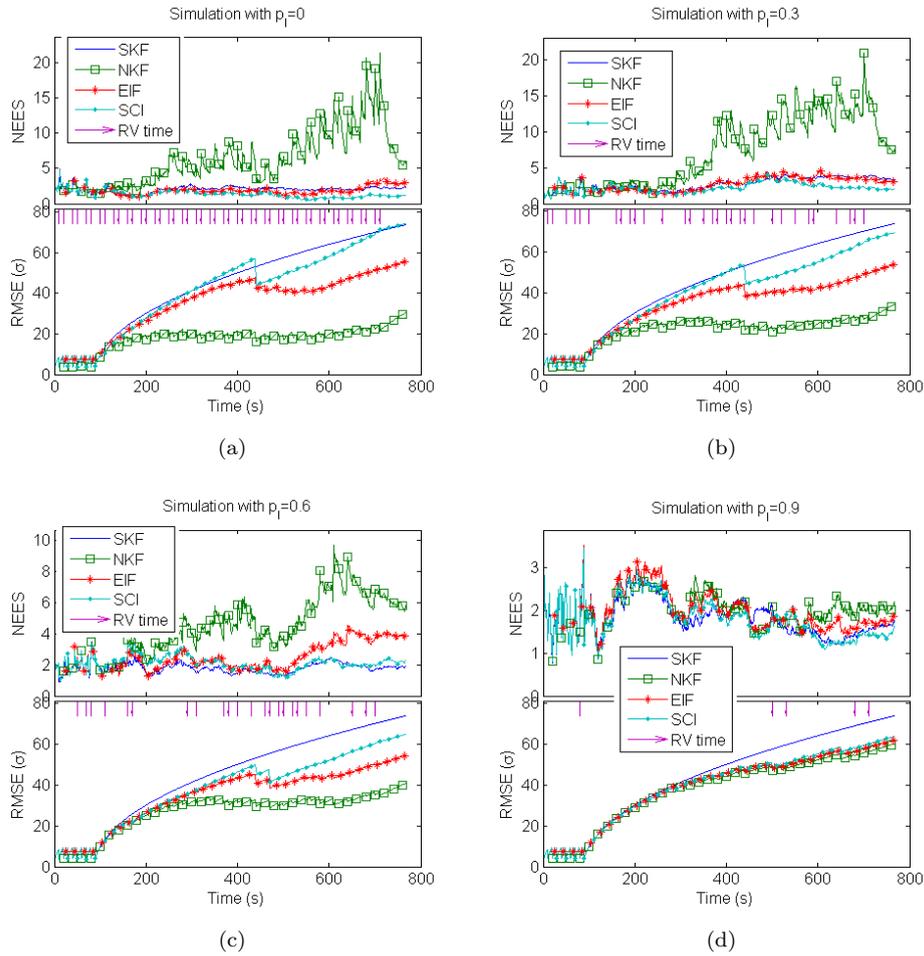


FIGURE 3.12: Simulation results for Vehicle 3 at packet loss rate (a) $p_l = 0$ (b) $p_l = 0.3$ (c) $p_l = 0.6$ and (d) $p_l = 0.9$. The vertical arrows show the time when Vehicle 3 receives broadcast. DEIF has smaller estimation error than SKF and SCI filters, and better consistency than NKF.

estimation is given.

We also compare the proposed method with SCI filter from [52]. In the SCI filter, assuming each state consists of correlated component and

independent component, whose covariances are

$$\mathbf{P}_{PV} = \mathbf{P}_{IND.,P} + \mathbf{P}_{DEP.,P} \quad (3.32)$$

$$\mathbf{P}_{RV} = \mathbf{P}_{IND.,R} + \mathbf{P}_{DEP.,R}.$$

As the range-only measurement is not enough to formulate a full estimate of the RV position, SCI filter in [52] can not be applied directly for the cooperation. We follow the idea of SCI and form a consistent covariance $\bar{\mathbf{P}}$ for the combined state vector $[x_{RV}^\top, x_{PV}^\top]^\top$. The $\bar{\mathbf{P}}$ is therefore

$$\begin{aligned} \bar{\mathbf{P}} &= \begin{bmatrix} \mathbf{P}_{IND.,R} + \frac{\mathbf{P}_{DEP.,R}}{\kappa} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{IND.,P} + \frac{\mathbf{P}_{DEP.,P}}{1-\kappa} \end{bmatrix} \\ &= \bar{\mathbf{P}}_{IND.} + \bar{\mathbf{P}}_{DEP.}. \end{aligned} \quad (3.33)$$

The range measurement is used to update the combined state vector in standard Kalman filter way. The independent component for RV is obtained in the corresponding entries of the combined independent component

$$\bar{\mathbf{P}}_{IND.}^+ = (\mathbf{I} - \mathbf{K}) \begin{bmatrix} \mathbf{P}_{IND.,R} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{IND.,P} \end{bmatrix} (\mathbf{I} - \mathbf{K})^\top + \mathbf{K}\mathbf{R}\mathbf{K}^\top. \quad (3.34)$$

We can see that with more frequent cooperation (lower packet loss rate), SCI tends to be over conservative about its estimation. In the other hand, EIF estimation maintains good consistency and performs better than SCI

on all occasions.

3.4.3.2 Experiment Data

The navigation data is used in the offline processing to compare different estimation filters. In the experiment, GPS logs are used as the benchmark to compute positioning errors. The arrows in Figure 3.13 indicate the time when the broadcasts are sent and successfully received by other vehicles. In the first 100 seconds, Vehicles 1 and 3 have good local measurements. Vehicle 2 exhibits a slow position drift with the information shared by Vehicles 1 and 3. Vehicle 2 obtains good local measurements from 200 to 300 seconds, and from 600 seconds onward. The proposed DEIF successfully improves estimation accuracy of all vehicles. In the two highlighted boxes, we can see that Vehicles 1 and 3 get position improvements from the broadcast given by Vehicle 2. Meanwhile, Vehicle 2 also benefits from the information sharing (at 400 seconds). On the other hand, the localization improvement for Vehicle 1 is less using an NKF. For Vehicles 2 and 3, the localization by NKF is even worse than the single vehicle localization (SKF).

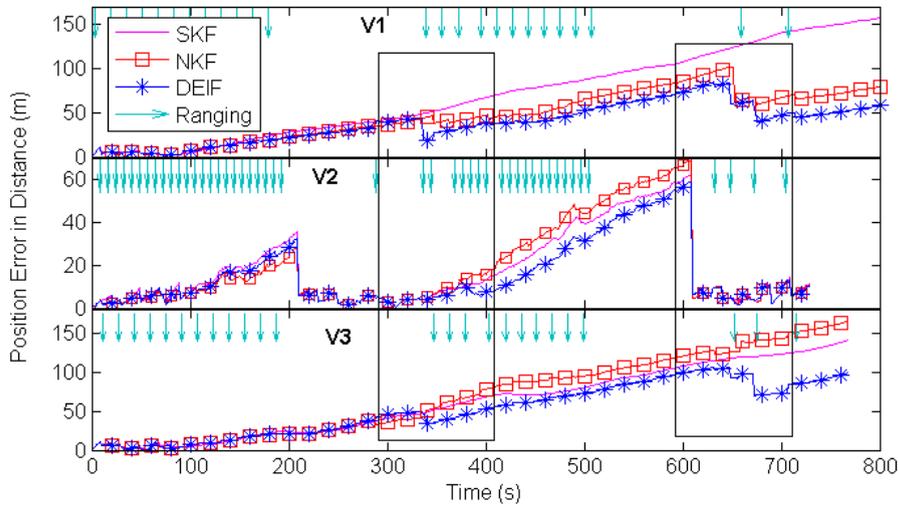


FIGURE 3.13: Cooperative localization results with field data. The arrows indicate the time when the broadcasts are sent and successfully received by other vehicles. NKF estimation sometimes is worse than SKF. DEIF improves estimation accuracy of all vehicles.

3.5 Summary

Although underwater communication is difficult, natural behaviors such as fish schooling have shown that limited communication still helps improve the overall performance. We have shown that cooperation under constrained communication make the distributed localization outperform the single-vehicle localization. However, the estimation may degenerate when the packet loss is beyond some point. This is due to the information double counting when the correlation is ignored. We also showed that distributed processing has some information loss even the correlation is exactly known. The reasons of these two will be explored in the next chapter.

With the challenges imposed by the underwater communication, we

reported the design and implementation of a distributed extended information filter for cooperative multi-AUV localization. This DEIF is especially suitable for underwater vehicles where the communication links have the problems of limited bandwidth and lossy packets.

Multi-vehicle cooperative localization is essentially a type of data fusion in cooperative intelligent vehicles. Data fusion, which aims at integration of data and knowledge from multiple sources, is an important process to achieve a better estimation in various applications. The proposed filter can also be used for cooperative object tracking, cooperative environment sensing or map building.

When vehicles have precise local measurements and/or infrequent communications, the inter-vehicle correlation may become trivial. In such a case, ignoring correlation in fusion might be able to work. We will discuss this in the next chapter.

Chapter 4

When Can One Ignore the Correlation?

4.1 Problem Statement

In the previous chapter we offer a distributed localization method dealing with unknown correlation due to lossy underwater communication. However, a *naïve* assumption which simply ignores the correlation when fusing data from different sources, was adopted in [80, 82]. It is simple in complex cooperation applications and seems to be working fine in existing works. We want to understand when this assumption is reasonable, and when is detrimental.

There are two types of cooperative localization problems: the multi-sensor tracking problem and the multi-vehicle localization problem. The multi-sensor tracking problem is concerned with a team of cooperative nodes tracking a common target, whereas the multi-vehicle localization problem deals with a team of cooperative vehicles estimating their own positions. In both problems, individual nodes or vehicles have no idea about the whole team; they also do not share every detailed local observation. Information is shared across the team and estimation is improved over the ones with single sensor tracking or single vehicle localization (without cooperation). We illustrate the idea by exploring the information flow in the decentralized system. We quantify the situation where the correlation can be safely ignored. We would like to ensure the estimation consistency when bathymetry information is incorporated into cooperative localization.

The work in this chapter was published in [70].

4.2 Multi-Sensor Tracking Problem

We present a simple example where two sensor nodes are tracking a common target. The central filter (CF) and single filter (SF) are used as the performance benchmarks. We answer the following questions:

- What is the optimal distributed filtering?

- Is there any gap between the optimal distributed filtering and the central filtering? If so, what is the gap?
- When can one ignore the correlation in fusion? When does the *naïve* assumption in filtering fail?

These answers give us a clear understanding about the pros and cons of implementing *naïve* filtering (NF) in the distributed localization.

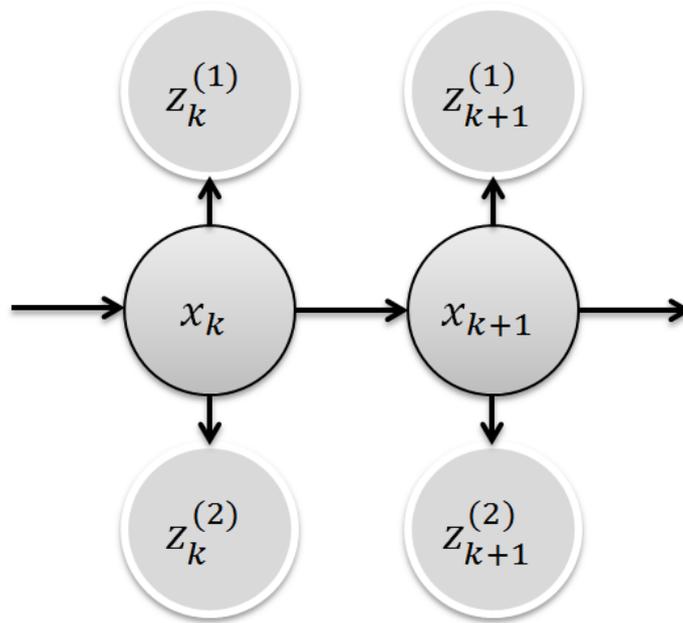


FIGURE 4.1: Multi-sensor tracking: a recursive two-step flow chart. The target propagates from previous position \mathbf{x}_k to current position \mathbf{x}_{k+1} with some propagation noise ω_k . At each step, each node makes an observation ($\mathbf{z}^{(1)}$ or $\mathbf{z}^{(2)}$) about the target position.

Figure 4.1 shows a recursive two-step process where two sensor nodes (node 1 and 2) track a target with position state \mathbf{x} of size $n \times 1$. The target propagates from previous position \mathbf{x}_k to current position \mathbf{x}_{k+1} with some

propagation noise ω_k . At each step, each node makes an observation ($\mathbf{z}^{(1)}$ or $\mathbf{z}^{(2)}$) about the target position. The propagation model and observation models are

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{x}_k + \omega_k \\ \mathbf{z}_k^{(1)} &= \mathbf{x}_k + \nu_k^{(1)} \\ \mathbf{z}_k^{(2)} &= \mathbf{x}_k + \nu_k^{(2)}\end{aligned}\tag{4.1}$$

where the propagation noise ω and observation noises $\nu^{(1)}$ and $\nu^{(2)}$ at all steps are independent zero-mean Gaussian processes with covariances \mathbf{Q} , $\mathbf{R}^{(1)}$ and $\mathbf{R}^{(2)}$. Without loss of generality, we assume $\det \mathbf{R}^{(1)} \leq \det \mathbf{R}^{(2)}$. The problem is to find the best estimate \mathbf{y}_{k+1} about the target position \mathbf{x}_{k+1} . We assess the filters' one-step and asymptotic performances. With the knowledge about target position at the previous step \mathbf{x}_k , one-step performance is the filter performance at the next step. When filters continue to be implemented over many steps, the estimation reaches a stable state where asymptotic performance can be derived.

Assuming a central unit with access to the local sensor data in real time, the central filtering (CF) simply stacks the local observations made at two nodes and follows the Kalman filtering to update the predicted estimate about the target. Similarly, single filtering (SF) follows Kalman filtering and uses the sensor data at node 1 only without cooperation from the other node.

Let the error covariance of the position estimate about state \mathbf{x}_k be \mathbf{P}_k . The estimate about the state \mathbf{x}_k is $\mathbf{y}_k = \mathbb{E}[\mathbf{x}_k]$. The one-step SF gives estimate with error covariance

$$\mathbf{P}_{\text{SF}} = (\mathbf{I} - (\mathbf{P}_k + \mathbf{Q})(\mathbf{P}_k + \mathbf{Q} + \mathbf{R}^{(1)})^{-1})(\mathbf{P}_k + \mathbf{Q}) \quad (4.2)$$

and one-step CF gives

$$\mathbf{P}_{\text{CF}} = (\mathbf{I} - (\mathbf{P}_k + \mathbf{Q})\mathbf{H}_{\text{CF}}\mathbf{S}_{\text{CF}}^{-1})(\mathbf{P}_k + \mathbf{Q}) \quad (4.3)$$

where

$$\mathbf{S}_{\text{CF}} = \mathbf{H}(\mathbf{P}_k + \mathbf{Q})\mathbf{H}_{\text{CF}}^{\top} + \begin{bmatrix} \mathbf{R}^{(1)} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^{(2)} \end{bmatrix} \quad (4.4)$$

$$\mathbf{H}_{\text{CF}} = \begin{bmatrix} \mathbf{I} \\ \mathbf{I} \end{bmatrix}$$

and \mathbf{I} is the identity matrix of size $n \times n$.

When the target keeps moving and observations are made at every step at the two nodes with the same settings, the filters reach stable states in which the estimation and performance approach constant values. The stable state estimation error covariances for both filters are derived by setting $\mathbf{P}_{\text{SF}} = \mathbf{P}_k$ and $\mathbf{P}_{\text{CF}} = \mathbf{P}_k$. In 1-dimensional space where $n = 1$,

they are

$$\begin{aligned} \mathbf{P}_{\text{SF,ss}} &= \frac{-\mathbf{Q} + \sqrt{\mathbf{Q}^2 + 4\mathbf{Q}\mathbf{R}^{(1)}}}{2} \\ \mathbf{P}_{\text{CF,ss}} &= \frac{-\mathbf{Q} + \sqrt{\frac{\mathbf{Q}^2(\mathbf{R}^{(1)} + \mathbf{R}^{(2)}) + 4\mathbf{Q}\mathbf{R}^{(1)}\mathbf{R}^{(2)}}{\mathbf{R}^{(1)} + \mathbf{R}^{(2)}}}}{2}. \end{aligned} \quad (4.5)$$

The CF result is similar to SF but at each time the observation data is fused by the two sensor data. This ‘fused’ observation therefore has error covariance $(\mathbf{R}^{(1)-1} + \mathbf{R}^{(2)-1})^{-1}$.

It should be noted that the estimates by CF and SF are consistent in that the actual error covariance of the estimate $\mathbb{E}[(\mathbf{y} - \mathbf{x})(\mathbf{y} - \mathbf{x})^\top]$ equals the estimated error covariance \mathbf{P} . Therefore we only state one of them here.

The performances of CF and SF are shown in the subsequent sections.

4.2.1 Optimal Distributed Filtering

4.2.1.1 Deriving the Optimal DF

The recursive two-step distributed filtering is shown in Figure 4.2. At the previous step k , sensor nodes exchange their local estimates and obtain a fused estimate $\mathbf{y}_k^{(f)} = \mathbf{y}_k$ with error covariance $\mathbf{P}_k^{(f)} = \mathbf{P}_k$. This fused estimate about the previous position \mathbf{x}_k is adopted by both nodes. After that, the target position is predicted at each node locally, and updated with local sensor data following the standard Kalman filtering method. At

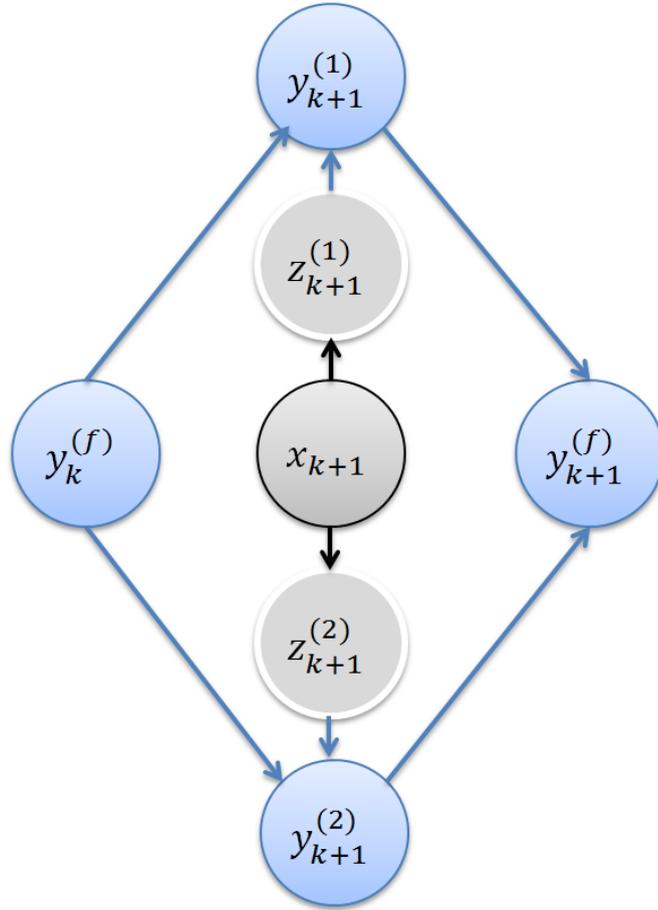


FIGURE 4.2: Multi-sensor tracking: distributed filtering using weighted-sum fusion.

the current step $k + 1$, again sensor nodes exchange their local estimates $\mathbf{y}_{k+1}^{(1)}$ and $\mathbf{y}_{k+1}^{(2)}$ and obtain the fused estimate $\mathbf{y}_{k+1}^{(f)}$. We use weighted sum to fuse the two estimates with weight λ

$$\begin{aligned} \mathbf{y}_{\text{DF}}^{(f)} &= f(\mathbf{y}_{k+1}^{(1)}, \mathbf{y}_{k+1}^{(2)}) \\ &= (1 - \lambda)\mathbf{y}_{k+1}^{(1)} + \lambda\mathbf{y}_{k+1}^{(2)}. \end{aligned} \tag{4.6}$$

With the fused estimate $\mathbf{y}_k^{(f)}$ and estimated error covariance $\mathbf{P}_k^{(f)}$, the local estimates are for nodes $i = 1, 2$

$$\begin{aligned}\mathbf{y}^{(i)} &= \mathbf{P}^{(i)}(\mathbf{P}_k^{(f)-1}\mathbf{y}_k^{(f)} + \mathbf{R}^{(i)-1}\mathbf{z}_{k+1}^{(i)}) \\ \mathbf{P}^{(i)} &= (\mathbf{R}^{(i)-1} + \mathbf{P}_k^{(f)-1})^{-1}\end{aligned}\quad (4.7)$$

The optimal weights λ^* is derived by minimizing the error covariance $\mathbb{E}[(\mathbf{y}_{\text{DF}}^{(f)} - \mathbf{x}_{k+1})(\mathbf{y}_{\text{DF}}^{(f)} - \mathbf{x}_{k+1})^\top]$ and in 1-dimensional space we obtain

$$\begin{aligned}\lambda^* &= \arg \min \mathbb{E}[(\mathbf{y}_{\text{DF}}^{(f)} - \mathbf{x}_{k+1})^2] \\ &= \frac{\mathbf{R}^{(1)}}{\mathbf{R}^{(1)} + \mathbf{R}^{(2)}}.\end{aligned}\quad (4.8)$$

With this optimal weight, the error covariance $\mathbb{E}[(\mathbf{y}_{\text{DF}}^{(f)} - \mathbf{x}_{k+1})(\mathbf{y}_{\text{DF}}^{(f)} - \mathbf{x}_{k+1})^\top]$ of the fused estimate turns out to be identical with Bar Shalom's state vector fusion (SVF) [13]

$$\mathbf{P}_{\text{SVF}}^{(f)} = \mathbf{P}_{k+1}^{(1)} - (\mathbf{P}_{k+1}^{(1)} - \mathbf{P}_{k+1}^{(12)})(\mathbf{P}_{k+1}^{(1)} + \mathbf{P}_{k+1}^{(2)} - \mathbf{P}_{k+1}^{(12)} - \mathbf{P}_{k+1}^{(12)\top})^{-1}(\mathbf{P}_{k+1}^{(1)} - \mathbf{P}_{k+1}^{(12)\top}) \quad (4.9)$$

where the cross-correlation $\mathbf{P}_{k+1}^{(12)} = \mathbb{E}[(\mathbf{y}_{k+1}^{(1)} - \mathbf{x}_{k+1})(\mathbf{y}_{k+1}^{(2)} - \mathbf{x}_{k+1})^\top]$ between the two estimates is required. Although $\mathbf{P}_{k+1}^{(12)}$ is not required for optimal fusion of the estimate in DF, it is required for a consistent estimation on the error covariance $\mathbf{P}_{\text{DF}}^{(f)*}$ which can be used for the subsequent steps. The stable state error covariance of optimal DF can also be derived by setting

$$\mathbf{P}_{\text{DF}}^{(f)*} = \mathbf{P}_k.$$

4.2.1.2 The Gap between the Optimal DF and CF

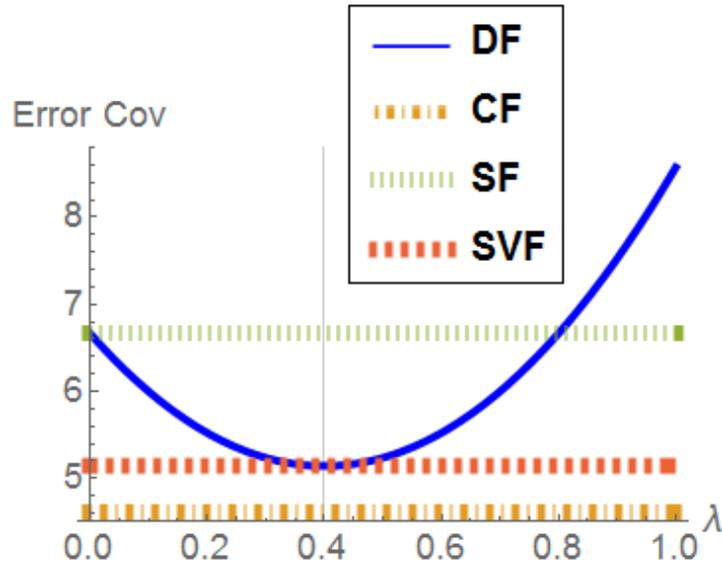


FIGURE 4.3: One-step error covariances of DF, SF, CF and SVF, against the weight λ used by DF. There is a gap between CF and optimal DF (SVF).

Figure 4.3 shows an example of the one-step performance comparing the error covariances of the estimates by DF, SF, CF and SVF, against the weight λ used by DF. The optimal weight λ^* is obtained at the lowest point of the DF curve, which gives identical performance as SVF. There is a gap between the optimal DF (or SVF) and CF. The same idea was stated by Bar-Shalom in [11]

The sufficient statistics for the global data set $D^{ij} = D^i \cup D^j$ cannot be expressed in terms of the sufficient statistics of the local data sets D^i and D^j (the local estimates \hat{x}^i and \hat{x}^j).

We have discussed the similarity information loss in distributed localization in Chapter 3. There is information loss when processed data (estimates) are transmitted instead of raw data (observations). This is because the CF is a Maximum a Posteriori (MAP) estimation which estimates the unobserved target position state \mathbf{x} with two observations $\mathbf{z}^{(1)}$ and $\mathbf{z}^{(2)}$. The prior distribution is known as $\mathcal{N}(\mathbf{x}, \mathbf{P}_k + \mathbf{Q})$. In the other hand, DF treats the two local estimates $\mathbf{y}_{k+1}^{(1)}$ and $\mathbf{y}_{k+1}^{(2)}$ as two ‘observations’. The fusion is a Maximum Likelihood (ML) estimation without using the prior knowledge about the state [23]. This is the best that distributed estimation can achieve when only local estimates are available for fusion. Therefore it is optimal only in the ML context. We take SVF as an example, since it is identical to the optimal DF. Let the error covariance of the two estimates from the nodes be

$$\mathbf{P}_{\text{SVF}} = \begin{bmatrix} \mathbf{P}_{k+1}^{(1)} & \mathbf{P}_{k+1}^{(12)} \\ \mathbf{P}_{k+1}^{(12)\top} & \mathbf{P}_{k+1}^{(2)} \end{bmatrix} \quad (4.10)$$

where

$$\begin{aligned}
\mathbf{P}_1 &= \mathbb{E}[(\mathbf{y}_{k+1}^{(1)} - \mathbf{x}_{k+1})(\mathbf{y}_{k+1}^{(1)} - \mathbf{x}_{k+1})^\top] \\
\mathbf{P}_2 &= \mathbb{E}[(\mathbf{y}_{k+1}^{(2)} - \mathbf{x}_{k+1})(\mathbf{y}_{k+1}^{(2)} - \mathbf{x}_{k+1})^\top] \\
\mathbf{P}_{k+1}^{(12)} &= \mathbb{E}[(\mathbf{y}_{k+1}^{(1)} - \mathbf{x}_{k+1})(\mathbf{y}_{k+1}^{(2)} - \mathbf{x}_{k+1})^\top]
\end{aligned} \tag{4.11}$$

The target position \mathbf{x}_{k+1} is to be determined based on the two ‘observations’ $\mathbf{y}_{k+1}^{(1)}$ and $\mathbf{y}_{k+1}^{(2)}$. The logarithm of the likelihood function $\mathcal{L}(\mathbf{x}_{k+1}; \mathbf{y}_{k+1}^{(1)}, \mathbf{y}_{k+1}^{(2)})$ is

$$\begin{aligned}
&\ln \mathcal{L}(\mathbf{x}_{k+1}; \mathbf{y}_{k+1}^{(1)}, \mathbf{y}_{k+1}^{(2)}) \\
&= \ln 2\pi - \frac{1}{2} \ln \left(\det(\mathbf{P}_{\text{SVF}}) \right. \\
&\quad \left. + \left(\begin{bmatrix} \mathbf{y}_{k+1}^{(1)} \\ \mathbf{y}_{k+1}^{(2)} \end{bmatrix} - \begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{x}_{k+1} \end{bmatrix} \right)^\top \mathbf{P}_{\text{SVF}}^{-1} \left(\begin{bmatrix} \mathbf{y}_{k+1}^{(1)} \\ \mathbf{y}_{k+1}^{(2)} \end{bmatrix} - \begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{x}_{k+1} \end{bmatrix} \right) \right).
\end{aligned} \tag{4.12}$$

It is maximized by setting $\frac{\partial \mathcal{L}(\mathbf{x}_{k+1}; \mathbf{y}_{k+1}^{(1)}, \mathbf{y}_{k+1}^{(2)})}{\partial \mathbf{x}} = 0$. The optimal estimate for position state \mathbf{x} in 1-dimensional space is

$$\begin{aligned}
\mathbf{y}_{\text{ML}}^* &= \arg \max \mathcal{L}(\mathbf{x}_{k+1}; \mathbf{y}_{k+1}^{(1)}, \mathbf{y}_{k+1}^{(2)}) \\
&= \frac{\mathbf{P}_{k+1}^{(2)} \mathbf{y}_{k+1}^{(1)} + \mathbf{P}_{k+1}^{(1)} \mathbf{y}_{k+1}^{(2)} - \mathbf{P}_{k+1}^{(12)} (\mathbf{y}_{k+1}^{(1)} + \mathbf{y}_{k+1}^{(2)})}{\mathbf{P}_{k+1}^{(1)} + \mathbf{P}_{k+1}^{(2)} - 2\mathbf{P}_{k+1}^{(12)}}.
\end{aligned} \tag{4.13}$$

The volume ratio of the error covariances compared with CF is

$$\frac{\det \mathbf{P}_{\text{SVF}}^{(f)}}{\det \mathbf{P}_{\text{CF}}} = \frac{\det \mathbf{P}_{\text{DF}}^{(f)*}}{\det \mathbf{P}_{\text{CF}}}. \tag{4.14}$$

Figure 4.4 shows that the ratio is always less than one for $n = 1$. The axes are $\mathbf{R}^{(1)}$ and $\mathbf{R}^{(2)}$ normalized by $\mathbf{P}_k + \mathbf{Q}$.

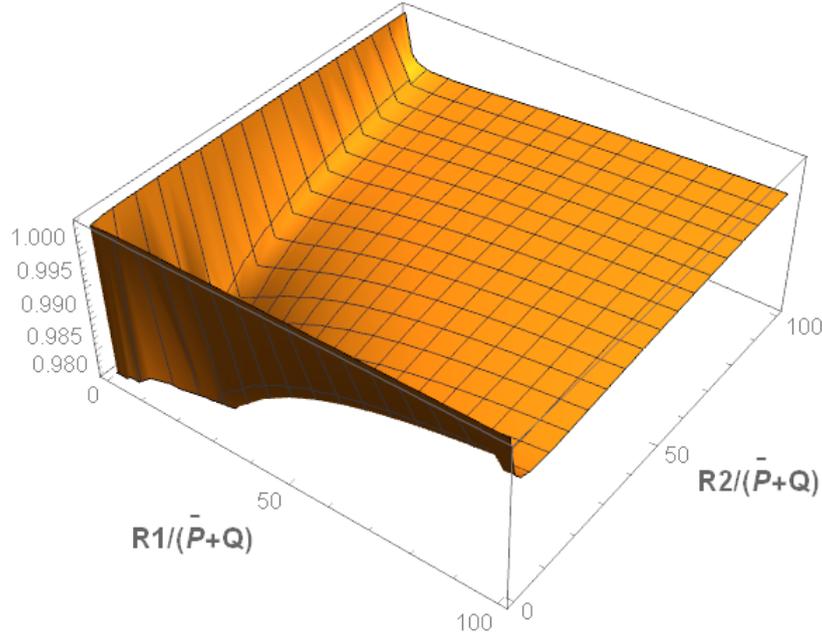


FIGURE 4.4: Ratio of error covariances - optimal DF (SVF) to CF is no larger than one.

4.2.2 When Can One Ignore the Correlation?

The *naïve* filter (NF) simply fuses the two estimates assuming no correlation between them. The procedure is similar to DF in Figure 4.2 but the weight of fusion is calculated by the estimated error covariances, that is, $\lambda_{\text{NF}} = \mathbf{P}_{k+1}^{(1)} \mathbf{P}^{(f)-1}$. The fused estimate is

$$\begin{aligned} \mathbf{y}_{\text{NF}}^{(f)} &= (\mathbf{I} - \lambda_{\text{NF}}) \mathbf{y}_{k+1}^{(1)} + \lambda_{\text{NF}} \mathbf{y}_{k+1}^{(2)} \\ \mathbf{P}_{\text{NF}}^{(f)} &= (\mathbf{P}_{k+1}^{(1)-1} + \mathbf{P}_{k+1}^{(2)-1})^{-1}. \end{aligned} \tag{4.15}$$

The resulting error covariance of the fused estimate is $\mathbb{E}[(\mathbf{y}_{\text{NF}}^{(f)} - \mathbf{x}_{k+1})(\mathbf{y}_{\text{NF}}^{(f)} - \mathbf{x}_{k+1})^\top]$.

By ignoring the correlation, the common information from the two estimates is double counted in the fusion, which leads to estimation overconfidence. The estimated error covariance $\mathbf{P}_{\text{NF}}^{(f)}$ is smaller than the actual error covariance, that is $\det(\mathbf{P}_{\text{NF}}^{(f)}) < \det \mathbb{E}[(\mathbf{y}_{\text{NF}}^{(f)} - \mathbf{x}_{k+1})(\mathbf{y}_{\text{NF}}^{(f)} - \mathbf{x}_{k+1})^\top]$. In fact, the estimated error covariance is even smaller than the estimated covariance by CF. The overconfidence prevents utilization of subsequent useful information and therefore the actual error covariance of NF estimate is sometimes even worse than that of SF. In such a case, cooperation using NF is no longer advantageous. We call it the ‘*dangerous*’ region. *Naïve* assumption fails here, that is when

$$\det \mathbb{E}[(\mathbf{y}_{\text{NF}}^{(f)} - \mathbf{x}_{k+1})(\mathbf{y}_{\text{NF}}^{(f)} - \mathbf{x}_{k+1})^\top] > \det \mathbf{P}_{\text{SF}}. \quad (4.16)$$

When the above inequality is not met, we consider that it is safe to use NF and one can ignore the correlation.

4.2.2.1 One-step Performance

We calculate the one-step *dangerous* region when the inequality in Equation (4.16) is met. For $n = 1$ the *dangerous* region is derived as

$$\begin{aligned} \bar{\mathbf{R}}^{(2)} &> 1 \\ \frac{\bar{\mathbf{R}}^{(1)}}{\bar{\mathbf{R}}^{(2)}} + \bar{\mathbf{R}}^{(2)} + 3\bar{\mathbf{R}}^{(1)}\bar{\mathbf{R}}^{(2)} &\leq \bar{\mathbf{R}}^{(2)} \end{aligned} \quad (4.17)$$

where $\bar{\mathbf{R}}^{(1)}$ and $\bar{\mathbf{R}}^{(2)}$ are the normalized error covariances and $\bar{\mathbf{R}}^{(i)} = \bar{\mathbf{R}}^{(i)}/(\mathbf{P}_k + \mathbf{Q}), i = 1, 2$. It is plotted in Figure 4.5. We can see that it is safe to use the one-step NF only if both local observations have very small errors compared with the sum of the estimation error and propagation error, or when the two local sensors have comparable observation errors.

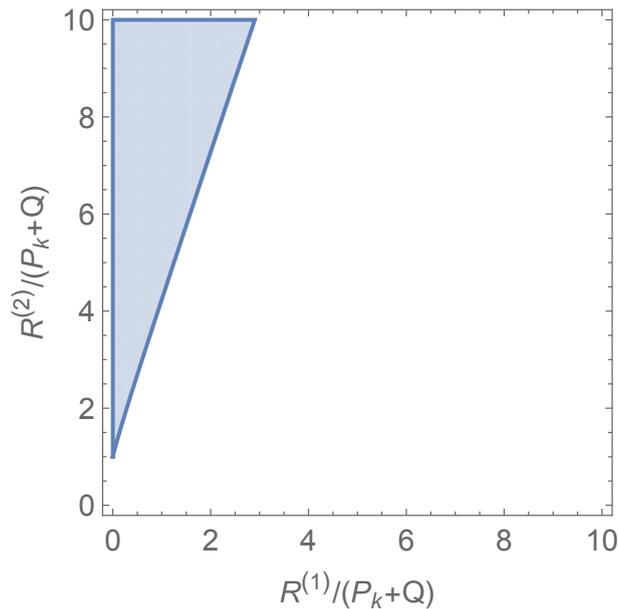
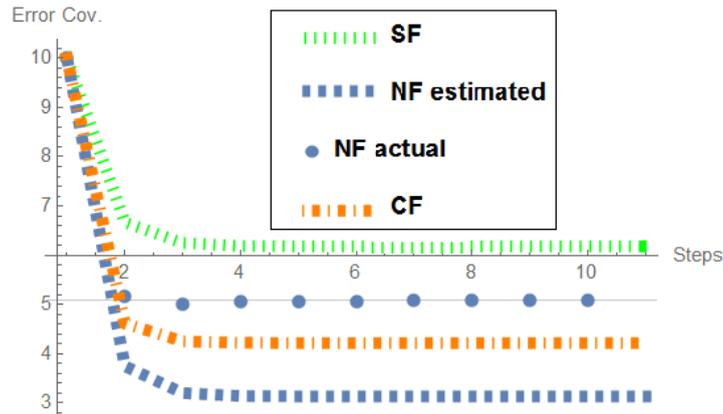
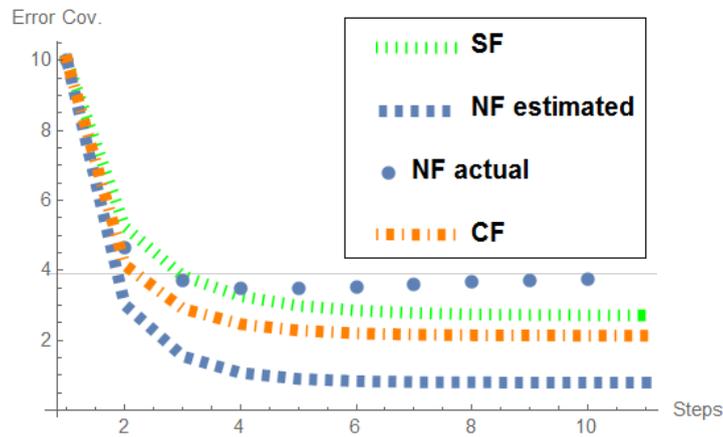


FIGURE 4.5: One-step performance: the *dangerous* region of implementing NF (assuming $\mathbf{R}^{(1)} < \mathbf{R}^{(2)}$).

4.2.2.2 Asymptotic Performance



(a)



(b)

FIGURE 4.6: *Naïve* filter in stable state: (a) Actual error covariance is smaller than SF. (b) Actual error covariance eventually gets worse than SF. In both cases, NF is overconfident about its estimation. The estimated error covariance by NF is even lower than the one by CF.

After the first step, if we continue to ignore the correlation and fuse the estimates with weights calculated from the estimated error covariances, the actual estimation error is likely to diverge and degrade further. Figure 4.6 shows two cases in stable state. The blue dots are the actual (by Monte Carlo simulation) error covariances of the NF estimates. The horizontal

line is the calculated asymptotic error covariances for NF estimates. The blue dashed line shows the estimated error covariances given by NF. We can see that NF is overconfident about its estimates. In Case (b), even though in the first step (Step 2) NF gives improvement over SF, its estimation error becomes larger than SF from Step 4 onward.

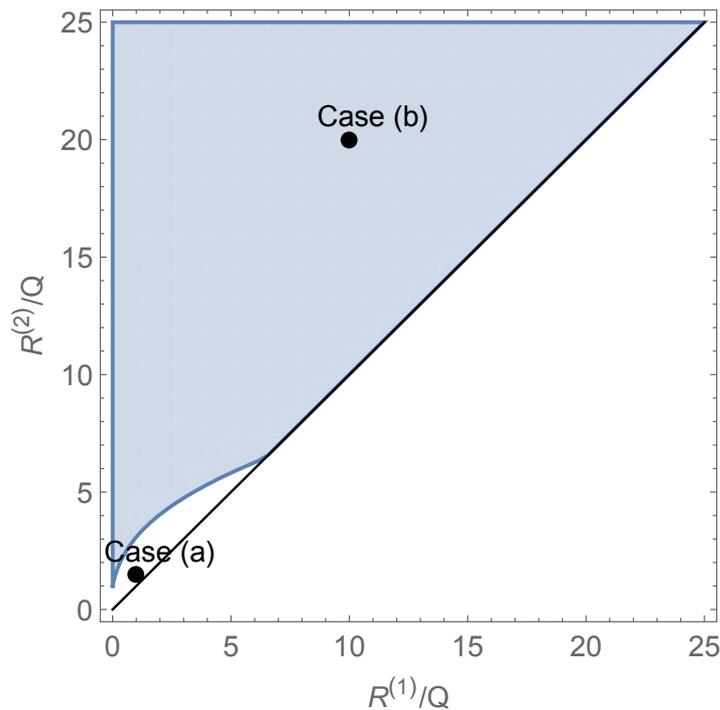


FIGURE 4.7: Asymptotic performance: the *dangerous* region of implementing NF (assuming $\mathbf{R}^{(1)} \leq \mathbf{R}^{(2)}$).

We calculate the stable state error covariances of NF estimates. Compared with stable state SF error covariances, the asymptotic *dangerous* region is plotted in Figure 4.7. It should be noted that the asymptotic performance does not depend on initial error covariance, and therefore the axes are $\mathbf{R}^{(1)}$ and $\mathbf{R}^{(2)}$ normalized by the process noise error covariance \mathbf{Q} . The two cases in Figure 4.6 are located in the plot. We can see that if the

two local sensors have small measurement errors (compared with process noise), one can ignore the correlation.

4.2.3 Summary

In this section, we examined the performances of distributed processing in underwater multi-sensor tracking problem. We showed that the optimal distributed filter is achieved only when correlation is exactly tracked. However, there is still information loss due to transmission of the processed data instead of the raw data. We also showed the consequence of ignoring the correlation - the estimation overconfidence and estimation divergence. The actual estimation is worse than it is believed to be in the filter, and could be worse than the single filter without cooperation. We derived the *dangerous* regions of implementing *naïve* filter. This can be used as a guideline for conditions under which one can ignore the correlation for underwater cooperative tracking.

4.3 Multi-Vehicle Localization

Multi-vehicle localization can be viewed as cooperative network where each node estimates its own state. Different from multi-sensor tracking problem, there are many states to be estimated (one for each node), and there is a

relative measurement relating the states of the cooperating nodes. Figure 4.8 shows the Bayesian network of two cooperating vehicles (Vehicle i and Vehicle j). The unobserved position state variables are $\mathbf{x}_k^{(i)}$ and $\mathbf{x}_k^{(j)}$. The observed variables are the observations $\mathbf{z}_k^{(i)}$ or $\mathbf{z}_k^{(j)}$ made at Vehicle i or j locally about their positions and relative distance r_k between them. $k = 0, 1, 2, \dots$ denotes the discrete time step.

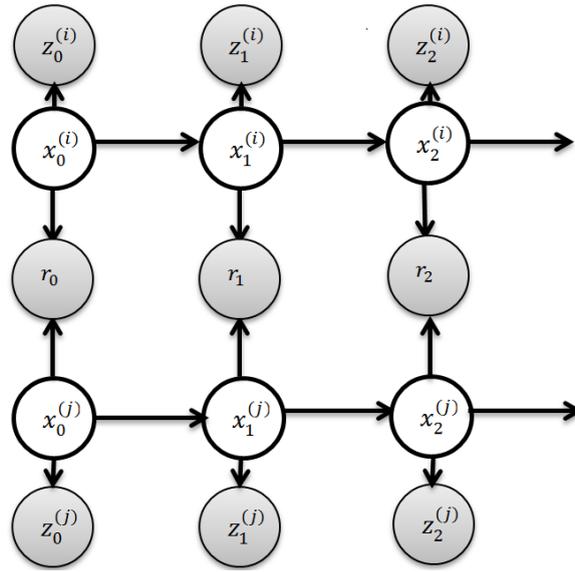


FIGURE 4.8: Bayesian network for cooperative localization of two vehicles: shaded nodes are observations and white nodes are unobserved position state variables.

The position of Vehicle i (so as Vehicle j) propagates as

$$\begin{aligned}
 \mathbf{x}_{k+1}^{(i)} &= f(\mathbf{x}_k^{(i)}) + \omega_k^{(i)} \\
 &= \mathbf{x}_k^{(i)} + \mathbf{v}_k^{(i)} + \omega_k^{(i)}
 \end{aligned} \tag{4.18}$$

where \mathbf{v} is the known speed. The observations are

$$\begin{aligned}
 \mathbf{z}_k^{(i)} &= h(\mathbf{x}_k^{(i)}) + \nu_k^{(i)} \\
 &= \mathbf{x}_k^{(i)} + \nu_k^{(i)} \\
 \mathbf{z}_k^{(j)} &= h(\mathbf{x}_k^{(j)}) + \nu_k^{(j)} \\
 &= \mathbf{x}_k^{(j)} + \nu_k^{(j)} \\
 r_k &= \mathbf{x}_k^{(i)} - \mathbf{x}_k^{(j)} + v_k
 \end{aligned} \tag{4.19}$$

4.3.1 Information Flow when Ignoring Correlation

Assuming all the noises and priors are Gaussian distributed, the joint Gaussian distribution of state \mathbf{x} and observations \mathbf{z} is given by

$$\text{Prob}\left(\begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix}\right) = \alpha e^{\frac{1}{2}} \left(\begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix} - \mu\right)^\top \mathbf{\Lambda} \left(\begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix} - \mu\right) \tag{4.20}$$

where $\mu = \begin{bmatrix} \mu_{\mathbf{x}} \\ \mu_{\mathbf{z}} \end{bmatrix} = \begin{bmatrix} \mathbb{E}(\mathbf{x}) \\ \mathbb{E}(\mathbf{z}) \end{bmatrix}$ and $\mathbf{\Lambda}$ is the precision matrix (or inverse covariance matrix). $\mathbf{\Lambda}$ has the following blockwise structure:

$$\mathbf{\Lambda} = \begin{bmatrix} \mathbf{\Lambda}_{\mathbf{xx}} & \mathbf{\Lambda}_{\mathbf{xz}} \\ \mathbf{\Lambda}_{\mathbf{zx}} & \mathbf{\Lambda}_{\mathbf{zz}} \end{bmatrix} = \mathbf{P}^{-1} \tag{4.21}$$

where \mathbf{P} is the covariance matrix for $\begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix}$ and $\mathbf{P} = \begin{bmatrix} \mathbf{P}_{\mathbf{xx}} & \mathbf{P}_{\mathbf{xz}} \\ \mathbf{P}_{\mathbf{zx}} & \mathbf{P}_{\mathbf{zz}} \end{bmatrix}$. It should be noted that the subscripts of the subblocks $\Lambda_{\mathbf{xx}}$, $\Lambda_{\mathbf{xz}} = \Lambda_{\mathbf{zx}}^\top$ and $\Lambda_{\mathbf{zz}}$ denote the sizes corresponding to \mathbf{x} and \mathbf{z} . $\Lambda_{\mathbf{xx}}^{-1}$ is the conditional covariance of \mathbf{x} given all the observations \mathbf{z} . It describes the covariance of the Gaussian probability $p(\mathbf{x}|\mathbf{z})$. We define the conditional covariance as

$$\begin{aligned} \mathbf{C}_{\mathbf{x}|\mathbf{z}} &= \Lambda_{\mathbf{xx}}^{-1} \\ &= \mathbf{P}_{\mathbf{xx}} - \mathbf{P}_{\mathbf{xz}}\mathbf{P}_{\mathbf{zz}}^{-1}\mathbf{P}_{\mathbf{zx}}. \end{aligned} \tag{4.22}$$

This is also the result of block matrix inversion. We also have

$$\mathbf{P}_{\mathbf{xx}}^{-1} = \Lambda_{\mathbf{xx}} - \Lambda_{\mathbf{xz}}\Lambda_{\mathbf{zz}}^{-1}\Lambda_{\mathbf{xz}}^\top. \tag{4.23}$$

When \mathbf{z} are linear observations about \mathbf{x} , $\mathbf{z} = \mathbf{H}\mathbf{x} + \nu$, we have $\mathbf{z} \sim \mathcal{N}(\mathbf{H}\mathbf{x}, \mathbf{R})$ and

$$\begin{aligned}
\mathbf{P}_{\mathbf{zx}} &= \mathbf{P}_{\mathbf{zx}}^\top = \mathbf{P}_{\mathbf{xx}}\mathbf{H}^\top \\
\mathbf{P}_{\mathbf{zz}} &= \mathbf{H}\mathbf{P}_{\mathbf{xx}}\mathbf{H}^\top + \mathbf{R} \\
\mathbf{\Lambda}_{\mathbf{xx}} &= \mathbf{H}^\top\mathbf{R}^{-1}\mathbf{H} + \mathbf{P}_{\mathbf{xx}}^{-1} \\
\mathbf{\Lambda}_{\mathbf{zx}} &= -\mathbf{\Lambda}_{\mathbf{xx}}\mathbf{P}_{\mathbf{zx}}\mathbf{P}_{\mathbf{zz}}^{-1} \\
&= -(\mathbf{H}^\top\mathbf{R}^{-1}\mathbf{H} + \mathbf{P}_{\mathbf{xx}}^{-1})\mathbf{P}_{\mathbf{xx}}\mathbf{H}^\top(\mathbf{H}\mathbf{P}_{\mathbf{xx}}\mathbf{H}^\top + \mathbf{R})^{-1} \\
&= -(\mathbf{H}^\top\mathbf{R}^{-1}\mathbf{H}\mathbf{P}_{\mathbf{xx}}\mathbf{H}^\top + \mathbf{H}^\top)(\mathbf{H}\mathbf{P}_{\mathbf{xx}}\mathbf{H}^\top + \mathbf{R})^{-1} \\
&= -(\mathbf{H}^\top\mathbf{R}^{-1}\mathbf{H}\mathbf{P}_{\mathbf{xx}}\mathbf{H}^\top + \mathbf{H}^\top\mathbf{R}^{-1}\mathbf{R})(\mathbf{H}\mathbf{P}_{\mathbf{xx}}\mathbf{H}^\top + \mathbf{R})^{-1} \\
&= -\mathbf{H}^\top\mathbf{R}^{-1}(\mathbf{H}\mathbf{P}_{\mathbf{xx}}\mathbf{H}^\top + \mathbf{R})(\mathbf{H}\mathbf{P}_{\mathbf{xx}}\mathbf{H}^\top + \mathbf{R})^{-1} \\
&= -\mathbf{H}^\top\mathbf{R}^{-1} \\
\mathbf{\Lambda}_{\mathbf{zz}}^{-1} &= \mathbf{R}.
\end{aligned} \tag{4.24}$$

The probability distribution of \mathbf{x} given \mathbf{z} is given by Bayesian Theorem

$$\begin{aligned}
p(\mathbf{x}|\mathbf{z}) &= \frac{p(\mathbf{z}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{z})} \\
&= (2\pi)^{-\frac{n_{\mathbf{z}}}{2}}\mathbf{R}^{-\frac{1}{2}}\exp\left\{-\frac{1}{2}(\mathbf{z} - \mathbf{H}\mathbf{x})^\top\mathbf{R}^{-1}(\mathbf{z} - \mathbf{H}\mathbf{x})\right\} \\
&\times (2\pi)^{-\frac{n_{\mathbf{x}}}{2}}\mathbf{P}_{\mathbf{xx}}^{-\frac{1}{2}}\exp\left\{-\frac{1}{2}(\mathbf{x} - \mu_{\mathbf{x}})^\top\mathbf{P}_{\mathbf{xx}}^{-1}(\mathbf{x} - \mu_{\mathbf{x}})\right\} \\
&\times (2\pi)^{\frac{n_{\mathbf{z}}}{2}}\mathbf{P}_{\mathbf{zz}}^{\frac{1}{2}}\exp\left\{\frac{1}{2}(\mathbf{z} - \mu_{\mathbf{z}})^\top\mathbf{P}_{\mathbf{zz}}^{-1}(\mathbf{z} - \mu_{\mathbf{z}})\right\}.
\end{aligned} \tag{4.25}$$

We examine the exponents and look for an estimated mean $\hat{\mathbf{x}}$ about \mathbf{x} given \mathbf{z} , such that $p(\mathbf{x}|\mathbf{z})$ is maximized. The maximum $\ln p(\mathbf{x}|\mathbf{z})$ is found

by taking the derivative $\frac{\partial \ln p(\mathbf{x}|\mathbf{z})}{\partial \mathbf{x}} = 0$. Therefore in Equation (4.25) we are only interested in the exponent terms involving \mathbf{x} . We are now left with

$$\begin{aligned}
& -\frac{1}{2}(\mathbf{z} - \mathbf{H}\mathbf{x})^\top \mathbf{R}^{-1}(\mathbf{z} - \mathbf{H}\mathbf{x}) - \frac{1}{2}(\mathbf{x} - \mu_{\mathbf{x}})^\top \mathbf{P}_{\mathbf{xx}}^{-1}(\mathbf{x} - \mu_{\mathbf{x}}) \\
& = -\frac{1}{2}[\mathbf{z}^\top \mathbf{R}^{-1}\mathbf{z} + \mathbf{x}^\top \mathbf{H}^\top \mathbf{R}^{-1}\mathbf{H}\mathbf{x} - \mathbf{z}^\top \mathbf{R}^{-1}\mathbf{H}\mathbf{x} - \mathbf{x}^\top \mathbf{H}^\top \mathbf{R}^{-1}\mathbf{z} \\
& \quad + \mathbf{x}^\top \mathbf{P}_{\mathbf{xx}}^{-1}\mathbf{x} - \mathbf{x}^\top \mathbf{P}_{\mathbf{xx}}^{-1}\mu_{\mathbf{x}} - \mu_{\mathbf{x}}^\top \mathbf{P}_{\mathbf{xx}}^{-1}\mathbf{x} + \mu_{\mathbf{x}}^\top \mathbf{P}_{\mathbf{xx}}^{-1}\mu_{\mathbf{x}}].
\end{aligned} \tag{4.26}$$

Again we drop the terms not involving \mathbf{x} . We group the similar terms with respect to \mathbf{x} and denote it as a function of \mathbf{x} , that is

$$\begin{aligned}
g(\mathbf{x}) & = -\frac{1}{2}[\mathbf{x}^\top (\mathbf{H}^\top \mathbf{R}^{-1}\mathbf{H} + \mathbf{P}_{\mathbf{xx}}^{-1})\mathbf{x} - (\mathbf{z}^\top \mathbf{R}^{-1}\mathbf{H} + \mu_{\mathbf{x}}^\top \mathbf{P}_{\mathbf{xx}}^{-1})\mathbf{x} - \mathbf{x}^\top (\mathbf{H}^\top \mathbf{R}^{-1}\mathbf{z} + \mathbf{P}_{\mathbf{xx}}^{-1}\mu_{\mathbf{x}})] \\
& = -\frac{1}{2}[\mathbf{x}^\top \mathbf{A}\mathbf{x} - \mathbf{b}^\top \mathbf{x} - \mathbf{x}^\top \mathbf{b}]
\end{aligned} \tag{4.27}$$

where $\mathbf{A} = \mathbf{H}^\top \mathbf{R}^{-1}\mathbf{H} + \mathbf{P}_{\mathbf{xx}}^{-1} = \mathbf{\Lambda}_{\mathbf{xx}}$ is symmetrical and $\mathbf{b} = \mathbf{H}^\top \mathbf{R}^{-1}\mathbf{z} + \mathbf{P}_{\mathbf{xx}}^{-1}\mu_{\mathbf{x}} = -\mathbf{\Lambda}_{\mathbf{xz}}\mathbf{z} + \mathbf{P}_{\mathbf{xx}}^{-1}\mu_{\mathbf{x}}$. We can denote $\ln p(\mathbf{x}|\mathbf{z}) = \mathcal{B} + g(\mathbf{x})$ where \mathcal{B}

groups all the terms not involving \mathbf{x} . The solution of $\hat{\mathbf{x}}$ is found such that

$$\begin{aligned}
\frac{\partial \ln p(\mathbf{x}|\mathbf{y})}{\partial \mathbf{x}} &= \frac{\partial (\mathcal{B} + f(\mathbf{x}))}{\partial \mathbf{x}} \\
&= \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \\
&= -\frac{1}{2} \left(\frac{\partial \mathbf{x}^\top \mathbf{A} \mathbf{x}}{\partial \mathbf{x}} - \frac{\partial \mathbf{b}^\top \mathbf{x}}{\partial \mathbf{x}} - \frac{\partial \mathbf{x}^\top \mathbf{b}}{\partial \mathbf{x}} \right) \\
&= -\frac{1}{2} [\mathbf{x}^\top (\mathbf{A} + \mathbf{A}^\top) - \mathbf{b}^\top - \mathbf{b}^\top] \\
&= -(\mathbf{x}^\top \mathbf{A} - \mathbf{b}^\top) \\
&= 0
\end{aligned} \tag{4.28}$$

Therefore we have $\mathbf{A}\hat{\mathbf{x}} = \mathbf{b}$. Substituting everything back, we have the exact inference about \mathbf{x} given all observations \mathbf{z} as a solution to:

$$\mathbf{\Lambda}_{\mathbf{xx}}\hat{\mathbf{x}} = \mathbf{P}_{\mathbf{xx}}^{-1}\mu_{\mathbf{x}} - \mathbf{\Lambda}_{\mathbf{xz}}\mathbf{z}. \tag{4.29}$$

The exact estimation can be obtained through Kalman filtering step by step. We convert the Bayesian network in Figure 4.8 into Markov net where only unobserved state variables are shown (Figure 4.9(c)).

A central filtering stacks the state variables and observations such that

$$\begin{aligned}
\mathbf{x} &= [\mathbf{x}_k^{(i)\top}, \mathbf{x}_k^{(j)\top}, \dots, \mathbf{x}_1^{(i)\top}, \mathbf{x}_1^{(j)\top}, \mathbf{x}_0^{(i)\top}, \mathbf{x}_0^{(j)\top}]^\top \\
\mathbf{z} &= [\mathbf{z}_k^{(i)\top}, \mathbf{z}_k^{(j)\top}, \mathbf{r}_k^\top, \dots, \mathbf{z}_1^{(i)\top}, \mathbf{z}_1^{(j)\top}, \mathbf{r}_1^\top, \mathbf{z}_0^{(i)\top}, \mathbf{z}_0^{(j)\top}, \mathbf{r}_0^\top]^\top
\end{aligned} \tag{4.30}$$

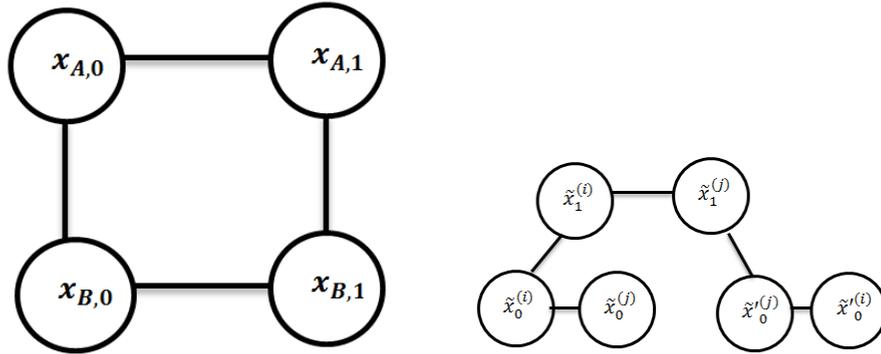
with variables at the last (current) step k as the first elements. We are

interested to know the estimated mean about state $\mathbf{x}_k^{(i)}$. In the case of 1-dimensional space, we define the estimated error covariance corresponding to $\mathbf{x}_k^{(i)}$ as σ^2 and $\sigma^2 = \mathbf{C}_{\mathbf{x}|\mathbf{z}}(1, 1)$. We define the error covariance of the estimated mean for state $\mathbf{x}_k^{(i)}$ as δ^2 and $\delta^2 = \mathbb{E}[(\hat{\mathbf{x}}_k^{(i)} - \mathbf{x}_k^{(i)})^2]$. The central Kalman filter has all information about the state and observations. Therefore we have an optimal filter which utilizes information fully and gives exact estimation performance as the estimated error covariance, that is $\sigma^2 = \delta^2$.

A distributed *naïve* filter simply ignores the inter-vehicle correlation and treats the information from the two vehicles as independent. We represent the estimation which ignores the correlation by unwrapping the full graphic model. We show how the *naïve* filtering deviates from the central estimator. The unwrapped network has all symbols with a tilde on top, for example, estimated mean is $\tilde{\mu}$, estimated covariance is $\tilde{\sigma}^2$, and actual error covariance is $\tilde{\delta}^2$ of the estimated mean.

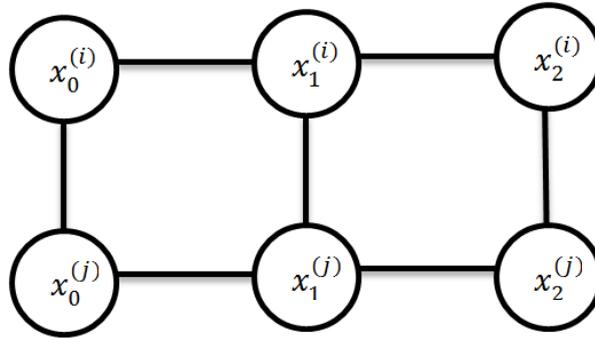
We plot the unwrapped graphs in Figure 4.9 for $k = 1, 2$. It can be seen that the unwrapping simply replicates the past nodes exponentially with $\mathbf{x}_k^{(i)}$ and $\mathbf{x}_k^{(j)}$ being the root nodes. The unwrapped structure assumes the duplicated nodes are from independent sources (independent priors and observations). The exact inference for the unwrapped network is obtained by

$$\tilde{\Lambda}_{\mathbf{xx}}\hat{\tilde{\mathbf{x}}} = \tilde{\mathbf{P}}_{\mathbf{xx}}^{-1}\tilde{\mu}_{\mathbf{x}} - \tilde{\Lambda}_{\mathbf{xz}}\tilde{\mathbf{z}}. \quad (4.31)$$

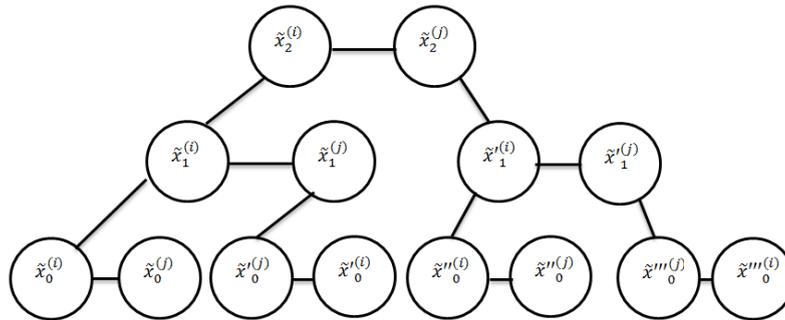


(a) Full graph ($k=1$).

(b) Unwrapped graph ($k=1$).



(c) Full graph ($k=2$).



(d) Unwrapped graph ($k=2$).

FIGURE 4.9: Full graph vs. Unwrapped graph with information double counting for $k = 1, 2$: the prime symbol indicates the replication of nodes.

The replication from the original full network is represented by the mapping matrices \mathbf{O}_x and \mathbf{O}_z such that

$$\tilde{\mathbf{x}} = \mathbf{O}_x \mathbf{x} \tag{4.32}$$

$$\tilde{\mathbf{z}} = \mathbf{O}_z \mathbf{z}.$$

For example, for $k = 1$, the mapping matrices are

$$\mathbf{O}_x = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{O}_z = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.33}$$

where the bold zeros and ones are the block zero matrix and identity matrix respectively, in appropriate sizes (2×2 for \mathbf{O}_x and 3×3 for \mathbf{O}_z). For $k = 2$, we have

$$\mathbf{O}_x = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix}. \quad (4.34)$$

In the mapping, state variables $\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}'_i, \tilde{\mathbf{x}}''_i$ and so on are copies of \mathbf{x}_i ; so are the observations. However, the unwrapped network enables a way for the estimation not to count in the duplication. The independence assumptions are reflected in the structure of $\tilde{\mathbf{\Lambda}}$. The precision matrix ($\mathbf{\Lambda}$ or $\tilde{\mathbf{\Lambda}}$) describes the pairwise statistical relationship given all other nodes. In a Markov network, the entries of precision matrix are only nonzero for neighboring nodes. The relationship between the precision matrices $\tilde{\mathbf{\Lambda}}$ and $\mathbf{\Lambda}$ is listed below:

1. $\tilde{\mathbf{\Lambda}}_{xz} \mathbf{O}_z = \mathbf{O}_x \mathbf{\Lambda}_{xz}$ and $\mathbf{\Lambda}_{xz} \mathbf{O}_z^\top = \mathbf{O}_x^\top \tilde{\mathbf{\Lambda}}_{xz}$: as $\mathbf{\Lambda}_{xz}$ is block-diagonal and the block diagonals of $\tilde{\mathbf{\Lambda}}_{xz}$ are simply replicates of the block diagonals of $\mathbf{\Lambda}_{xz}$. $\mathbf{\Lambda}_{xz}$ and $\tilde{\mathbf{\Lambda}}_{xz}$ are equal after partially being projected to each other's domain.

2. Λ_{zz} and $\tilde{\Lambda}_{zz}$ are diagonal matrices and $\mathbf{O}_z \Lambda_{zz}^{-1} = \tilde{\Lambda}_{zz}^{-1} \mathbf{O}_z$: Λ_{zz}^{-1} is the diagonal observation error covariance matrix. $\tilde{\Lambda}_{zz}^{-1}$ replicates the block diagonals (set of observations) of Λ_{zz}^{-1} . Similarly, Λ_{zz}^{-1} and $\tilde{\Lambda}_{zz}^{-1}$ are equal after partially being projected to each other's domain.
3. An error matrix \mathbf{E} is defined such that $\tilde{\Lambda}_{xx} \mathbf{O}_x + \mathbf{E} = \mathbf{O}_x \Lambda_{xx}$: the row of \mathbf{E} consists all zeros for the nodes who have the same neighbors in the unwrapped network as the original nodes in the full network.
4. $\mathbf{E} = -(\tilde{\mathbf{P}}_{xx}^{-1} \mathbf{O}_x - \mathbf{O}_x \mathbf{P}_{xx}^{-1})$: the property is proved by using block matrix inversion and the relationship summarized above.

Proof. Using the block matrix inversions

$$\mathbf{P}_{xx}^{-1} = \Lambda_{xx} - \Lambda_{xz} \Lambda_{zz}^{-1} \Lambda_{xz}^{\top},$$

$$\tilde{\mathbf{P}}_{xx}^{-1} = \tilde{\Lambda}_{xx} - \tilde{\Lambda}_{xz} \tilde{\Lambda}_{zz}^{-1} \tilde{\Lambda}_{xz}^{\top},$$

we have

$$\begin{aligned}
& \mathbf{O}_x(\Lambda_{xx} - \mathbf{P}_{xx}^{-1}) - (\tilde{\Lambda}_{xx} - \tilde{\mathbf{P}}_{xx}^{-1})\mathbf{O}_x \\
&= \mathbf{O}_x \underline{\Lambda_{xz} \Lambda_{zz}^{-1} \Lambda_{xz}^\top} - \underline{\tilde{\Lambda}_{xz} \tilde{\Lambda}_{zz}^{-1} \tilde{\Lambda}_{xz}^\top} \mathbf{O}_x && \text{by block matrix inversion} \\
&= \underline{\tilde{\Lambda}_{xz} \mathbf{O}_z \Lambda_{zz}^{-1} \Lambda_{xz}^\top} - \underline{\tilde{\Lambda}_{xz} \tilde{\Lambda}_{zz}^{-1} \mathbf{O}_z \Lambda_{xz}^\top} && \text{by relationship between } \Lambda_{xz} \text{ and } \tilde{\Lambda}_{xz} \\
&= \tilde{\Lambda}_{xz} (\mathbf{O}_z \Lambda_{zz}^{-1} - \tilde{\Lambda}_{zz}^{-1} \mathbf{O}_z) \Lambda_{xz}^\top && \text{grouping the similar terms} \\
&= \mathbf{0}. && \text{as } \mathbf{O}_z \Lambda_{zz}^{-1} = \tilde{\Lambda}_{zz}^{-1} \mathbf{O}_z
\end{aligned}$$

The changes in the deviation are underlined, and explanations given on the same lines. Meanwhile, as the error matrix is defined as $\mathbf{E} = \mathbf{O}_x \Lambda_{xx} - \tilde{\Lambda}_{xx} \mathbf{O}_x$, we have

$$\mathbf{O}_x(\Lambda_{xx} - \mathbf{P}_{xx}^{-1}) - (\tilde{\Lambda}_{xx} - \tilde{\mathbf{P}}_{xx}^{-1})\mathbf{O}_x = \mathbf{E} + (\tilde{\mathbf{P}}_{xx}^{-1} \mathbf{O}_x - \mathbf{O}_x \mathbf{P}_{xx}^{-1}).$$

Therefore

$$\mathbf{E} = -(\tilde{\mathbf{P}}_{xx}^{-1} \mathbf{O}_x - \mathbf{O}_x \mathbf{P}_{xx}^{-1}).$$

□

The relationship and properties here will be used in the proofs in the following sections.

4.3.1.1 Estimated Covariance

The inference about the estimated covariance of node $\tilde{\mathbf{x}}_k^{(i)}$ was derived in [87] but we restate it as there is some difference in the way of unwrapping.

Let \mathbf{e} be a column vector with the first element valued as 1 and all other elements, zero ($\mathbf{e}(1) = 1$ and $\mathbf{e}(i) = 0, i \neq 1$).

$$\Lambda_{\mathbf{xx}} \mathbf{C}_{\mathbf{x}|\mathbf{z}} = \mathbf{I} \quad (4.35)$$

$$\Lambda_{\mathbf{xx}} \mathbf{C}_{\mathbf{x}(1)|\mathbf{z}}^\top = \mathbf{e} \quad \text{taking first column of each side} \quad (4.36)$$

$$\mathbf{O}_x \Lambda_{\mathbf{xx}} \mathbf{C}_{\mathbf{x}(1)|\mathbf{z}}^\top = \mathbf{O}_x \mathbf{e} \quad \text{left-multiplied by } \mathbf{O}_x \quad (4.37)$$

$$(\tilde{\Lambda}_{\mathbf{xx}} \mathbf{O}_x + \mathbf{E}) \mathbf{C}_{\mathbf{x}(1)|\mathbf{z}}^\top = \mathbf{O}_x \mathbf{e} \quad \text{as } \tilde{\Lambda}_{\mathbf{xx}} \mathbf{O}_x + \mathbf{E} = \mathbf{O}_x \Lambda_{\mathbf{xx}} \quad (4.38)$$

$$\tilde{\Lambda}_{\mathbf{xx}} \mathbf{O}_x \mathbf{C}_{\mathbf{x}(1)|\mathbf{z}}^\top + \mathbf{E} \mathbf{C}_{\mathbf{x}(1)|\mathbf{z}}^\top = \mathbf{O}_x \mathbf{e}. \quad (4.39)$$

For the unwrapped network, we have similar equation

$$\tilde{\Lambda}_{\mathbf{xx}} \tilde{\mathbf{C}}_{\mathbf{x}(1)|\mathbf{y}}^\top = \tilde{\mathbf{e}}. \quad (4.40)$$

We subtract Equation (4.40) from Equation (4.39) and get

$$\tilde{\Lambda}_{\mathbf{xx}} \tilde{\mathbf{C}}_{\mathbf{x}(1)|\mathbf{z}}^\top = \tilde{\Lambda}_{\mathbf{xx}} \mathbf{O}_x \mathbf{C}_{\mathbf{x}(1)|\mathbf{z}}^\top + \mathbf{E} \mathbf{C}_{\mathbf{x}(1)|\mathbf{z}}^\top + \tilde{\mathbf{e}} - \mathbf{O}_x \mathbf{e} \quad (4.41)$$

$$\tilde{\mathbf{C}}_{\mathbf{x}(1)|\mathbf{z}}^\top = \mathbf{O}_x \mathbf{C}_{\mathbf{x}(1)|\mathbf{z}}^\top + \tilde{\Lambda}_{\mathbf{xx}}^{-1} \mathbf{E} \mathbf{C}_{\mathbf{x}(1)|\mathbf{z}}^\top + \tilde{\Lambda}_{\mathbf{xx}}^{-1} (\tilde{\mathbf{e}} - \mathbf{O}_x \mathbf{e}).$$

As the first row of $\tilde{\Lambda}_{\mathbf{xx}}^{-1}$ is $\tilde{\mathbf{C}}_{\mathbf{x}(1)|\mathbf{z}}$, taking the first element (row) of both sides gives the relationship between $\tilde{\sigma}^2$ and σ^2

$$\begin{aligned}\tilde{\sigma}^2 &= \sigma^2 + \tilde{\mathbf{C}}_{\mathbf{x}(1)|\mathbf{z}}\mathbf{E}\mathbf{C}_{\mathbf{x}(1)|\mathbf{z}}^\top + \tilde{\mathbf{C}}_{\mathbf{x}(1)|\mathbf{z}}(\tilde{\mathbf{e}} - \mathbf{O}_{\mathbf{x}}\mathbf{e}) \\ &= \sigma^2 + \tilde{\mathbf{C}}_{\mathbf{x}(1)|\mathbf{z}}\mathbf{E}\mathbf{C}_{\mathbf{x}(1)|\mathbf{z}}^\top.\end{aligned}\tag{4.42}$$

The last term is dropped as our unwrapping contains no duplication of the root node and therefore $\tilde{\mathbf{e}} - \mathbf{O}_{\mathbf{x}}\mathbf{e} = \mathbf{0}$.

The difference $\tilde{\mathbf{C}}_{\mathbf{x}(1)|\mathbf{z}}\mathbf{E}\mathbf{C}_{\mathbf{x}(1)|\mathbf{z}}^\top$ in Equation (4.42) is actually the first element (entry (1, 1)) of matrix multiplication $\tilde{\mathbf{C}}_{\mathbf{x}|\mathbf{z}}\mathbf{E}\mathbf{C}_{\mathbf{x}|\mathbf{z}}^\top$. We have

$$\begin{aligned}\tilde{\mathbf{C}}_{\mathbf{x}|\mathbf{z}}\mathbf{E}\mathbf{C}_{\mathbf{x}|\mathbf{z}}^\top &= \tilde{\Lambda}_{\mathbf{xx}}^{-1}(\mathbf{O}_{\mathbf{x}}\Lambda_{\mathbf{xx}} - \tilde{\Lambda}_{\mathbf{xx}}\mathbf{O}_{\mathbf{x}})\Lambda_{\mathbf{xx}}^{-\top} \\ &= \tilde{\Lambda}_{\mathbf{xx}}^{-1}\mathbf{O}_{\mathbf{x}} - \mathbf{O}_{\mathbf{x}}\Lambda_{\mathbf{xx}}^{-1}.\end{aligned}\tag{4.43}$$

The first element of these two terms in subtraction are just $\tilde{\sigma}^2$ and σ^2 .

4.3.1.2 Estimated Mean

Using the relationship $\tilde{\Lambda}_{\mathbf{xz}}\mathbf{O}_{\mathbf{z}} = \mathbf{O}_{\mathbf{x}}\Lambda_{\mathbf{xz}}$, Equation (4.31) is

$$\begin{aligned}\tilde{\Lambda}_{\mathbf{xx}}\hat{\mathbf{x}} &= \tilde{\mathbf{P}}_{\mathbf{xx}}^{-1}\tilde{\mu}_{\mathbf{x}} - \tilde{\Lambda}_{\mathbf{xz}}\tilde{\mathbf{z}} \\ &= \tilde{\mathbf{P}}_{\mathbf{xx}}^{-1}\mathbf{O}_{\mathbf{x}}\mu_{\mathbf{x}} - \tilde{\Lambda}_{\mathbf{xz}}\mathbf{O}_{\mathbf{z}}\mathbf{z} \\ &= \tilde{\mathbf{P}}_{\mathbf{xx}}^{-1}\mathbf{O}_{\mathbf{x}}\mu_{\mathbf{x}} - \mathbf{O}_{\mathbf{x}}\Lambda_{\mathbf{xz}}\mathbf{z}.\end{aligned}\tag{4.44}$$

We subtract the above equation from a left-multiplied Equation (4.29) by

\mathbf{O}_x

$$\mathbf{O}_x \Lambda_{xx} \hat{\mathbf{x}} = \mathbf{O}_x \mathbf{P}_{xx}^{-1} \mu_x - \mathbf{O}_x \Lambda_{xz} \mathbf{z} \quad (4.45)$$

and we obtain

$$\begin{aligned} \tilde{\Lambda}_{xx} \hat{\mathbf{x}} &= \mathbf{O}_x \Lambda_{xx} \hat{\mathbf{x}} + \tilde{\mathbf{P}}_{xx}^{-1} \mathbf{O}_x \mu_x - \mathbf{O}_x \mathbf{P}_{xx}^{-1} \mu_x && \text{rearranging the subtraction} \\ &= (\tilde{\Lambda}_{xx} \mathbf{O}_x + \mathbf{E}) \hat{\mathbf{x}} + (\tilde{\mathbf{P}}_{xx}^{-1} \mathbf{O}_x - \mathbf{O}_x \mathbf{P}_{xx}^{-1}) \mu_x && \text{by grouping similar terms} \\ &= \tilde{\Lambda}_{xx} \mathbf{O}_x \hat{\mathbf{x}} + \mathbf{E} \hat{\mathbf{x}} + (\tilde{\mathbf{P}}_{xx}^{-1} \mathbf{O}_x - \mathbf{O}_x \mathbf{P}_{xx}^{-1}) \mu_x \\ \hat{\mathbf{x}} &= \mathbf{O}_x \hat{\mathbf{x}} + \tilde{\Lambda}_{xx}^{-1} \mathbf{E} \hat{\mathbf{x}} + \tilde{\Lambda}_{xx}^{-1} (\tilde{\mathbf{P}}_{xx}^{-1} \mathbf{O}_x - \mathbf{O}_x \mathbf{P}_{xx}^{-1}) \mu_x. \end{aligned}$$

Taking the first element (row) of the estimated mean, we have

$$\begin{aligned} \hat{\mathbf{x}}^{(i)} &= \hat{\mathbf{x}}^{(i)} + \tilde{\mathbf{C}}_{x(1)|z} \mathbf{E} \mu + \tilde{\mathbf{C}}_{x(1)|z} (\tilde{\mathbf{P}}_{xx}^{-1} \mathbf{O}_x - \mathbf{O}_x \mathbf{P}_{xx}^{-1}) \mu_x \\ &= \hat{\mathbf{x}}^{(i)} + \tilde{\mathbf{C}}_{x(1)|z} \mathbf{E} \hat{\mathbf{x}} - \tilde{\mathbf{C}}_{x(1)|z} \mathbf{E} \mu_x \\ &= \hat{\mathbf{x}}^{(i)} + \tilde{\mathbf{C}}_{x(1)|z} \mathbf{E} (\hat{\mathbf{x}} - \mu_x). \end{aligned} \quad (4.46)$$

We now analyze the the performance of the estimated mean which includes:

- The mean error of the estimated mean $\mathbb{E}[\hat{\mathbf{x}}^{(i)} - \mathbf{x}_k^{(i)}]$, and
- The error covariance of the estimated mean $\tilde{\delta}^2 = \mathbb{E}[(\hat{\mathbf{x}}^{(i)} - \mathbf{x}_k^{(i)})^2]$.

The mean error of the estimated mean is zero because

$$\begin{aligned}
\mathbb{E}[\hat{\mathbf{x}}^{(i)} - \mathbf{x}_k^{(i)}] &= \mathbb{E}[\hat{\mathbf{x}}^{(i)} + \tilde{\mathbf{C}}_{\mathbf{x}(1)|\mathbf{z}}\mathbf{E}\hat{\mathbf{x}} - \tilde{\mathbf{C}}_{\mathbf{x}(1)|\mathbf{z}}\mathbf{E}\mu_{\mathbf{x}} - \mathbf{x}_k^{(i)}] \\
&= \mathbb{E}[\hat{\mathbf{x}}^{(i)} - \mathbf{x}_k^{(i)}] + \tilde{\mathbf{C}}_{\mathbf{x}(1)|\mathbf{z}}\mathbf{E}\mathbb{E}[\hat{\mathbf{x}} - \mu_{\mathbf{x}}] \\
&= 0.
\end{aligned} \tag{4.47}$$

The two expectations are zero because the estimation error of the original full network is zero-mean. The error covariance of the estimated mean is

$$\begin{aligned}
\tilde{\delta}^2 &= \mathbb{E}[(\hat{\mathbf{x}}^{(i)} - \mathbf{x}_k^{(i)})^2] \\
&= \mathbb{E}[(\hat{\mathbf{x}}^{(i)} + \tilde{\mathbf{C}}_{\mathbf{x}(1)|\mathbf{z}}\mathbf{E}\hat{\mathbf{x}} - \tilde{\mathbf{C}}_{\mathbf{x}(1)|\mathbf{z}}\mathbf{E}\mu_{\mathbf{x}} - \mathbf{x}_k^{(i)})^2] \\
&= \mathbb{E}[(\hat{\mathbf{x}}^{(i)} - \mathbf{x}_k^{(i)})^2] + \mathbb{E}[(\tilde{\mathbf{C}}_{\mathbf{x}(1)|\mathbf{z}}\mathbf{E}(\hat{\mathbf{x}} - \mu_{\mathbf{x}}))^2] + 2\mathbb{E}[(\hat{\mathbf{x}}^{(i)} - \mathbf{x}_k^{(i)})\tilde{\mathbf{C}}_{\mathbf{x}(1)|\mathbf{z}}\mathbf{E}(\hat{\mathbf{x}} - \mu_{\mathbf{x}})].
\end{aligned} \tag{4.48}$$

In the above equation, the first term $\mathbb{E}[(\mu - x_{A,k})^2] = \sigma^2$, the last term is zero. The second term

$$\begin{aligned}
\mathbb{E}[(\tilde{\mathbf{C}}_{\mathbf{x}(1)|\mathbf{z}}\mathbf{E}(\hat{\mathbf{x}} - \mu_{\mathbf{x}}))^2] &= \mathbb{E}[\tilde{\mathbf{C}}_{\mathbf{x}(1)|\mathbf{z}}\mathbf{E}(\hat{\mathbf{x}} - \mu_{\mathbf{x}})(\hat{\mathbf{x}} - \mu_{\mathbf{x}})^\top \mathbf{E}^\top \tilde{\mathbf{C}}_{\mathbf{x}(1)|\mathbf{z}}^\top] \\
&= \tilde{\mathbf{C}}_{\mathbf{x}(1)|\mathbf{z}}\mathbf{E}\mathbb{E}[(\hat{\mathbf{x}} - \mu_{\mathbf{x}})(\hat{\mathbf{x}} - \mu_{\mathbf{x}})^\top] \mathbf{E}^\top \tilde{\mathbf{C}}_{\mathbf{x}(1)|\mathbf{z}}^\top \\
&= \tilde{\mathbf{C}}_{\mathbf{x}(1)|\mathbf{z}}\mathbf{E}(\mathbf{P}_{\mathbf{xx}} - \mathbf{\Lambda}_{\mathbf{xx}}^{-1})\mathbf{E}^\top \tilde{\mathbf{C}}_{\mathbf{x}(1)|\mathbf{z}}^\top.
\end{aligned} \tag{4.49}$$

Therefore, the error covariance of the estimated mean is

$$\begin{aligned}
\tilde{\delta}^2 &= \sigma^2 + \tilde{\mathbf{C}}_{\mathbf{x}(1)|\mathbf{z}}\mathbf{E}(\mathbf{P}_{\mathbf{xx}} - \mathbf{\Lambda}_{\mathbf{xx}}^{-1})\mathbf{E}^\top \tilde{\mathbf{C}}_{\mathbf{x}(1)|\mathbf{z}}^\top \\
&= \delta^2 + \tilde{\mathbf{C}}_{\mathbf{x}(1)|\mathbf{z}}\mathbf{E}(\mathbf{P}_{\mathbf{xx}} - \mathbf{\Lambda}_{\mathbf{xx}}^{-1})\mathbf{E}^\top \tilde{\mathbf{C}}_{\mathbf{x}(1)|\mathbf{z}}^\top.
\end{aligned} \tag{4.50}$$

It is easy to see that $\mathbf{P}_{\mathbf{xx}} \succeq \mathbf{\Lambda}_{\mathbf{xx}}^{-1}$ and $\tilde{\mathbf{C}}_{\mathbf{x}(1)|\mathbf{z}} \mathbf{E}(\mathbf{P}_{\mathbf{xx}} - \mathbf{\Lambda}_{\mathbf{xx}}^{-1}) \mathbf{E}^\top \tilde{\mathbf{C}}_{\mathbf{x}(1)|\mathbf{z}}^\top \geq 0$.

We have $\tilde{\delta}^2 \geq \delta^2 = \sigma^2$.

4.3.1.3 Summary of the Relationship

$$\begin{aligned}
 \tilde{\sigma}^2 &= \sigma^2 + \tilde{\mathbf{C}}_{\mathbf{x}(1)|\mathbf{z}} \mathbf{E} \mathbf{C}_{\mathbf{x}(1)|\mathbf{z}}^\top \\
 \hat{\mathbf{x}}^{(i)} &= \hat{\mathbf{x}}^{(i)} + \tilde{\mathbf{C}}_{\mathbf{x}(1)|\mathbf{z}} \mathbf{E} (\hat{\mathbf{x}} - \mu_{\mathbf{x}}) \\
 \mathbb{E}[\hat{\mathbf{x}}^{(i)} - \mathbf{x}_k^{(i)}] &= 0 \\
 \mathbb{E}[(\hat{\mathbf{x}}^{(i)} - \mathbf{x}_k^{(i)})^2] &= \tilde{\delta}^2 = \delta^2 + \tilde{\mathbf{C}}_{\mathbf{x}(1)|\mathbf{z}} \mathbf{E} (\mathbf{P}_{\mathbf{xx}} - \mathbf{\Lambda}_{\mathbf{xx}}^{-1}) \mathbf{E}^\top \tilde{\mathbf{C}}_{\mathbf{x}(1)|\mathbf{z}}^\top
 \end{aligned} \tag{4.51}$$

By constructing the unwrapped network of *naïve* filter, we derive the performance of the *naïve* filter compared with central filter using the summarized equations above. The estimation overconfidence can be explained by the first relationship. The estimation divergence comes from the second and fourth equations but it is not easy to visualize. We shall quantify the divergence region in the next section.

4.3.2 Multi-Vehicle Localization with Bathymetric Aids

The multi-vehicle localization consists of vehicles estimating their own positions, and cooperating with an additional relative measurement relating the two vehicles. When bathymetry map is used to assist the localization, the local measurements consist of water depth measurements, and they

are fairly accurate. We look at conditions under which correlation can be ignored.

We consider the recursive two-step process where two vehicles (Vehicle $i = 1$ and $j = 2$) localize themselves (position states $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$) respectively. Similarly we denote previous time step k and current time step $k + 1$. At each step, each node makes a local observation ($\mathbf{z}^{(1)}$ or $\mathbf{z}^{(2)}$) about their respective positions. A range measurement \mathbf{r} is made between vehicles. The propagation model in Equation (4.18) is simplified as a random walk process

$$\begin{aligned}\mathbf{x}_{k+1}^{(1)} &= \mathbf{x}_k^{(1)} + \omega_k^{(1)} \\ \mathbf{x}_{k+1}^{(2)} &= \mathbf{x}_k^{(2)} + \omega_k^{(2)}.\end{aligned}\tag{4.52}$$

The propagation noises are independent of each other with the same error covariance \mathbf{Q} . The measurements $\mathbf{z}_k^{(1)}$, $\mathbf{z}_k^{(2)}$ and r_k in Equation (4.19) have error covariance $\mathbf{R}^{(1)}$, $\mathbf{R}^{(2)}$ and \mathbf{R}_r respectively. Without loss of generality, we assume $\mathbf{R}^{(1)} \leq \mathbf{R}^{(2)}$.

Compared with multi-sensor tracking problem, there are two differences. The first difference lies in an additional variable called ranging r at each step. It relates the two position variables. The second difference comes from individual process noise on position of each vehicle. ω_1 and ω_2 are independent noises.

In step k , the estimates at Vehicle i and j are $\mathbf{y}_k^{(1)}$ and $\mathbf{y}_k^{(2)}$, with the same error covariance \mathbf{P}_k and correlation coefficient ρ_k . Central KF stacks the two state variables and the centralized error covariance is therefore

$$\begin{bmatrix} \mathbf{P}_k & \rho_k \mathbf{P}_k \\ \rho_k \mathbf{P}_k & \mathbf{P}_k \end{bmatrix}.$$

Figure 4.10 and Figure 4.11 show two examples of centralized processing, with respect to different values of ρ_k and \mathbf{R}_r . The different settings of the two cases have a common extreme situation: when the ranging error is 0, the two estimates are fully correlated (and therefore have the same error covariance) after cooperation. When the ranging error approaches zero, the two estimates are almost fully correlated with similar error covariance after cooperation.

We also analyze the performance of SF, NF and DF (with a weighting factor). A SF at Vehicle 1 has

$$\begin{aligned} \mathbf{y}_{k+1}^{(1)} &= \mathbf{y}_k^{(1)} + (\mathbf{P}_k + \mathbf{Q}^{(1)})(\mathbf{P}_k + \mathbf{Q}^{(1)} + \mathbf{R}^{(1)})^{-1}(\mathbf{z}_{k+1}^{(1)} - \mathbf{y}_{k+1}^{(1)}) \\ \mathbf{P}_{k+1}^{(1)} &= (\mathbf{P}_k^{-1} + \mathbf{R}^{(1)-1})^{-1}. \end{aligned} \quad (4.53)$$

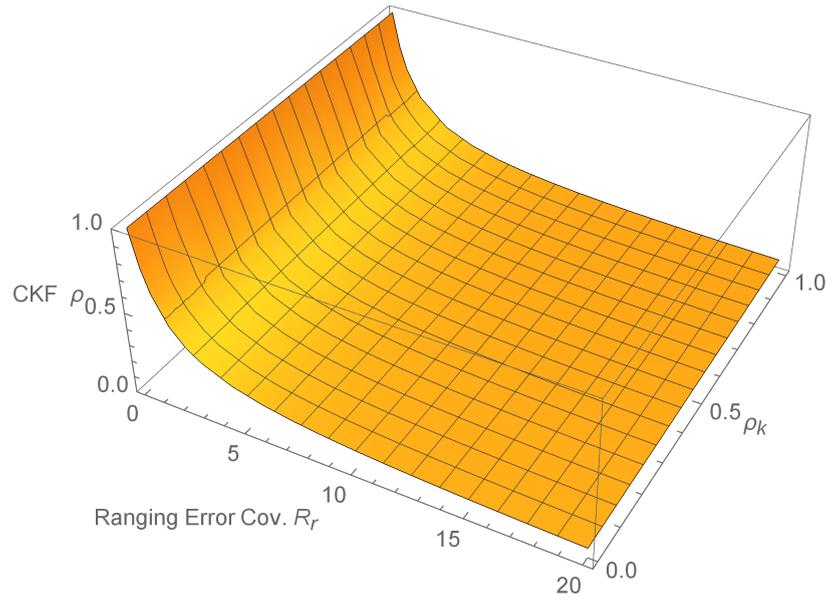
This is standard KF and Vehicle 2 obtains its estimation in the same way.

NF fuses the estimated position from Vehicle 2 about Vehicle 1 such that

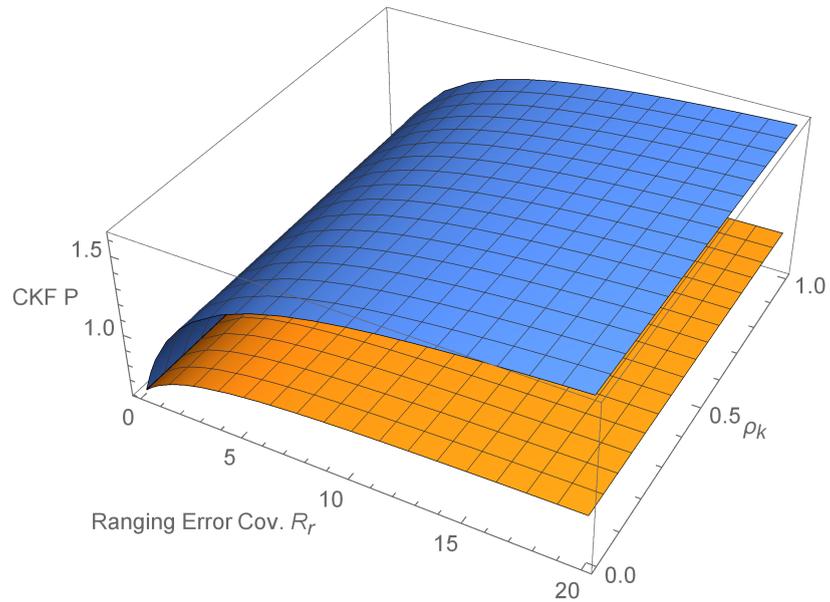
$$\mathbf{y}_{\text{NF}}^{(1)} = \lambda_{\text{NF}} \mathbf{y}_{k+1}^{(1)} + (1 - \lambda_{\text{NF}})(\mathbf{y}_{k+1}^{(2)} + r_{k+1}),$$

where the weight $\lambda_{\text{NF}} = (\mathbf{P}_{k+1}^{(2)} + \mathbf{R}_r)(\mathbf{P}_{k+1}^{(1)} + \mathbf{P}_{k+1}^{(2)} + \mathbf{R}_r)^{-1}$.

DF fuses the estimated position from Vehicle 2 using a weight λ_1 such that $\mathbf{y}_{\text{DF}}^{(1)} = (1 - \lambda_1)\mathbf{y}_{k+1}^{(1)} + \lambda_1(\mathbf{y}_{k+1}^{(2)} + r_{k+1})$. At Vehicle 2, the DF fuses

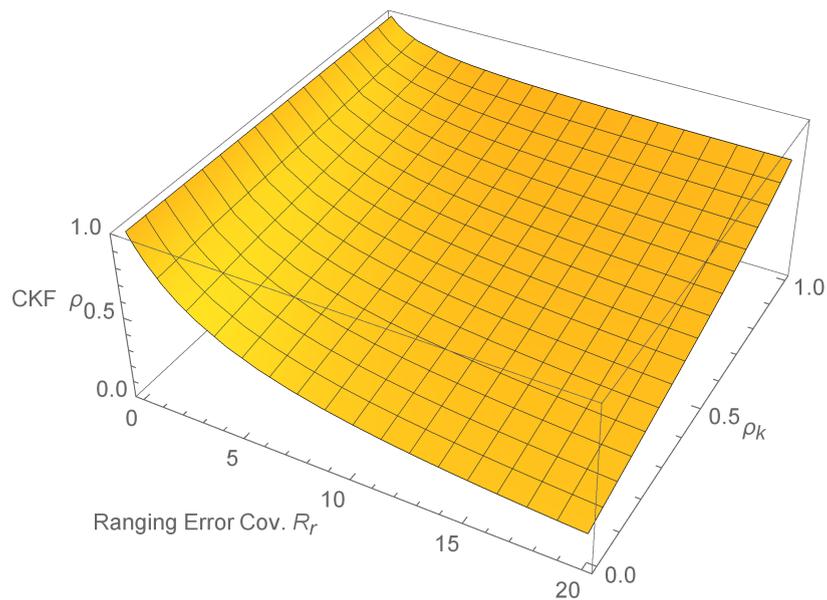


(a) Case 1: Correlation coefficient in one-step cooperation.

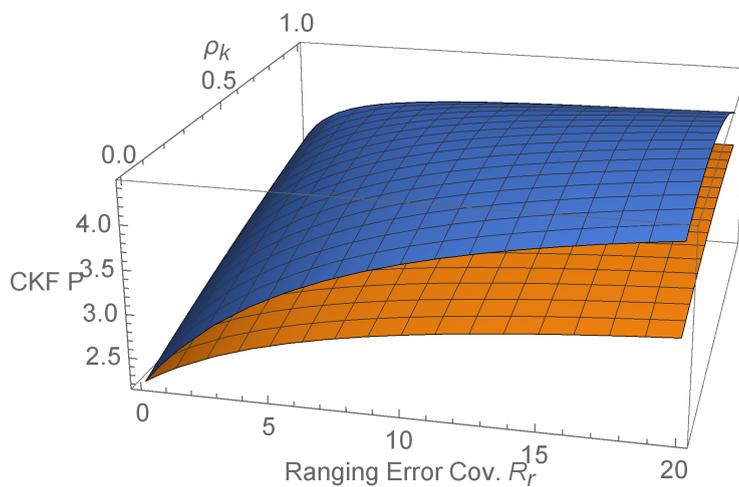


(b) Case 1: Error covariances of two vehicles in one-step cooperation.

FIGURE 4.10: Case 1: $\mathbf{P}_k = 5$, $\mathbf{Q} = 10$, $\mathbf{R}^{(1)} = 1$, $\mathbf{R}^{(2)} = 2$. When ranging error approaches 0, the two estimates are about fully correlated with similar error covariances after cooperation.



(a) Case 2: Correlation coefficient in one-step cooperation.



(b) Case 2: Error covariances of two vehicles in one-step cooperation.

FIGURE 4.11: Case 2: $\mathbf{P}_k = 5$, $\mathbf{Q} = 1$, $\mathbf{R}^{(1)} = 10$, $\mathbf{R}^{(2)} = 40$. When ranging error approaches 0, the two estimates are about fully correlated with similar error covariances after cooperation.

the estimated position from Vehicle 1 using a weight λ_2 such that $\mathbf{y}_{\text{DF}}^{(2)} = (1 - \lambda_2)\mathbf{y}_{k+1}^{(2)} + \lambda_2(\mathbf{y}_{k+1}^{(1)} - r_{k+1})$. The error covariances can be calculated accordingly. The optimal weights can be obtained such that

$$\begin{aligned}\lambda_1^* &= \arg \min \mathbb{E}[(\mathbf{y}_{\text{DF}}^{(1)} - \mathbf{x}_{k+1}^{(1)})^2] \\ \lambda_2^* &= \arg \min \mathbb{E}[(\mathbf{y}_{\text{DF}}^{(2)} - \mathbf{x}_{k+1}^{(2)})^2].\end{aligned}\tag{4.54}$$

When ranging error $\mathbf{R}_r = 0$, we have $\lambda_1^* + \lambda_2^* = 1$ and the two estimates are fully correlated after cooperation. When the ranging error \mathbf{R}_r is very small, $\lambda_1^* + \lambda_2^*$ also approaches 1. In underwater communications, we do have the ability to achieve very small ranging error, compared with positioning error. For simplicity, we set $\mathbf{R}_r = 0$ and therefore the correlation coefficient goes to 1.

4.3.2.1 One-Step Performance

We calculate the one-step *dangerous* region, similar to the situation in Equation (4.16), that is

$$\det \mathbb{E}[(\mathbf{y}_{\text{NF}}^{(1)} - \mathbf{x}_{k+1})(\mathbf{y}_{\text{NF}}^{(f)} - \mathbf{x}_{k+1})^\top] > \det \mathbf{P}_{k+1}^{(1)}.\tag{4.55}$$

Firstly, we consider the extreme cases. When $\mathbf{Q} \rightarrow 0$, the inequality is the same as Equation (4.17) except that the normalized $\bar{\mathbf{R}}^{(i)} = \frac{\mathbf{R}^{(i)}}{\mathbf{P}_k}$. It has the same region in the two-sensor tracking problem. This is easy

to understand as the zero propagation noise makes the two states fully correlated. In such a case, we can see that the DF has the optimal weight $\lambda_{\text{DF}}^* = \frac{\mathbf{R}^{(1)}}{\mathbf{R}^{(1)} + \mathbf{R}^{(2)}}$. The same result can be derived by minimizing the error covariance in Equation (4.54).

When $\mathbf{Q} \gg \mathbf{P}_k$ and $\mathbf{R}^{(i)} \ll \mathbf{Q}$, NF approaches the performance of CF, because the assumption of independence is almost valid. We can ignore the correlation.

When the propagation noise is neither too small nor too large, we derive the *dangerous* region shown in Figure 4.12 when Equation (4.55) is met. This region tells us when the local propagation error is small, the states are more correlated. Vehicle 2 maintains most of the correlated information when $\mathbf{R}^{(2)} > \mathbf{R}^{(1)}$. In such a case, if the correlation in the estimate from Vehicle 2 is ignored, the fused estimate is worse than the estimate from SF.

4.3.2.2 Asymptotic Performance

Figure 4.13 shows the estimation performance in stable state. The cases are located in Figure 4.14 using their parameter values. The normalized $\bar{\mathbf{R}}^{(i)} = \mathbf{R}^{(i)}/\mathbf{Q}$. In the three cases, cooperative localization with bathymetric aids is most similar to Case (b) and Case (c). The relative distance measured by acoustic signals has small error (\mathbf{R}_r is in the sub-meters range); the local measurement, i.e., the water depth has much smaller errors compared with

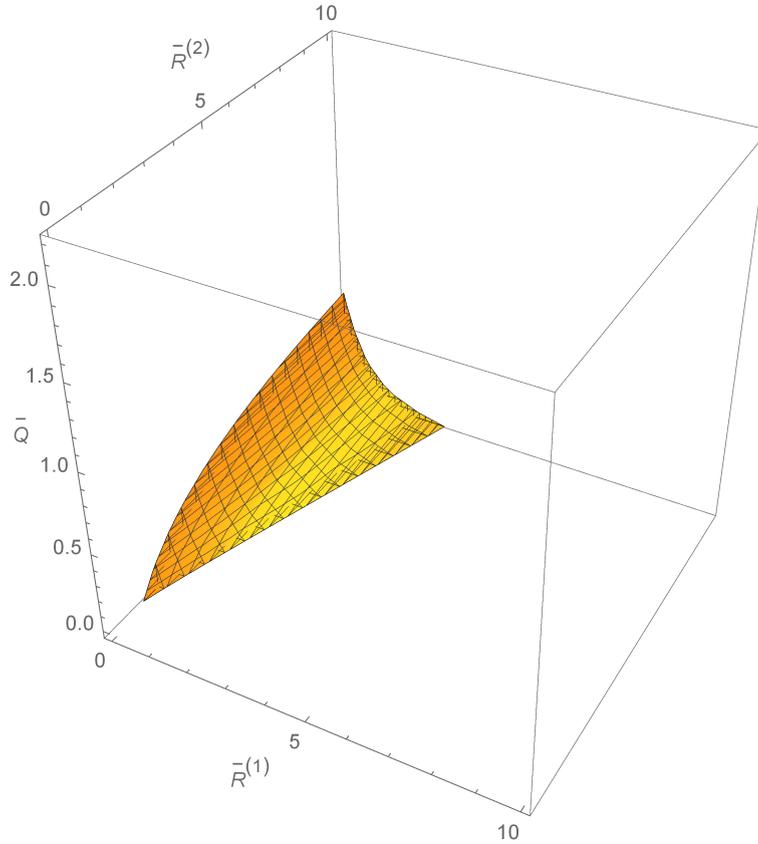
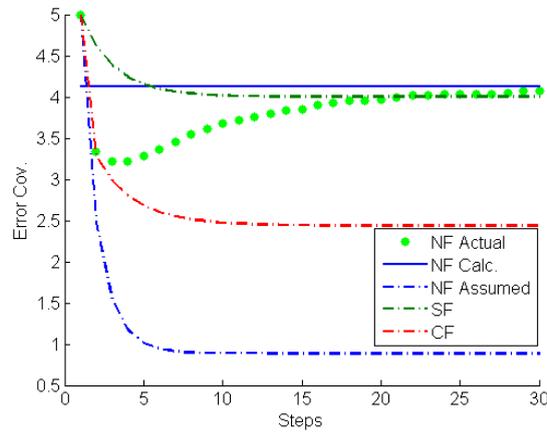


FIGURE 4.12: One-step Performance: The *dangerous* region of implementing NF in multi-vehicle localization. ($\bar{\mathbf{R}}^{(2)} > \bar{\mathbf{R}}^{(1)}$)

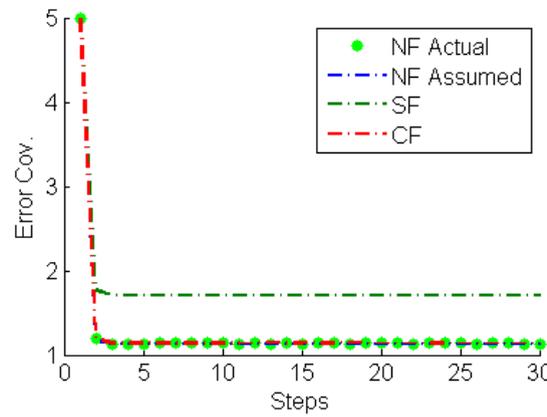
the propagation errors accumulated between cooperation. Even if any one of the cooperative vehicles has little or none of the bathymetry aids (large measurement error like Case (c)), it is still safe to ignore the correlation.

4.4 Summary

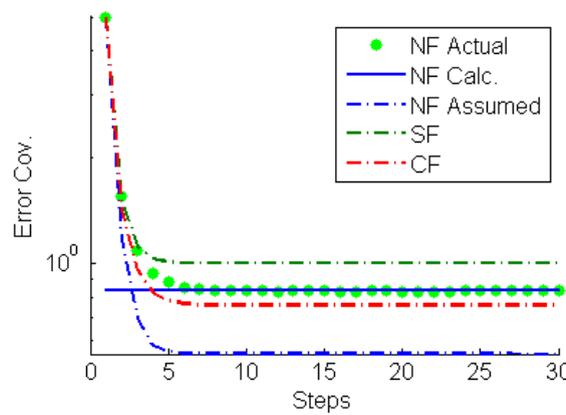
We demonstrated why there is information loss in distributed localization, and how the information is double counted if one ignores the correlation.



(a) Actual error covariance eventually grows larger than SF.



(b) It is safe to ignore correlation and the performance is close to CF.



(c) Actual error covariance is smaller than SF but the estimation is overconfident.

FIGURE 4.13: *Naïve* filter in stable state: One can ignore the correlation in Case (b) and Case (c) but NF becomes detrimental in Case (a).

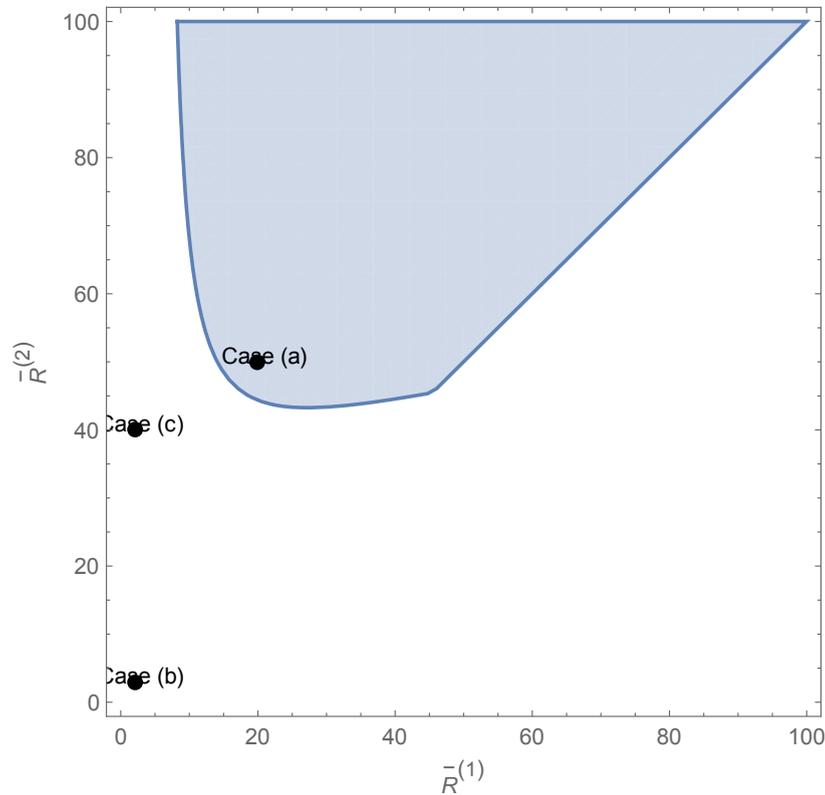


FIGURE 4.14: Asymptotic Performance: The *dangerous* region of implementing NF in multi-vehicle localization. The three cases in Figure 4.13 are located in the regions.

For a cooperation method, when the actual localization is worse than single-vehicle localization, we call it *dangerous* as this cooperation makes the estimation worse. When the actual localization is better than single-vehicle localization, we perceive it safe as this cooperation helps. For both types of cooperative localization - multi-sensor tracking problem and multi-vehicle localization problem, we quantified the safe and *dangerous* regions when ignoring correlation during fusion. This can be used as a guideline to justify the *naïve* assumption. The *naïve* assumption is justified safe to use in cooperative localization with bathymetric aids in the next chapter.

Chapter 5

Localization with Bathymetric Aids

5.1 Problem Statement

Chapter 4 shows that cooperative localization with bathymetric aids can ignore the correlation when fusing information from range updates. With this justification, we proceed to explore the bathymetry-aided navigation on a single vehicle. This can be safely extended to cooperative navigation of multiple vehicles. The first work investigates how the bathymetric terrain map benefits the localization. As many works have claimed that the localization performance highly depends on the bathymetry variation [35, 37, 47, 62], we justify this assumption through a careful analysis. It

is again proved that the advantage of bathymetric aids is path dependent. However the bathymetry variation is not a sufficient condition for a good localization. A concept of information entropy map, is formulated and used in Chapter 6 later as an evaluation metric in path planning.

The work in this chapter and Chapter 6 was published in [71].

5.2 Probability Map Based Localization

We denote the entire location space at time k as \mathbf{X}_k . $\text{Bel}(\mathbf{X}_k = x)$ denotes the vehicle's belief that it is at the location x at time k . The action \mathbf{a}_k denotes the action taken at time step k towards the next step $k + 1$, and $\mathbf{A}_{1:k} \doteq \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k\}$. The same definition and notation apply to the sensing data \mathbf{y}_k . Markov process assumes that given the present state, the future and past states are independent. In formal terms, it is stated

$$P(\mathbf{x}_k | \mathbf{x}_0, \dots, \mathbf{x}_{k-1}, \mathbf{a}_0, \dots, \mathbf{a}_{k-1}, \mathbf{zS}_0, \dots, \mathbf{z}_k,) = P(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{a}_{k-1}). \quad (5.1)$$

This only works if the environment is static and does not change with time [83]. Kalman filter described in the previous chapter is one of the common methods used. However, Kalman filter assumes Gaussian noises in estimation and measurements. It does not perform well for situations where the models are nonlinear or the belief is multimodal. For example,

a vehicle moving along the ridge of a hill may arrive at a belief that has two peaks at its two sides. In such situations, filters with non-parametric representation can give better description.

Two types of localization methods are introduced in the subsequent sections - grid-based Markov localization and particle filtering. We examine the effect of bathymetry aids on localization. This includes how the bathymetry affects the localization belief, which estimation filter should be used, and how the localization performance should be quantified.

5.2.1 Grid-Based Markov Localization

The grid-based localization uses a histogram to represent the belief distribution in map grids. We use the algorithm in [34] to simulate the grid-based Markov localization near St John's Island, Singapore. We assume that we have no idea where the vehicle is at the beginning, and hence we initialize a uniform prior distribution in this area. The grid-based Markov localization has the ability to represent situations where the position of the vehicle is held by multiple, distinct beliefs. The probability could be any form instead of single Gaussian. It also caters to the situation where the initial position is unknown.

There are two steps in the Markov localization: action (propagation) and sensing (observation). The corresponding estimation is to predict and

update the belief. The uncertainty in the prediction smooths out the location possibility while the observation for update reinforces the places which have similar water depth (bathymetry) as measured. The resolution of the surveyed bathymetry restricts the accuracy of positioning. The computation load is comparable to the grid map resolution and is fairly high. In the St John's map, for example, we have an area of 936 meters by 349 meters, with 1 meter resolution, giving a total of 326664 grids, of which 192956 grids are underwater.

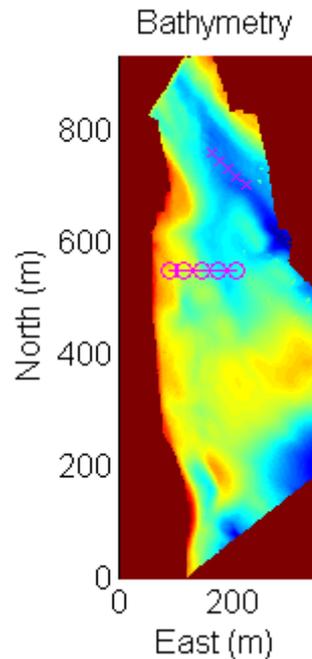
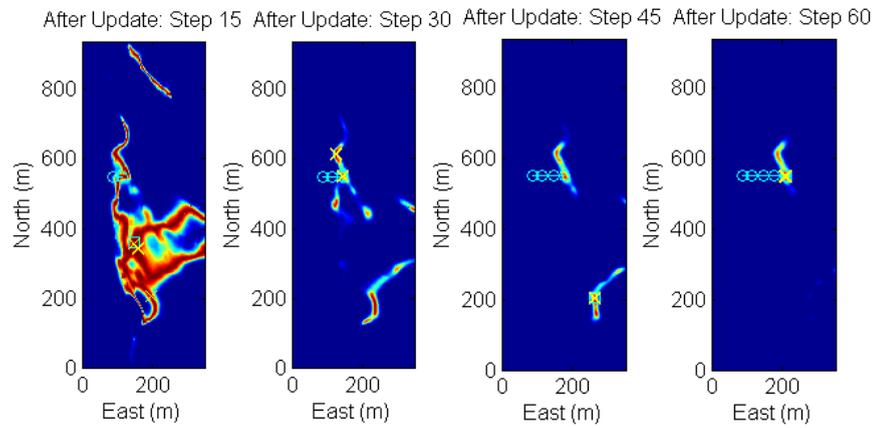
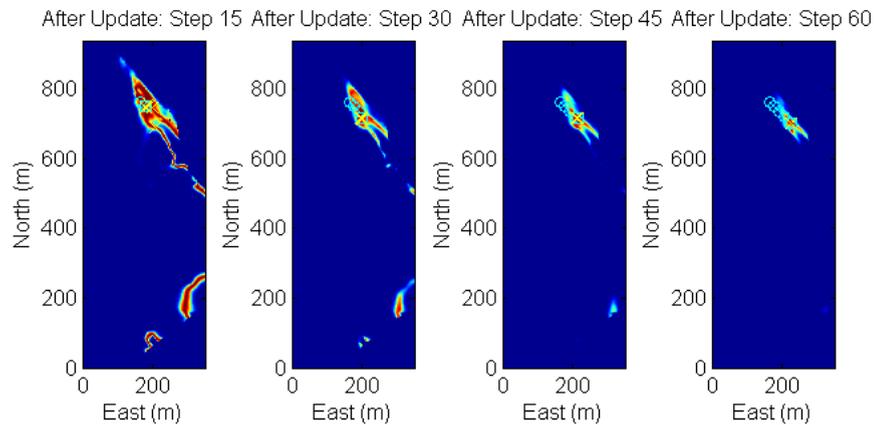


FIGURE 5.1: Bathymetry map near St. John Island, Singapore. Circles: Path 1 (Speed 2 m/s). Crosses: Path 2 (Speed 1.41m/s).

Figure 5.1 shows two test paths on the bathymetry map. The simulation assumes of odometric error of 1 meter per second in standard deviation, and measurement error of 0.1 meters in standard deviation. There



(a) Path 1.



(b) Path 2.

FIGURE 5.2: Grid-based Markov localization with measurement updates. Yellow crosses: top 5 possible locations. Cyan circles: true path. Green square: estimated locations with top possibility.

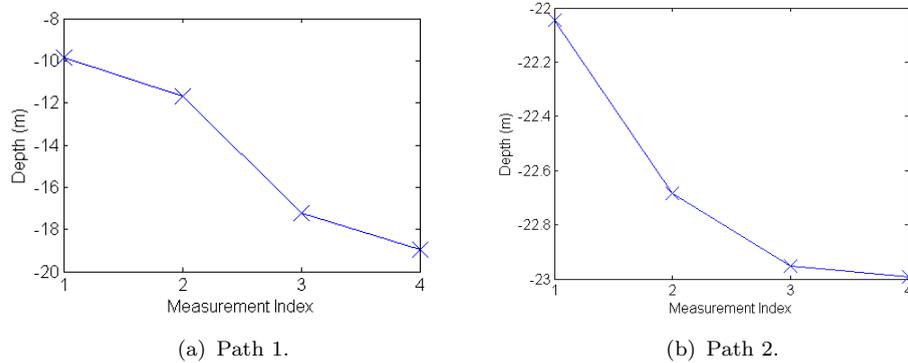


FIGURE 5.3: Depth measurements along two paths.

is a measurement every other 15 seconds. Both start with uniform beliefs on the vehicle location over the map. The decision of the position by the top beliefs at initial few measurements is mostly incorrect as there could be many locations with similar top probability. Multiple hypotheses on the location appears. With more bathymetry measurements (from left to right in Figure 5.2), the probability is updated from multiple peaks or ridges to eventually a single peak. The decision on the estimated position converges. However Path 2 localization converges from the second depth measurement onward, whereas Path 1 localization only converges at the 4th measurement. In terms of bathymetry variations along the path, Path 1 has more variations as shown in Figure 5.3. The underwater topology variations (richness of features) on the path are closely related to the positioning performance. The sparser the underwater topology, the less effective are bathymetry aids to localization. However bathymetry variation is not the sufficient or necessary condition for a good localization. The uniqueness

of the measured depth compared with the bathymetry in the prior belief decides the localization performance. Multiple AUVs with cooperation are able to extract more uniqueness in the feature space.

5.2.2 Particle Filtering and Multiple Hypotheses

At time step k the particle filter gives the particle set

$$\{\mathbf{x}_k^{(i)}, q_k^{(i)}\} \quad (5.2)$$

where $q_k^{(i)}$ is the weight for particle i at position $\mathbf{x}_k^{(i)}$ and $\sum_{i=1}^N q_k^{(i)} = 1$.

The computation of PF is determined by the number of particles used. Figure 5.4 shows the corresponding PF estimation on Path 1. The density of the particles cannot be seen due to the overlapping of the particles. However it gives similar result as Figure 5.2(a) with less computation load (5000 particles).

We model the particles with Gaussian mixture model (GMM) using Expectation Maximization (EM) method [16]. Classifying the particles is essentially a clustering problem. A Gaussian Mixture consists of a linear superposition of Gaussians

$$p(\mathbf{x}) = \sum_{i=1}^K \lambda_i \mathcal{N}(\mathbf{x}|\mu_i, \Sigma_i) \quad (5.3)$$

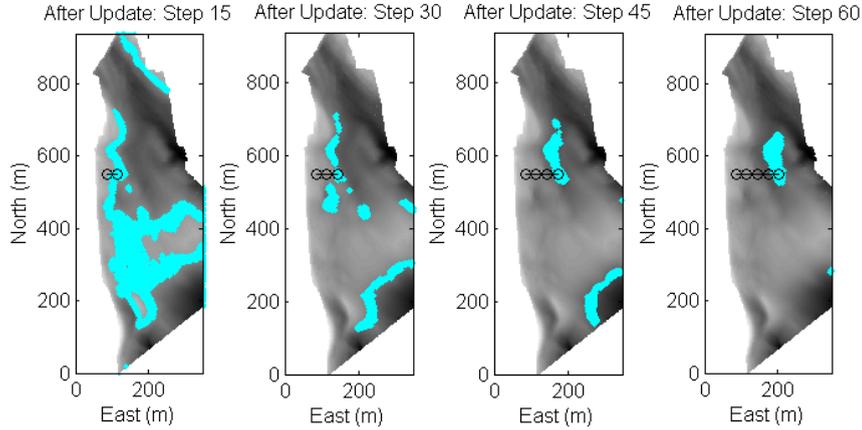


FIGURE 5.4: Particle filter localization with measurement updates for Path 1. Black circles: true path. Cyan regions: distribution of particles.

where $\sum_{i=1}^K \lambda_i = 1$ are the mixing coefficients and K is the number of Gaussians. EM is a recursive method to maximize the likelihood function with respect to the parameters comprising the means and covariances of the Gaussians and the mixing coefficients. Considering the mixing coefficients as prior probabilities for the particles, for a given value ‘ \mathbf{x} ’, we evaluate the corresponding posterior probabilities, also called *responsibilities*:

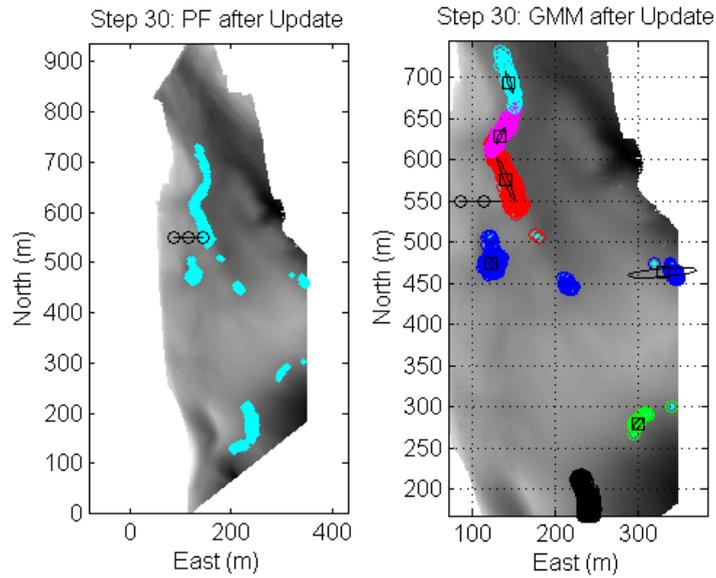
$$\begin{aligned}
 \gamma_i(\mathbf{x}) &= p(i|\mathbf{x}) \\
 &= \frac{p(i)p(\mathbf{x}|i)}{p(\mathbf{x})} \\
 &= \frac{\lambda_i \mathcal{N}(\mathbf{x}|\mu_i, \Sigma_i)}{\sum_{j=1}^K \lambda_j \mathcal{N}(\mathbf{x}|\mu_j, \Sigma_j)}.
 \end{aligned} \tag{5.4}$$

When N_i particles are assigned to the i th particles, we have $\lambda_i = \frac{N_i}{N}$.

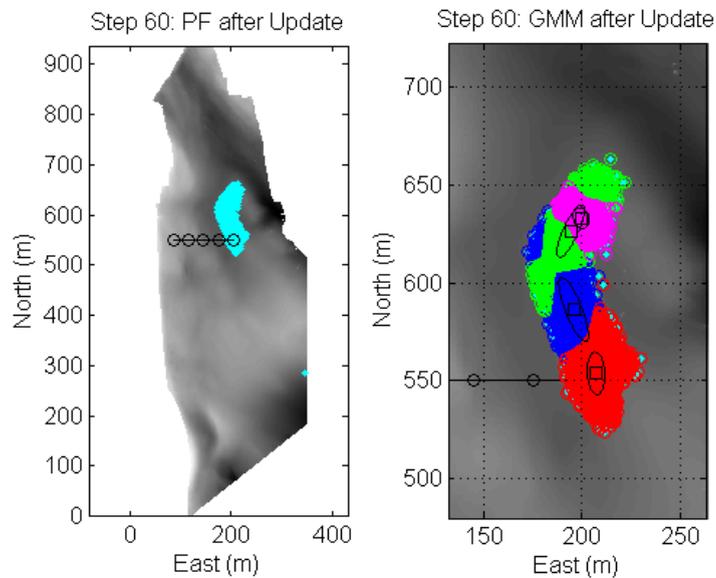
In each iteration, $\gamma_i(\mathbf{x})$ is evaluated and the parameters are re-estimated.

With the new estimated parameters, the log likelihood $\ln p(\mathbf{x}|\mu, \Sigma, \lambda)$ can

be evaluated. The iteration stops when there is convergence or maximum number of iterations is reached.



(a) Step 45.



(b) Step 60.

FIGURE 5.5: Gaussian mixture model estimated by EM method.

Figure 5.5 shows two examples of the Gaussian mixtures estimated by

EM method. Each cluster is plotted in different colors, with the mean in black square and error covariance in black ellipses. When particles form multiple clusters, GMM is more representative. When particles form fewer clusters, the number of particles affects the calculation and estimation performance.

Generally, the Gaussian mixture model for multiple hypothesis increases the computation in modeling. The individual Gaussian model is also not very representative. Overall prediction and update with multiple Gaussian does not give much advantage compared with particle filter itself.

5.3 Information Entropy Map

In the previous section, we have demonstrated that with bathymetric measurements incorporated into localization, the *a posteriori* description of the location uncertainty is often poorly described by a Gaussian distribution. Multimodal distributions may arise when the location uncertainty is bifurcated at bathymetric ridges. Particle filter (PF) is used as a flexible tool to represent general densities. However, the traditional evaluation measures such as error of estimated mean and estimated error covariance are only suitable for single Gaussian-distribution cases [26]. Gaussian mixture model has the problem with varying optimal number of Gaussians.

We adopt information entropy measure to characterize the estimation uncertainty, which does not require parametric estimation of the position. This section introduces two types of information entropy measures for filtering uncertainty - the grid-based discrete entropy and PF-based entropy. Both display similar trends in describing the estimation uncertainty. We will illustrate the entropy values along different paths.

5.3.1 Grid-based Discrete Entropy

From a mathematical perspective, the calculation of entropy is based on the probability of all possible outcomes. However for particle filter localization, simply counting the probability (weights) of particles for discrete entropy will result in loss of location and dispersion information. Particles have to be related to geographical location. One way is to count the number of particles in the map grids. We calculate the discrete entropy of the vehicle position estimation

$$H(\mathbf{X}) = - \sum_{i=1}^{MN} P(\mathbf{x}_i) \log P(\mathbf{x}_i) \quad (5.5)$$

where M and N are the number of grids in longitude and latitude directions, i is the index of grid up from 1 to MN . $P(\mathbf{x}_i)$ is the probability mass function at i th grid centered at position \mathbf{x}_i and $\sum_i^{MN} P(\mathbf{x}_i) = 1$. $P(\mathbf{x}_i)$ is

obtained by summing the weights of particles which fall into the i th grid. In the case of $P(\mathbf{x}_i) = 0$ for some i , the value of $0 \log 0$ is defined to be 0.

The entropy value defines the uncertainty (or uniqueness in the other way) of the localization on a grid map. When there are a few grids with high probability while the rest of the grids have low probability, the entropy value is small. Therefore the filter has less uncertainty about vehicle's location because the vehicle most likely falls into one of the few grids with high probability. If most grids have equal probability, the filter is more uncertain where the vehicle is located.

The next information for localization comes from the observed water depth. As mentioned before, this is in fact a sum of two measurements - the vehicle depth (the depth from the sea surface to the vehicle) and the altitude (the depth from vehicle to sea bottom). The single-point water depth measurement z is assumed to be corrupted with zero-mean Gaussian noise $z_k = h_k(\mathbf{x}_x) + \nu_k$ and $\nu_k \sim \mathcal{N}(0, \mathbf{R})$. Comparing with the bathymetry map in records, the localization can be refined.

Let the water depth variable be Z . With all the bathymetry map readings, we can find $P(\mathbf{Z}|\mathbf{X})$ for all possible values (z, \mathbf{x}) , and subsequently we obtain $P(z) = \sum_{\mathbf{x}} P(z|\mathbf{x})$. According to Bayes Theorem, we have

$$P(\mathbf{x}|z) = \frac{P(z|\mathbf{x})P(\mathbf{x})}{P(z)} \quad (5.6)$$

and therefore the conditional entropy is

$$H(\mathbf{X}|\mathbf{Z}) = \sum_{z_i} p(z_i)H(\mathbf{X}|Z = z_i). \quad (5.7)$$

The conditional entropy $H(\mathbf{X}|\mathbf{Z})$ tells on average how much localization uncertainty is left after observing the water depth within an area. The reduced amount is the mutual information $I(\mathbf{X}; \mathbf{Z})$ provided by prior about the location \mathbf{X} and bathymetry information \mathbf{Z} . We can also calculate the conditional entropy when a single measurement is made, that is $H(\mathbf{X}|Z = z)$.

It should be noted that the accuracy of discrete entropy depends on the size of the map grid (map resolution). An extreme example is when all particles fall into one grid in the map, the discrete entropy falls to a minimum value - zero. Meanwhile, number of particles in the filter is also critical to the accuracy of discrete entropy. To avoid the discretization error from particle filter, bivariate kernel density [19] is first used to estimate the distribution. Then the discrete distribution is interpolated and normalized on the map grids.

5.3.2 Particle Filter Based Entropy

For dynamic model and measurement model described in Equations (2.1) and (2.2), the entropy in a running particle filter has been derived in [17] as

$$H(p(\mathbf{x}_k|Z_{1:k})) \approx \log \left(\sum_{i=1}^N p(z_k|\mathbf{x}_k^i) q_{k-1}^i \right) - \sum_{i=1}^N \log \left(p(z_k|\mathbf{x}_k^i) \left(\sum_{j=1}^N p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^j) q_{k-1}^j \right) \right) q_k^i, \quad (5.8)$$

where $Z_{1:k} = \{z_1, z_2, \dots, z_k\}$ includes the measurements in history up to time step k . $p(\mathbf{x}_k|Z_k)$ is the posterior distribution after the series of bathymetry measurements.

As measurements may not be available at every step, we derive the entropy in predicting vehicle position from time step $k-1$ to k . $\mathbf{x}_{k|k-1}^i$ denotes the predicted position of the i th particle in particle set. $q_{k|k-1}^i$ is the associated particle weight. The probability distribution at the predicted stage represented by PF is

$$p(\mathbf{x}_k|Z_{1:k-1}) \approx \sum_{i=1}^N q_{k|k-1}^i \delta(\mathbf{x} - \mathbf{x}_{k|k-1}^i). \quad (5.9)$$

The weak convergence law for PF states that

$$\lim_{N \rightarrow \infty} \sum_{i=1}^N g(\mathbf{x}_{k|k-1}^i) q_{k|k-1}^i = \int_{\mathcal{X}} g(\mathbf{x}_k) p(\mathbf{x}_k|Z_{1:k-1}) d\mathbf{x}_k. \quad (5.10)$$

Therefore the entropy is

$$\begin{aligned}
H(p(\mathbf{x}_k|Z_{1:k-1})) &= - \int_{\mathcal{X}} \log p(\mathbf{x}_k|Z_{1:k-1}) p(\mathbf{x}_k|Z_{1:k-1}) d\mathbf{x}_k \\
&= - \lim_{N \rightarrow \infty} \sum_{i=1}^N \log p(\mathbf{x}_{k|k-1}^i | Z_{1:k-1}) q_{k|k-1}^i \\
&= - \lim_{N \rightarrow \infty} \sum_{i=1}^N \log \left(\lim_{N \rightarrow \infty} \sum_{j=1}^N p(\mathbf{x}_{k|k-1}^i | \mathbf{x}_{k-1}^j) q_{k-1}^j \right) q_{k|k-1}^i \\
&\approx - \sum_{i=1}^N \log \left(\sum_{j=1}^N p(\mathbf{x}_{k|k-1}^i | \mathbf{x}_{k-1}^j) q_{k-1}^j \right) q_{k|k-1}^i.
\end{aligned} \tag{5.11}$$

It can be seen that this is equivalent to the PF-entropy approximation in Equation (5.8) when $p(z_k|x_k^i) = 1$. It is equivalent to an observation which provides no information updating the distribution.

5.3.3 Empirical Convergence and Contributing Factors

A simple random walk process is used to illustrate the grid-based entropy and PF-based entropy. The measurement is observed position with additive Gaussian noise. A standard Kalman filter tracks the estimated covariance and therefore the theoretical entropy of the Gaussian distribution $H = \frac{1}{2} \log\{(2\pi e) \det \Sigma\}$ is calculated as benchmark. Figure 5.6 shows the standard deviation error with two different initial errors.

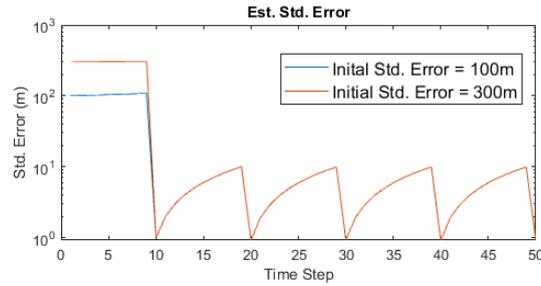


FIGURE 5.6: Gaussian random walk process: Estimation error is reduced at every measurement update.

Figure 5.7 shows PF-based entropy is almost identical to the theoretical entropy, except when number of particle is only 1000 (Figure 5.7(a) and 5.7(b)). Grid-based discrete entropy has different values but the same trend as the estimation error. Both entropy values drop with the estimation error reduction from measurement update. Resampling of particles does not affect PF-based entropy. This is because PF-based entropy is calculated based on the transition and weight update of each particle and resampling happens after that if needed. Resampling of particles affect the performance of Grid-based discrete entropy. This is because resampling is necessary as the first step for the kernel density estimation for the discrete entropy.

The advantage of grid-based discrete entropy is that it can be used to calculate the average conditional entropy $H(\mathbf{X}|\mathbf{Z})$. $H(\mathbf{X}|\mathbf{Z})$ can be used to describe the effectiveness of a bathymetry map based on a particular prior knowledge. The drawback is that the accuracy depends on the grid size as the bin weights are obtained by including particles which fall into

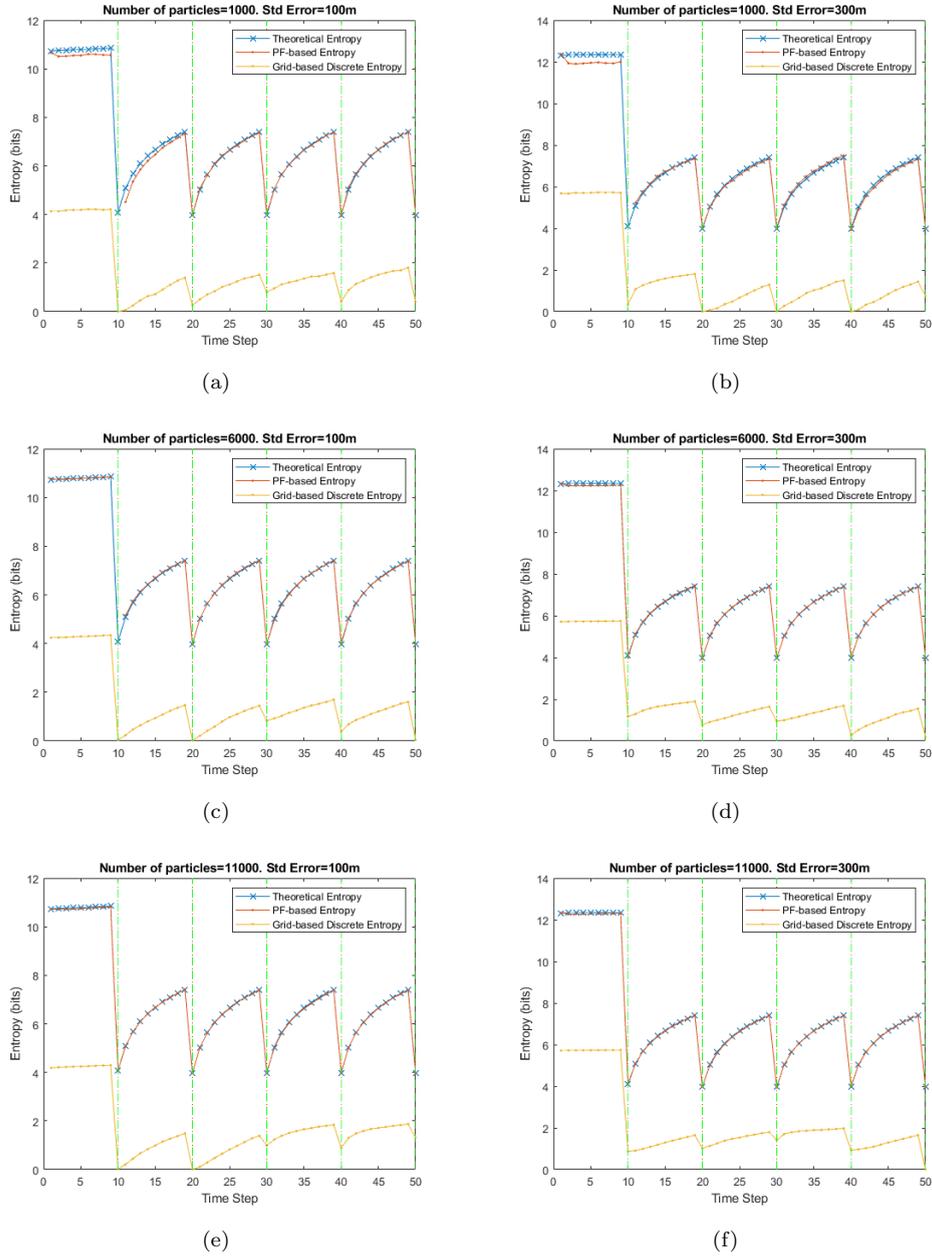


FIGURE 5.7: Gaussian random walk process: PF-based Entropy and grid-based discrete entropy, versus theoretical entropy. Vertical red lines: time steps when measurements are available. Vertical green dashed lines: time steps when particles are resampled. Grid-based entropy is more sensitive to particle numbers and estimation error values.

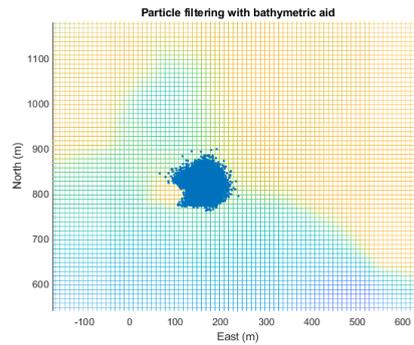
the same grid. Under-estimation of entropy occurs when either the prior or number of particle is small. For example, with the same initial error,

grid-based discrete entropy values are different with different number of particles (comparing each column of Figure 5.7). This is because a smaller number of particles is insufficient to fully describe larger estimation error. In the other hand, PF-based entropy only calculates the conditional entropy $H(\mathbf{X}|\mathbf{Z} = z)$ with a specific observation but it is generally more stable with respect to particle numbers and prior knowledge. It is an approximation of the entropy of probability density function (PDF) using probability mass function (PMF). For PF-based entropy, the variation is only more obvious when the estimation error is larger and number of particles is relatively small (Figure 5.7(a) and 5.7(b)).

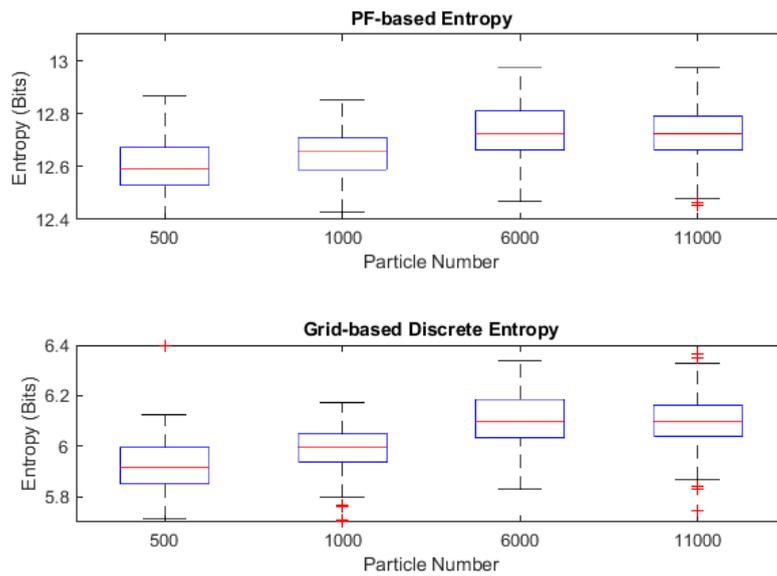
An example of the effect of particle number with bathymetry observations is shown in Figure 5.8. With particle numbers varying from 500 to 11000, the grid-based discrete entropy has larger variation of the median value in red line than the PF-based entropy. It also gives more outliers. This means grid-based entropy is more sensitive to the particle numbers.

5.3.4 Localization Performance

With the comparison from previous section, we use PF-based entropy as the localization uncertainty measure. We examine the PF-based entropy along two paths shown in Figure 5.9(a). Both paths have the same large initial

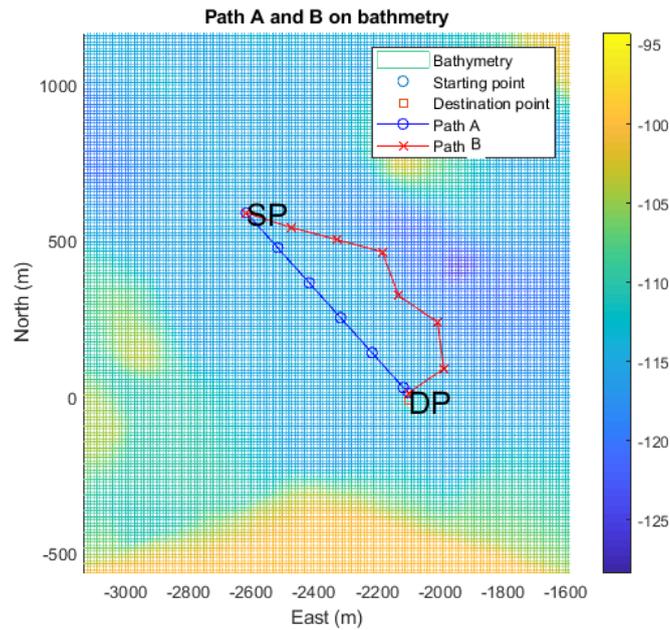


(a) A non-Gaussian distribution after observation (Particle Numbers = 11000): particle distribution highly depends on the bathymetry map.

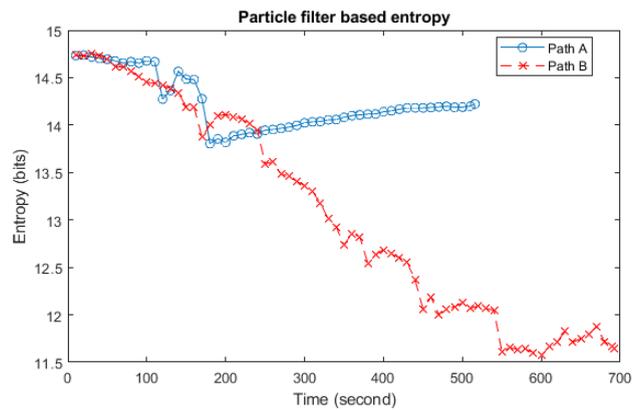


(b) Entropy Boxplot: PF-based entropy versus grid-based entropy in the same range. Grid-based entropy has larger variation in the median of the entropy values.

FIGURE 5.8: PF-based entropy versus grid-based discrete entropy: Grid-based discrete entropy varies more, with respect to particle number.



(a) Two different paths (A and B) from the same starting point (SP) to the destination point (DP).



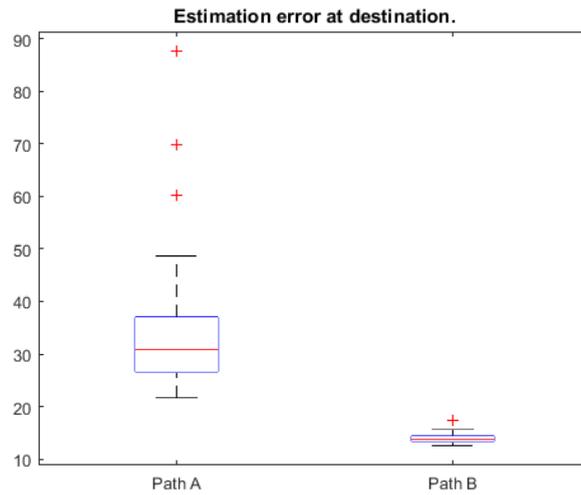
(b) Entropy of the particle filter for paths A and B shown in (a).

FIGURE 5.9: The entropy of the particles in a bathymetric navigation particle filter depends on the path taken from source to destination.

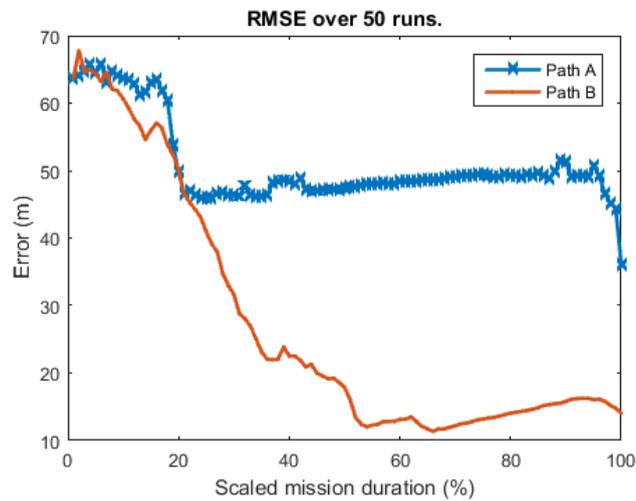
uncertainty. Bathymetric measurements made at every 10 seconds help reduce the localization uncertainty initially for both paths. Straight-line path A goes through a flat area with little bathymetry variation. The entropy of localization uncertainty increases from roughly the 200th second. Path B takes a detour and therefore longer time to reach the destination. Without bathymetric information, the vehicle would incur a larger uncertainty for longer missions due to error accumulation in pure dead reckoning. With bathymetric information, the PF-based entropy decreases rapidly as the vehicle moves along the area with significant bathymetric variability. The significant variation along path B makes the measured bathymetry unique and therefore improves localization accuracy. At the destination, Path B has a smaller entropy compared with Path A.

In Figure 5.10(b), the root-mean-squared errors along Path A and B over 50 runs are plotted. As the mission time for each run is different, time is scaled to the percentage of the total mission time. The localization error has the same trend of the PF-entropy in Figure 5.9(b). Figure 5.10(a) shows the localization error when the vehicle reaches the destination.

With this example, we have shown that the localization performance can be evaluated using information entropy measure. We also show that paths with different bathymetry give different localization results.



(a) When vehicle reaches the destination, the localization error and the covariance are much larger for Path A than that for Path B.



(b) Localization error along Path A is also larger than that along Path B.

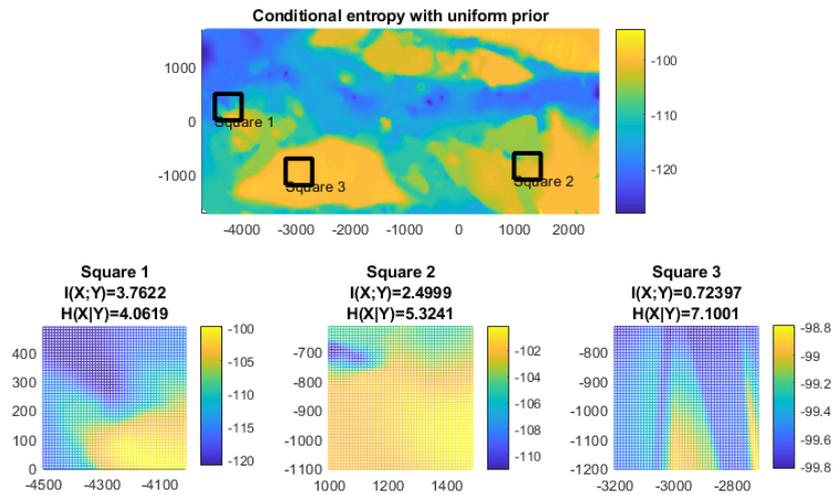
FIGURE 5.10: Although Path B takes a detour and therefore longer time to reach the destination, the localization error of Path B is much smaller than a that of a straight line by Path A.

5.3.5 Information Entropy Map Based on Different Priors

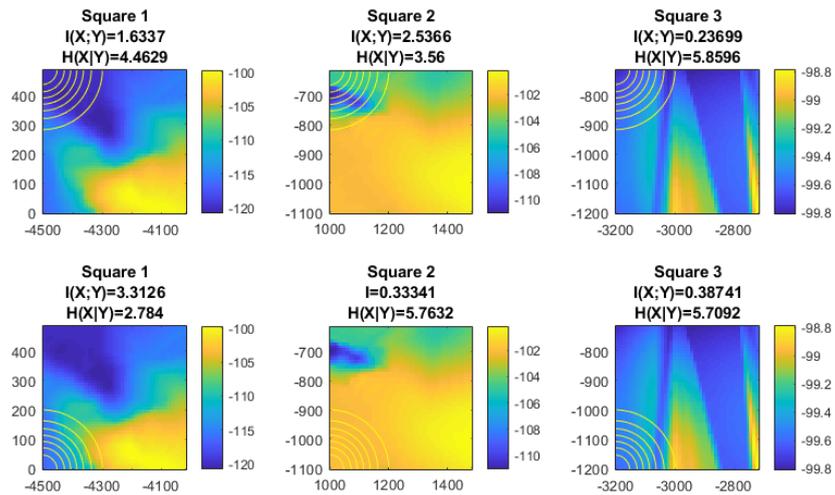
The intuitive way is to select the path with the most bathymetry variability in the map. However, is bathymetry variability the sufficient condition for optimal localization? we answer this question in this section by examining the information entropy map.

Discrete conditional entropy is first used to evaluate the effect of the information that the local bathymetry provides. Given a prior distribution, the conditional entropy $H(\mathbf{X}|\mathbf{Z})$ and mutual information $I(\mathbf{X};\mathbf{Z})$ of three areas are calculated. Among the three areas in Figure 5.11(a), Square 3 has the smallest bathymetry variation. With uniform prior, Square 3 shows the smallest mutual information and largest conditional entropy. In the other hand, Square 1 has the smallest conditional entropy. On average, bathymetry measurements help improve the localization accuracy most for Square 1.

If the prior is Gaussian-distributed and centered at top left (Figure 5.11(b)), Square 2 outperforms Square 1 as the bathymetry variation at the top left is larger for Square 2. Mutual information between the prior and measurement is therefore the most in Square 2 compared with Square 1 and 3. When Gaussian prior is centered at bottom left, Square 2 gives smallest mutual information as the bottom left topology is almost flat.



(a) Conditional entropy of three different areas with uniform prior.



(b) Conditional entropy with different Gaussian priors.

FIGURE 5.11: Information entropy map for different areas: Different prior distributions yield different conditional entropy values.

The examples with different priors show that the effect of bathymetric aids also depends on the priors - how localization information is known before measurements. To be specific, the effect of bathymetric aids depends on how the bathymetry matching helps in improving the prior knowledge of localization. Figure 5.12) shows the conditional entropy maps with different Gaussian priors (plots on the right for each row). For a Gaussian prior, it is good to have more bathymetry variation in the direction of larger uncertainty. For example in the first row of Figure 5.12, Gaussian prior has more uncertainty in the north-south direction, and therefore bathymetry valleys in east west direction give smaller entropy value. A simple extreme case is when an AUV goes along a straight bathymetry valley (V-shape). The lowest point along the path has the same water depth. The valley is steep and therefore the bathymetry variation at the vehicle's left and right sides is high; the high variation bounds the localization error at the two sides. However, the measured water depth along the path does not change (zero variation), and the localization uncertainty in heading directions keeps increasing as every point along the path has the same bathymetry topology around it.

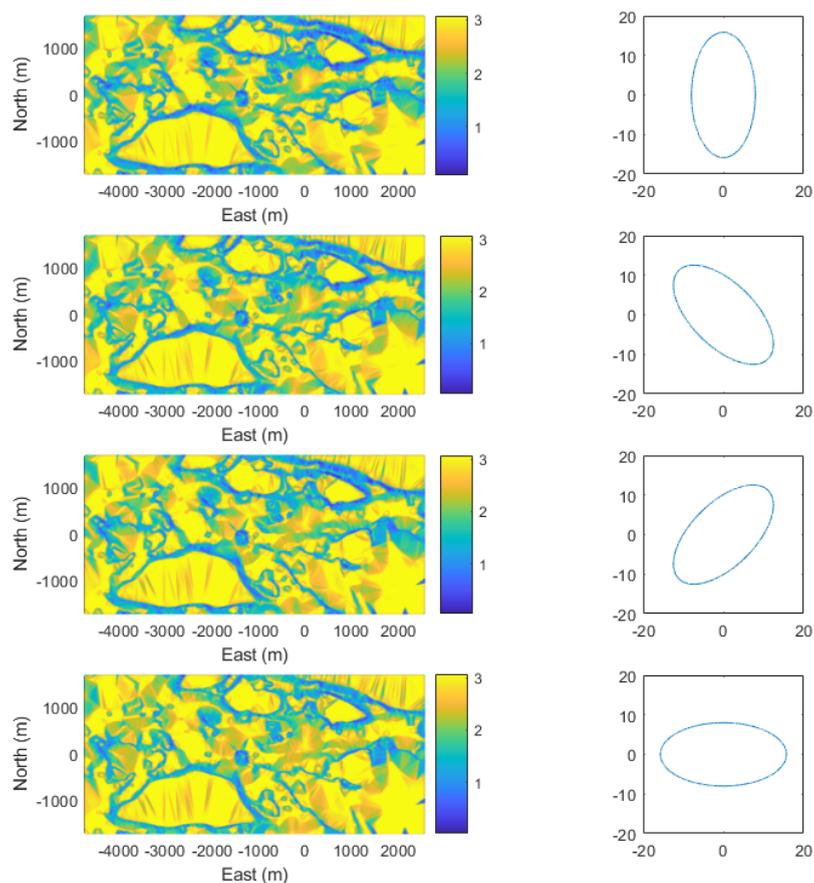


FIGURE 5.12: Information entropy map based on different Gaussian priors: It is good to have more bathymetry variation in the direction where the larger uncertainty of the Gaussian prior lies.

5.4 Summary

We described the localization with bathymetric aids. Nonparametric filters show that localization distribution with bathymetry measurements is neither Gaussian nor uniform. In terms of computational load, accuracy, empirical convergence and sensitivity to various factors, Particle filter (PF)

turns to be the best: it handles multimodal ambiguities, and is not computationally heavy as long as the number of particles is large enough for description of the localization.

Based on particle filters, we formulated PF-based entropy - an information theoretical approach to quantify the localization uncertainty. We built information entropy map to analyze how bathymetry maps benefit localization.

Chapter 6

Navigation with Bathymetric Aids

6.1 Problem Formulation

Chapter 5 has shown with examples, that the localization accuracy strongly depends on the path that an AUV takes, if the AUV uses bathymetric aids for navigation. So how does one select a path that yields good localization? Given a starting point and a destination, we define our problem as how to plan a path such that the localization uncertainty is minimized when vehicle reaches the destination.

The optimal path is found using approximate dynamic programming (ADP) introduced in Section 6.2. The Q-function of the ADP is obtained

by a cycle of reinforcement learning in Section 6.3. The state value of the ADP is obtained by Gaussian process regression (GPR) in Section 6.4. The paths generated are evaluated in Section 6.5. Summary is made in the last.

The work in this chapter and Chapter 5 was published in [71].

6.2 Approximate Dynamic Programming

Rather than appeal to heuristics (bathymetry variation, terrain dispersion, roughness, etc.), we pose the path planning as an optimization problem and solve it by breaking the problem down into a collection of simpler subproblems. This is the concept of dynamic programming. Due to the problem of continuous state domain and “curse of dimensionality”, we approximate the state value and optimize the path in the framework of reinforcement learning and Gaussian process regression.

We define the state S as the positions of particles in a particle filter, that is, $S_k = \{\mathbf{x}_k^i, q_k^i\}$. For simplicity, particles are resampled to have the same weights. Therefore the state space consists of the positions of all particles. Given a starting point and a destination, the path is planned with policy $\pi(S_k)$. The policy which contains a series of actions $\pi(S_k) = a_k, a_{k+1}, a_{k+2}, \dots$. The policy is chosen such that the localization uncertainty is minimized when vehicle reaches the destination. It is

formulated as,

$$\begin{aligned}\pi(S_k) &\leftarrow \arg \min_{a \in \mathcal{A}(S_k)} Q(S_k, a_k) \\ Q(S_k, a_k) &= \sum_{S_{k+1}} p_{\text{tr}}(S_k \xrightarrow{a_k} S_{k+1}) V(S_{k+1}) \\ V(S_k) &= \min_{a_k \in \mathcal{A}(S_k)} Q(S_k, a_k)\end{aligned}\tag{6.1}$$

where $V(\cdot)$ is the value function of a state. $p_{\text{tr}}(S_k \xrightarrow{a_k} S_{k+1})$ is the transition probability from state S_k to state S_{k+1} , due to the non-deterministic evolution of vehicle position by taking action a_k . This planning formulation is essentially an optimization problem. Compared to standard Bellman's equation [65], the transition reward is zero and the discount factor is 1. Therefore, the value of any given state S is the entropy value of state at the destination. In other words, given an optimal path, all states along the same path have the same entropy value.

In the subsequent sections, transition probability $p_{\text{tr}}(S_k \xrightarrow{a_k} S_{k+1})$ is set to 1 during path planning. This is because the performance evaluation of the algorithm is run over Monte Carlo simulations to include the non-deterministic evolution of vehicle position.

Equation (6.1) suffers from “the curse of dimensionality”. The state space depends on the number of particles and the action is in a continuous domain. It is not possible to use dynamic programming to solve the sequential decision process problem. We resort to the techniques of

approximate dynamic programming (ADP) in [65] and solve the optimization problem using reinforcement learning and Gaussian process regression (GPR). Q-function [79] $Q(\cdot)$ across all possible actions is obtained by a cycle of reinforcement learning - policy generation and evaluation.

6.3 Iterative Path Planning Algorithm

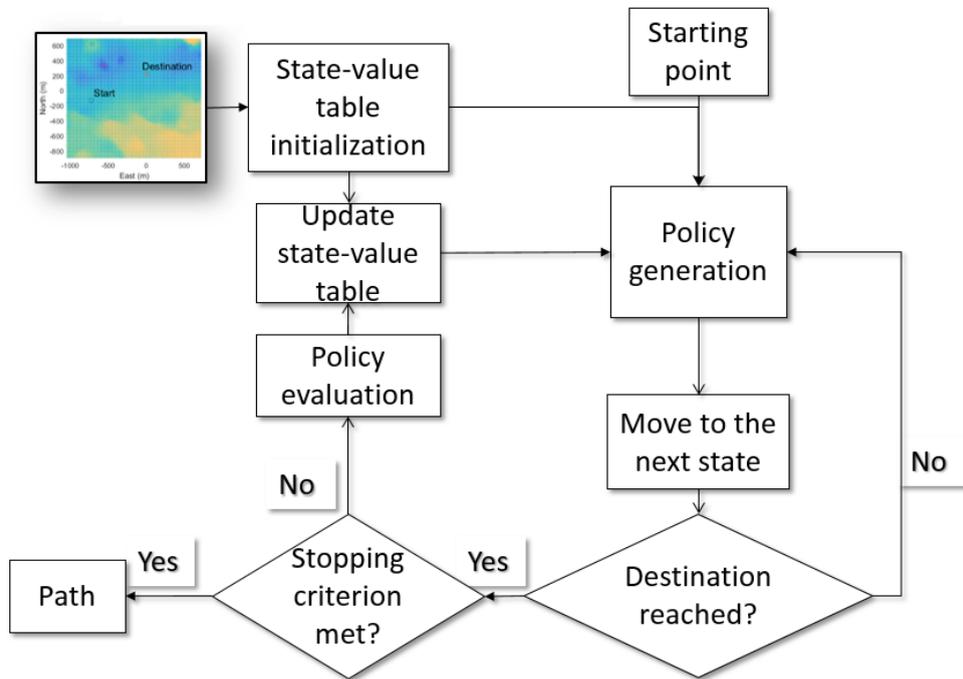


FIGURE 6.1: Flowchart of iterative path planning algorithm.

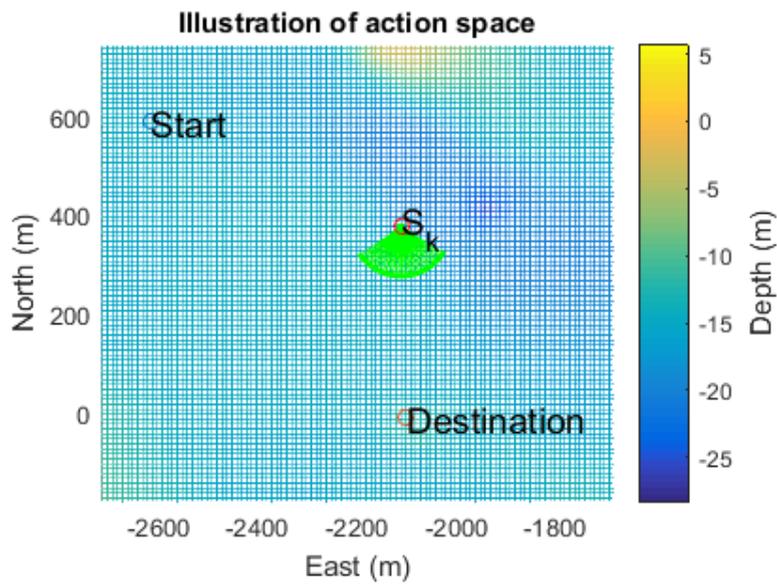
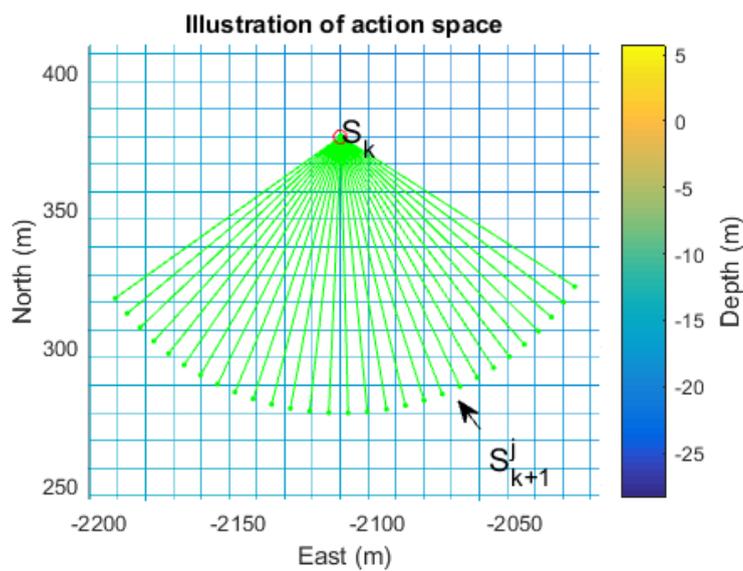
Figure 6.1 shows the flowchart of the iterative path planning algorithm. Each iteration consists of two major steps - policy generation and policy evaluation. Every time a path is generated, localization along the path is simulated to enforce the learning of the path values.

The algorithm starts with randomly generated paths, given the starting point and destination. The localization along the paths is simulated to get the estimated values $V^*(S)$. A state-value table is constructed with each entry recording a state S and corresponding value $V^*(S)$. In each iteration, the policy is generated according to the estimated state value from the table. At the end of each iteration, a path is generated and is then evaluated from simulation. New state-value entries are used to update the state-value table. The state-value table is refined over iterations.

6.3.1 Policy Generation

Given any state S_k , we approximate the continuous space using a discrete system. We form an action space $\mathcal{A}(a_k)$ containing all possible actions. The possible actions are the headings linearly spaced within $(-\frac{\pi}{4}, \frac{\pi}{4})$ when vehicle heads towards the destination (Figure 6.2(a)). To cover all the bathymetry grids, the resulting positions after $\tau = 100$ time steps need to be roughly 10 meters apart from each other. Therefore, there are approximately 29 actions in the action space.

A resulting state is generated for each action such that $(S_k, a_k) \rightarrow S_{k+1}$ (Figure 6.2(b)). The state value $V(S_{k+1})$ is estimated using Gaussian process regression (GPR) [66]. We will explain the detailed GPR in the next section. The policy is updated with the action that leads to the next

(a) Given any state S_k , an action space is generated.

(b) A zoom-in plot of the action space and resulting state.

FIGURE 6.2: Action space for policy generation: The next waypoints from action set include all possible map grids when looking at the destination.

state with minimum value. If the current position is within τ -step moves to the destination, we navigate the vehicle to the destination and the path planning is completed.

6.3.2 Policy Evaluation

After the path is generated, we evaluate the path by simulating the localization along the planned path. The navigation follows waypoints generated along the path at τ time steps apart. To follow the waypoints, the vehicle compares its estimated position with the targeted waypoint, and generates control and command accordingly. The details of path execution are presented in Section 6.5. The states along the path have the same value as the state when vehicle reaches the destination. The path is evaluated at the median value over Monte Carlo simulations. This minimizes the discretization error due to the limited number of particles. The new state-value entries are added to the state-value table if their values are smaller than the values of the nearby states. We also remove the nearby states with large values. Instead of conventional Euclidean distance, the distance between state is evaluated using Bhattacharyya Coefficient [27] and is explained in details in Section 6.4.

6.3.3 The Policy Iteration Algorithm

The algorithm is summarized below:

Algorithm 1 Policy iteration

Randomly generate a number of paths (e.g. 500) and construct the state-value table

repeat

 Start from starting point

while *The destination is not reached* **do**

 Generate action space

for *each action in the action space* **do**

 | Estimate the value based on state-value table

end

 Choose the action with the minimum value and move

end

 Re-evaluate the value of the generated path

 Update the state-value table

until *Stabilized*;

It should be noted that this path planning algorithm plans path offline to generate the state-value table. When AUV executes the path, a new planned path can be generated if AUV detects itself away from the planned path or carrying a new state in the localization filter. The refined state-value table is used to generate the path. However, if the state or estimated position is far away from the ones recorded in the state-value table, the iterative path planning has to start over to optimize the state-value table.

6.4 Gaussian Process Regression for Path Planning

In the section of policy generation, the value function $V(S)$ (the time step subscript is dropped for simplicity of notation) of a state S is modeled as a Gaussian process. The reason is that we only have limited number of available data (the state-value table) to estimate the continuous value in the high-dimensional state space (The state is in dimension of 2×6000 where 6000 is the number of the particles). Therefore we perceive the states with jointly Gaussian distribution, and infer the state value in a continuous space with a Gaussian process prior.

To estimate $V(S)$, we choose some nearby states with smaller values. For example, we only use the nearby states whose values are below the 75th percentile. This is because we know that the available value of the states in the state-value table is larger than the optimal value as the paths are not optimal. We purposely remove some state-value entries as it is obviously not the optimal one. Let the chosen states and their values be $\mathcal{D} = \{(T_i, v_{T,i})\}$. The value function $V(\cdot)$ can be inferred using Bayes Theorem:

$$p(V(\cdot)|\mathcal{D}) = \frac{p(V(\cdot))p(\mathcal{D}|V(\cdot))}{p(\mathcal{D})}. \quad (6.2)$$

The values are observations on the value function $V(T)$, which is a Gaussian process (GP). We have

$$\begin{aligned} v_{T,i} &= V(T_i) + \epsilon_i \\ V &\sim \mathbf{GP}(\cdot|0, \mathbf{K}) \\ \epsilon_i &\sim (N)(\cdot|0, \sigma^2). \end{aligned} \tag{6.3}$$

\mathbf{K} is kernel function defined by the covariance of the states. The prior on $V(\cdot)$ is a GP and likelihood is Gaussian. Therefore the posterior on $V(\cdot)$ is also a GP. We can make predictions on new state S

$$p(v_S|S, \mathcal{D}) = \int p(v_S|S, V(\cdot), \mathcal{D})p(V(\cdot)|\mathcal{D})dV(\cdot). \tag{6.4}$$

Figure 6.3 illustrates GPR state estimation in an 1-dimensional example. The value is to be estimated at the location indicated by the vertical blue line. The red dash-dot line is the optimal value to be estimated. Initialization generates paths with state values larger than value of the optimal path. Therefore, nearby states with smaller values are used to estimate the state value. The state value is estimated as shown by the blue curve.

The distance between states in the illustration (Figure 6.3) is simply the distance in the x-axis. In our path planning problem, the state variable consists of positions of all particles. The state space is much larger. We cannot simply use a Euclidean distance to measure the similarity between states. To evaluate the distance between states S and T , the Bhattacharyya

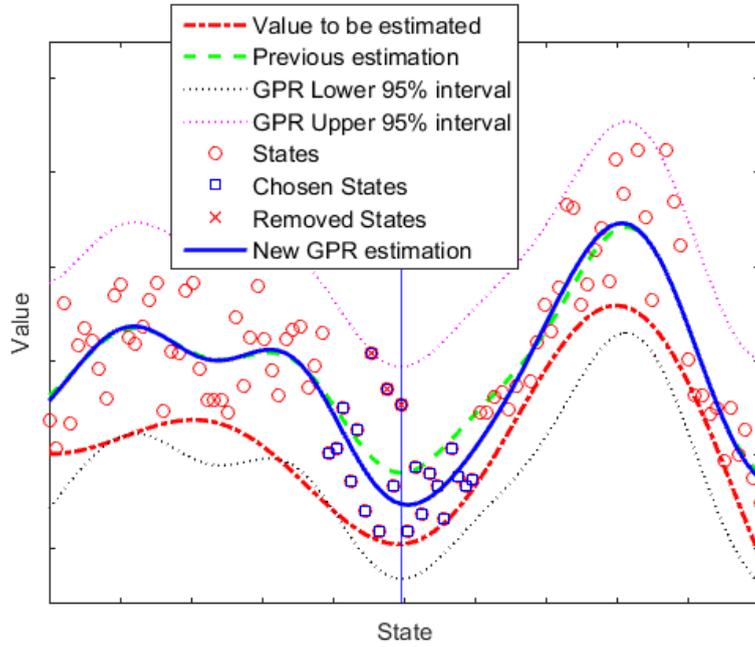


FIGURE 6.3: Illustration of Gaussian process regression in 1-dimensional space.

Coefficient $\rho(S, T)$ [27] is used. Bhattacharyya Coefficient $\rho(S, T)$ measures the level of overlapping between two distributions and therefore is suitable to describe the similarity of the states (sets of particles). Let the discrete densities of particle sets S and T be $\{\hat{s}_u\}_{u=1, \dots, m}$ and $\{\hat{t}_u\}_{u=1, \dots, m}$ respectively, where m is the number of bins and $\sum_{u=1}^m \hat{s}_u = 1, \sum_{t=1}^m \hat{t}_u = 1$. We have $\rho(S, T) = \sum_{u=1}^m \sqrt{\hat{s}_u \hat{t}_u}$. The distance $\mathcal{B}(S, T)$ between state S and state T is

$$\mathcal{B}(S, T) = \sqrt{1 - \rho(S, T)}. \quad (6.5)$$

6.5 Simulation and Performance

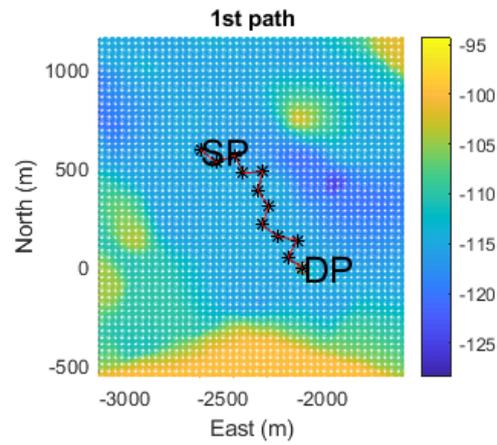
6.5.1 Underwater Vehicle Navigation

Navigation is the activity of ascertaining one's position, planning and following a route. To evaluate how the path planning benefits localization, the vehicle needs to follow the planned path. A series of waypoints are sampled from the planned path, with the destination as the last waypoint. The vehicle compares its estimated position $\hat{\mathbf{x}}_{\mathbf{k}}$ with the targeted waypoint, and gives an action that directs the vehicle to head towards the targeted waypoint. We set a 10-meter range to determine whether vehicle has reached the waypoint. Once $\hat{\mathbf{x}}_{\mathbf{k}}$ is within 10 meters of the waypoint, the vehicle changes to target the subsequent waypoint. If the vehicle reaches a later waypoint before the current one, it continues to follow the next waypoint. The mission ends when vehicle reaches the destination (within a 10-meter range) or the mission exceeds the maximum allowable duration.

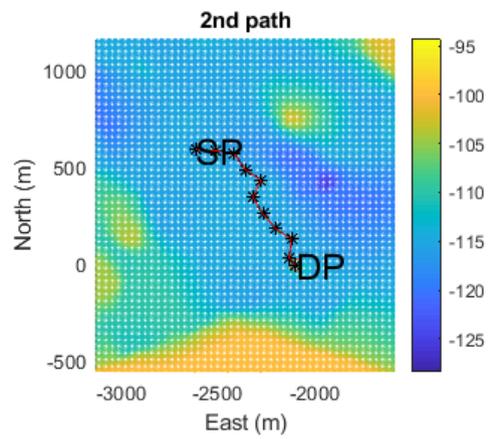
Our bathymetry map has a resolution of 10 meters. The vehicle makes a measurement at every 10 seconds when moving at 1 meter per second.

6.5.2 Mission 1

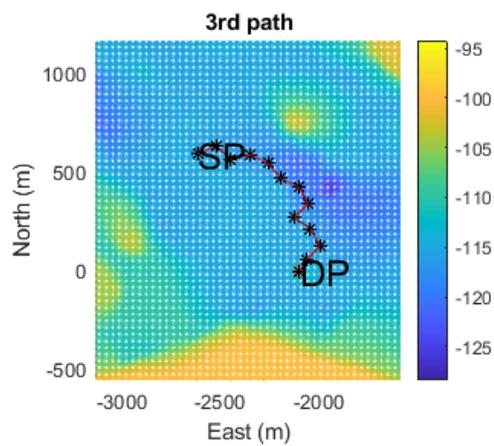
We tested our algorithm on bathymetry data collected at a test location in Singapore waters. With the starting point (SP) and destination point



(a) Iteration 1

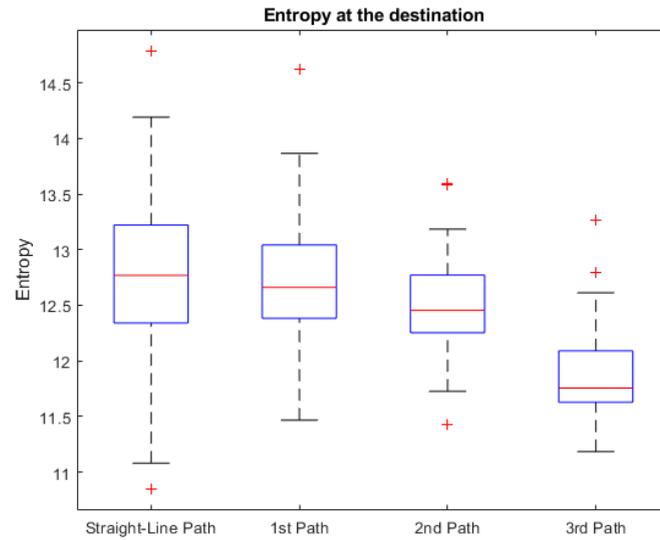


(b) Iteration 2

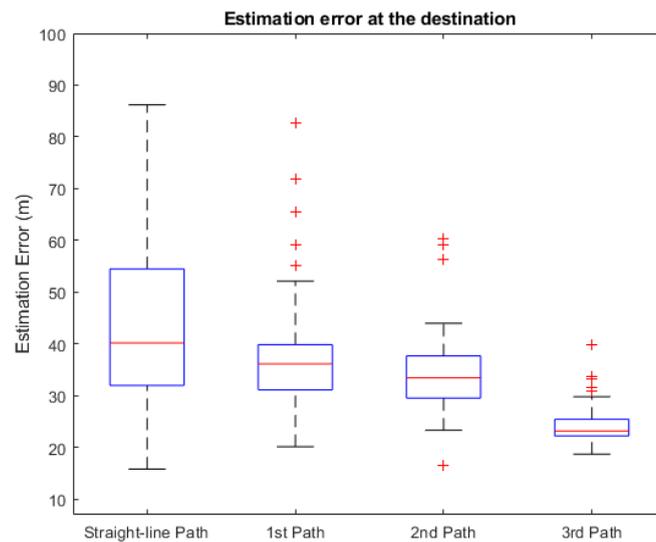


(c) Iteration 3

FIGURE 6.4: Mission 1: As the algorithm iterates, the planned path evolves to one with more bathymetric variation.



(a) PF-based entropy at the destination: the value drops with iterations, and is also smaller than the entropy of a straight-line path.



(b) Estimation error at the destination: the localization error drops with iteration, and is smaller than the error of a straight-line path.

FIGURE 6.5: Mission 1: Performance at the destination.

(DP) being defined, Figure 6.4 shows that as the algorithm iterates, the planned path evolves to one with more bathymetric variation.

The navigation accuracy is estimated with 50 simulated runs. The entropy at the destination (Figure 6.5(a)) drops with iterations, and is smaller compared with the entropy at the end of a straight-line path. The localization errors at the destination are shown in Figure 6.5(b) for straight-line path and generated paths over iterations 1 to 3. With the same propagation and observation capability, routes through more bathymetric variation have better localization accuracy. A good path is generated within a few iterations.

6.5.3 Mission 2

We test another pair of starting and destination points. In contrast to the pair in Mission 1, this pair has a small bathymetric basin (dark blue region) between them. In the first three iterations in Figure 6.6, paths are generated along the southwest side of the basin. From the fourth iteration, the generated path starts to move to the other side of the basin, and comes back to the southwest side. The PF-based entropy values and localization errors at the destination drop over iterations and have similar trend (Figure 6.7). Looking at the bathymetry along the planned path in Figure 6.8, we can see that paths from later iterations do have larger range in bathymetry along their paths. However Path 4 with largest bathymetry range does not give smallest localization error.

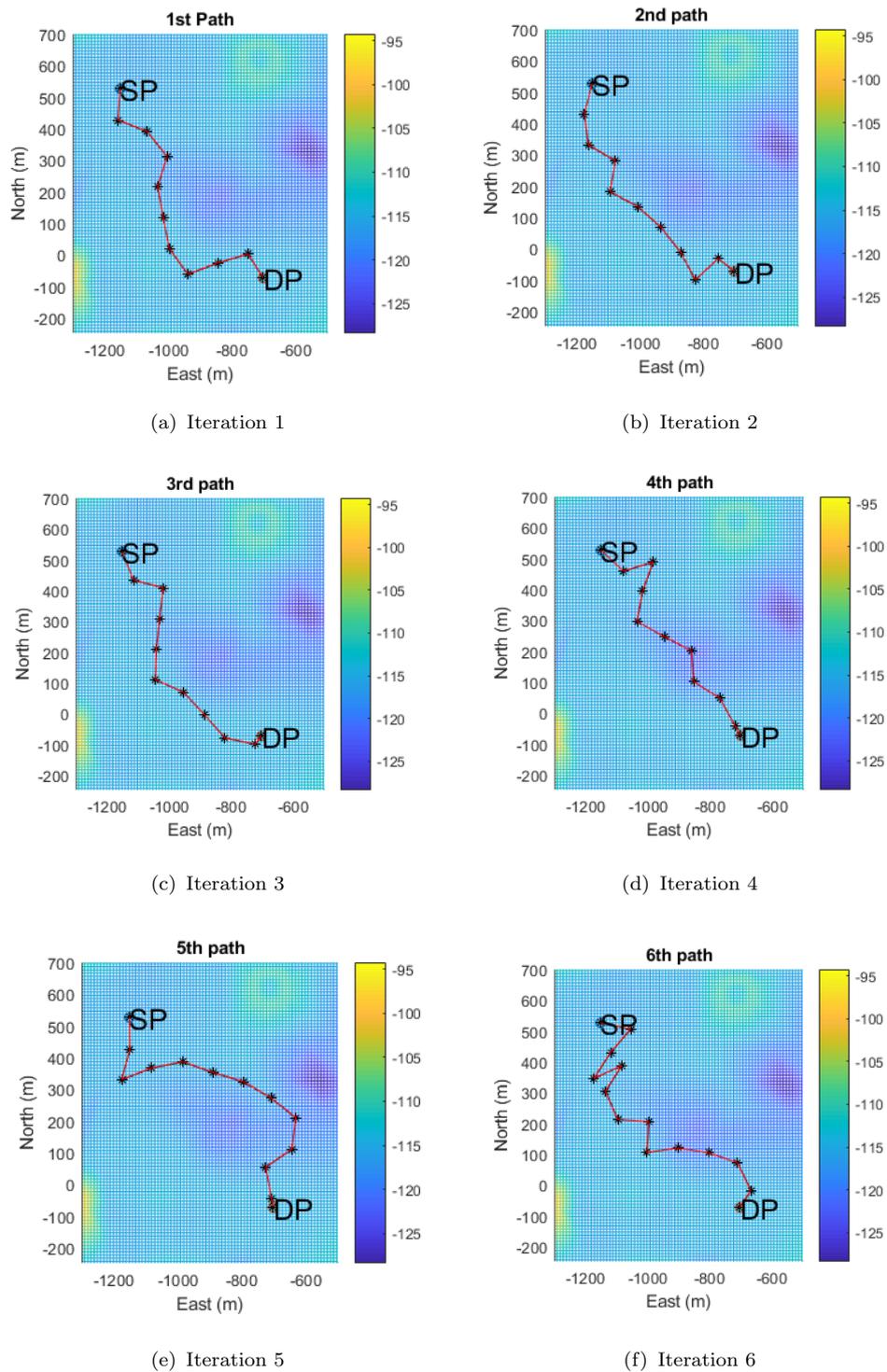
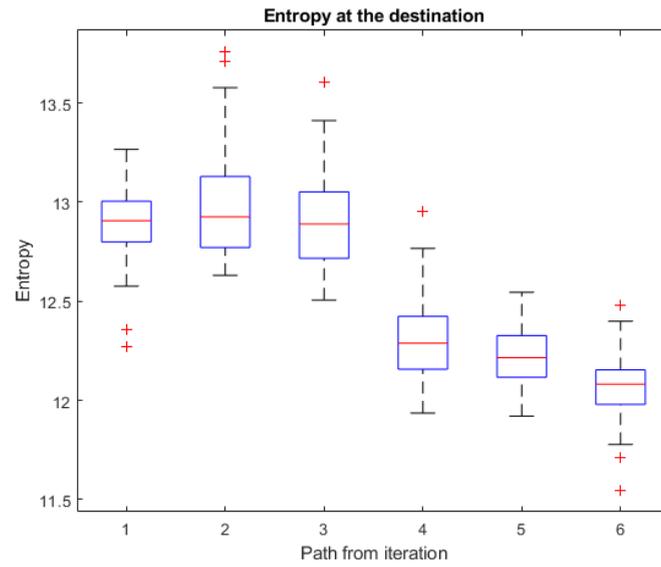
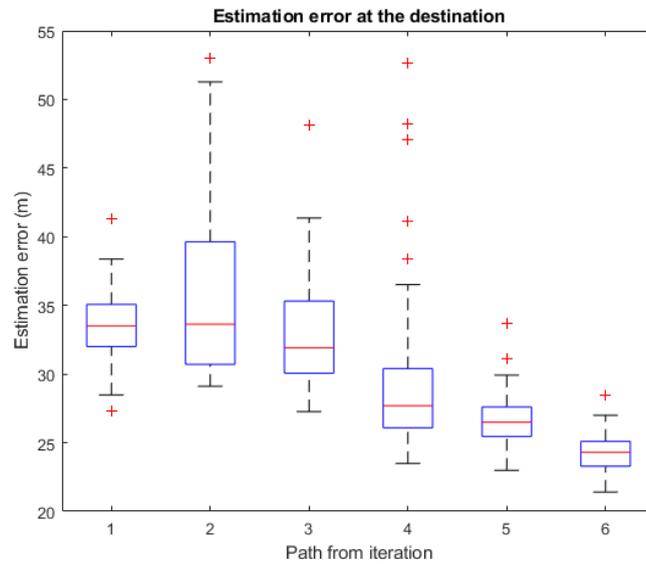


FIGURE 6.6: Mission 2: In the first three iterations, planned path evolves to one side of the bathymetry basin. From the fourth iteration, the planned path evolves to the other side of the basin.



(a) PF-based entropy at the destination: The paths following the basin edge with more bathymetric variation do not yield smallest entropy values. Entropy values drops over iterations.



(b) Estimation error at the destination: The localization errors drops over iterations. However the bathymetric variations along paths over iterations do not increase.

FIGURE 6.7: Mission 2: Performance at the destination.

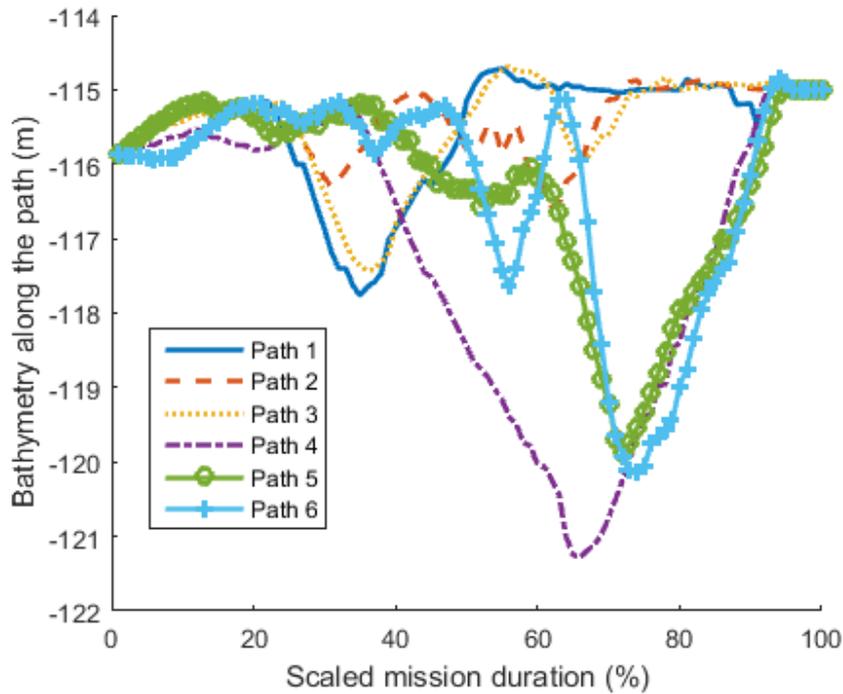


FIGURE 6.8: Illustration of Gaussian process regression in 1-dimensional space.

6.6 Summary

With the particle filter based entropy and the information entropy introduced in Chapter 5, we presented a path planning algorithm. Given a starting point and destination, the algorithm generates a sub-optimal path of fine, such that the localization accuracy is maximized when vehicle reaches the destination. We showed that the entropy measure is consistent with the localization performance. It is important to highlight that there are many definitions on the bathymetry variation. It can be the change of bathymetry (single-point water depth) along the path, or the bathymetry range within the path (difference between maximum and minimum water depth). It can

also be the local bathymetry variation (over some area) along the path. The last definition needs to define an area size to compute the variation. For any of the definitions, paths along maximum bathymetric variation may not always lead to the smallest positioning error. The bathymetry matching performance depends on the prior knowledge about the location and the exact bathymetry, specifically, how unique the measured bathymetry is compared with the others in the prior. The conditional entropy measure evaluates this performance and shows consistent trend with the localization performance. Path planning with the information entropy measure could be extended to other planning problem and eventually cooperative localization of small team of AUVs.

Chapter 7

Conclusions and Future Work

A list of contributions in this thesis is summarized as:

- We focus on a cooperating team of small-sized, low-cost, sensor-limited AUVs. We showed that the cooperation improves localization but also easily aggravates the performance when communication loss is higher.
- We proposed a new cooperative multi-vehicle localization algorithm using distributed extended information filter (DEIF). It is effective in recording the correlated information in light of constrained underwater communication. Simulations show that DEIF gives better performance compared with single-vehicle localization and existing cooperative localization method.

- As *naïve* filter is easy to implement in complex situation, we answer the question as to when it is safe or detrimental to ignore the correlation in cooperation.
- We formalize the concept of information entropy measure, to quantify the localization performance, and the effectiveness of bathymetry on localization. We concluded that the uniqueness of the measured depth compared with the bathymetry in the localization prior decides the localization performance.
- With the conclusion above, we proposed a path planning algorithm for navigation with bathymetric aids. The algorithm generates near-optimal paths based on bathymetry map, with good localization accuracy at the destination.

In cooperative localization of underwater vehicles, one cannot assume that the inter-vehicle correlations are available. This is because the constrained underwater communications prevent vehicles from keeping track of all the estimates shared in the team. It may hamper the cooperation as the fused information might be false or overconfident when correlation is underestimated. We examined the problem and proposed a novel design of the distributed localization method. The method is able to record the correlated information from the most recent cooperation and transmit with

small packets, providing consistent position estimates in event of packet loss.

However, ignoring correlation when fusing data seems working in some work. This dissertation studied the conditions and provided the justification where the correlation can be ignored. For multi-sensor tracking problem, the condition is when the local measurements have relatively small errors such that the local estimates are nearly independent of each other. For cooperative localization problem, besides having small local measurement errors which validates the assumption of independence, the other condition is where the local propagation is long enough such that the local estimates are nearly independent.

With the assurance of the small local measurement errors, cooperative localization with bathymetric aids can simply ignore the correlation. We explored the bathymetry-aided localization and navigation of a single vehicle. This can be extended to cooperative vehicles without the concerns of tracking inter-vehicle correlation. The effectiveness of bathymetry aids on localization is shown with an information entropy measure. It showed that the localization improvement depends on how unique the bathymetry map against a prior knowledge about the localization. With this idea, a path planning algorithm is developed for a particle filtering localization. Simulations show that the algorithm generates sub-optimal paths within a few iterations.

This research is open in applicability to other areas related to cooperative positioning and navigation. For example, terrestrial swarm robotics on land is one very popular topic where this research can be of relevance. Mobile and distributed sensor networks have the potential to revolutionize the way in which information is collected, fused and disseminated. It is closely related to distributed data fusion network [42], which attracts interest in many areas with its advantage of scalability, modularity and graceful degradation of performance in case of failures.

There are a number of avenues for future work. Firstly, the path planning with information entropy measure should be used with different problem set up. A good localization along the path is also important in surveying and environment sensing. Secondly, there is opportunity to extend the planning by introducing the presence of peer vehicles. For example, the goal can be changed to optimize the localization performance of the whole team or a particular vehicle, or to maximize the communication quality for information exchange. There could be some spatial correlation in the measurements among the team, for example, due to the tidal changes. The cooperative AUVs can be used for bathymetry map building. With knowledge on partial bathymetry map, the paths can be planned adaptively, and a fuller map can be developed from there.

Bibliography

- [1] M. Ahmed and G. Pottie, *Fusion in the Context of Information Theory*, S. S. Iyengar and R. R. Brooks, Eds. CRC Press, 2004.
- [2] A. Alcocer, P. Oliveira, and A. Pascoal, “Underwater acoustic positioning systems based on buoys with gps,” in *Proceedings of the Eighth European Conference on Underwater Acoustics*, vol. 8, 2006, pp. 1–8.
- [3] J. C. Alleyne, “Position estimation from range only measurements,” Master’s thesis, Naval Postgrad. Schl., Monterey, CA, USA, Sep. 2000.
- [4] D. Alspach and H. Sorenson, “Nonlinear bayesian estimation using gaussian sum approximations,” *Automatic Control, IEEE Transactions on*, vol. 17, no. 4, pp. 439–448, Aug 1972.
- [5] G. Antonelli, F. Arrichiello, S. Chiaverini, and G. Sukhatme, “Observability analysis of relative localization for auvs based on ranging and depth measurements,” in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2010, pp. 4276–4281.
- [6] M. Aoki, *Optimization of Stochastic Systems, Topics in Discrete-Time Systems*. Academic Press, New York, 1967.
- [7] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking,” *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb 2002.
- [8] P. Baccou, B. Jouvencel, V. Creuze, and C. Rabaud, “Cooperative positioning and navigation for multiple auv operations,” in *MTS/IEEE*

- Oceans 2001. An Ocean Odyssey. Conference Proceedings (IEEE Cat. No.01CH37295)*, vol. 3, Nov 2001, pp. 1816–1821 vol.3.
- [9] A. Bahr, “Cooperative localization for autonomous underwater vehicles,” Ph.D. dissertation, the Massachusetts Institute of Technology and the Woods Hole Oceanographic Institution, February 2009.
- [10] A. Bahr, J. Leonard, and A. Martinoli, “Dynamic positioning of beacon vehicles for cooperative underwater navigation,” in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, Oct 2012, pp. 3760–3767.
- [11] Y. Bar-Shalom, “Comments on ”comparison of two-sensor tracking methods based on state vector fusion and measurement fusion” by j. roecker et al,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 24, no. 4, pp. 456–457, July 1988.
- [12] ———, *Multitarget-Multisensor Tracking: Principles And Techniques*, 3rd ed. YBS Publishing, 1995.
- [13] Y. Bar-Shalom and L. Campo, “The effect of the common process noise on the two-sensor fused-track covariance,” *Aerospace and Electronic Systems, IEEE Transactions on*, vol. AES-22, no. 6, pp. 803–805, Nov 1986.
- [14] S. Barkby, S. B. Williams, O. Pizarro, and M. V. Jakuba, “Bathymetric slam with no map overlap using gaussian processes,” in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2011, pp. 1242–1248.
- [15] A. R. Benaskeur, “Consistent fusion of correlated data sources,” in *IEEE 2002 28th Annual Conference of the Industrial Electronics Society. IECON 02*, vol. 4, Nov 2002, pp. 2652–2656 vol.4.
- [16] C. Bishop, *Pattern Recognition and Machine Learning*. Springer-Verlag New York, 2006.

- [17] Y. Boers, H. Driessen, A. Bagchi, and P. Mandal, “Particle filter based entropy,” in *2010 13th International Conference on Information Fusion*, July 2010, pp. 1–8.
- [18] N. Bore, I. Torroba, and J. Folkesson, “Sparse gaussian process slam, storage and filtering for auv multibeam bathymetry,” in *2018 IEEE/OES Autonomous Underwater Vehicles (AUV 2018)*, (Porto, Portugal), Nov 2018.
- [19] Z. I. Botev, J. F. Grotowski, and D. P. Kroese, “Kernel density estimation via diffusion,” *Ann. Statist.*, vol. 38, no. 5, pp. 2916–2957, 10 2010. [Online]. Available: <https://doi.org/10.1214/10-AOS799>
- [20] J. Busquets, J. V. Busquets, D. Tudela, F. Pérez, J. Busquets-Carbonell, A. Barberá, C. Rodríguez, A. J. García, and J. Gilabert, “Low-cost AUV based on arduino open source microcontroller board for oceanographic research applications in a collaborative long term deployment missions and suitable for combining with an usv as autonomous automatic recharging platform,” in *2012 IEEE/OES Autonomous Underwater Vehicles (AUV)*, Sept 2012, pp. 1–10.
- [21] J. Byron and R. Tyce, “Designing a vertical / horizontal AUV for deep ocean sampling,” in *OCEANS 2007*, Sept 2007, pp. 1–10.
- [22] S. Carreño, P. Wilson, P. Ridaio, and Y. Petillot, “A survey on terrain based navigation for auvs,” in *OCEANS 2010 MTS/IEEE SEATTLE*, Sept 2010, pp. 1–7.
- [23] K. C. Chang, R. K. Saha, and Y. Bar-Shalom, “On optimal track-to-track fusion,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 4, pp. 1271–1276, Oct 1997.
- [24] L. Chen, P. Arambel, and R. Mehra, “Fusion under unknown correlation - covariance intersection as a special case,” in *Information Fusion, 2002. Proceedings of the Fifth International Conference on*, vol. 2, July 2002, pp. 905–912 vol.2.

- [25] M. Chitre, "Path planning for cooperative underwater range-only navigation using a single beacon," in *2010 International Conference on Autonomous and Intelligent Systems, AIS 2010*, June 2010, pp. 1–6.
- [26] R. Chou, Y. Boers, M. Podt, and M. Geist, "Performance evaluation for particle filters," in *14th International Conference on Information Fusion*, July 2011, pp. 1–7.
- [27] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, vol. 2, 2000, pp. 142–149 vol.2.
- [28] J. Curcio, J. Leonard, J. Vaganay, A. Patrikalakis, A. Bahr, D. Battle, H. Schmidt, and M. Grund, "Experiments in moving baseline navigation using autonomous surface craft," in *Proceedings of OCEANS 2005 MTS/IEEE*, Sept 2005, pp. 730–735 Vol. 1.
- [29] A. Czirók and T. Vicsek, "Collective behavior of interacting self-propelled particles," *Physica A: Statistical Mechanics and its Applications*, vol. 281, no. 1, pp. 17 – 29, 2000. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0378437100000133>
- [30] P. D. Deshpande, M. N. Sangekar, B. Kalyan, M. A. Chitre, S. Shahabudeen, V. Pallayil, and T. B. Koay, "Design and development of auvs for cooperative missions," in *Defence Technology Asia, Singapore*, March 22 2007.
- [31] R. M. Eustice, H. Singh, and J. J. Leonard, "Exactly sparse delayed-state filters for view-based slam," *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1100–1114, Dec 2006.
- [32] D. Fairbair, "Using entropy to assess the efficiency of terrain representation," in *25th International Cartographic Conference. 2011, Paris, France*, 2011.
- [33] M. F. Fallon, G. Papadopoulos, J. J. Leonard, and N. M. Patrikalakis, "Cooperative AUV navigation using a single maneuvering surface

- craft,” *I. J. Robotics Res.*, vol. 29, no. 12, pp. 1461–1474, 2010. [Online]. Available: <https://doi.org/10.1177/0278364910380760>
- [34] D. Fox, W. Burgard, and S. Thrun, “Markov localization for mobile robots in dynamic environments,” *Journal of Artificial Intelligence Research*, 1999.
- [35] R. P. Franca, A. T. S. and Rafael D. S. Castro, B. N. Green, and D. Marelli, “Trajectory generation for bathymetry based auv navigation and localization,” in *10th IFAC Conference on Manoeuvring and Control of Marine Craft MCMC 2015: Copenhagen*, vol. 48, 24–26 August 2015.
- [36] A. S. Gadre and D. J. Stilwell, “Toward underwater navigation based on range measurements from a single location,” in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 5, April 2004, pp. 4472–4477 Vol.5.
- [37] E. Galceran, S. Nagappa, M. Carreras, P. Ridao, and A. Palomer, “Uncertainty-driven survey path planning for bathymetric mapping,” in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, Nov 2013, pp. 6006–6012.
- [38] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice*, 2nd ed. New York, USA: Wiley, 2000.
- [39] S. Grime and H. Durrant-Whyte, “Data fusion in decentralized sensor networks,” *Control Engineering Practice*, vol. 2, no. 5, pp. 849 – 863, 1994. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0967066194903492>
- [40] U. Hanebeck, K. Briechle, and J. Horn, “A tight bound for the joint covariance of two random vectors with unknown but constrained cross-correlation,” in *Multisensor Fusion and Integration for Intelligent Systems, 2001. MFI 2001. International Conference on*, 2001, pp. 85–90.
- [41] G. Indiveri, “Geotechnical surveys with cooperative autonomous marine vehicles: the ec wimust project,” in *2018 IEEE/OES Autonomous Underwater Vehicles (AUV 2018), (Porto, Portugal)*, Nov 2018.

- [42] S. J. Julier and J. K. Uhlmann, “A non-divergent estimation algorithm in the presence of unknown correlations,” in *American Control Conference, 1997. Proceedings of the 1997*, vol. 4, 1997, pp. 2369–2373 vol.4.
- [43] —, “General decentralized data fusion with covariance intersection (CI),” in *Handbook of Multisensor Data Fusion*. CRC Press LLC, 2001, ch. 12.
- [44] —, “Simultaneous localisation and map building using split covariance intersection,” in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, vol. 3, 2001, pp. 1257–1262 vol.3.
- [45] —, “Unscented filtering and nonlinear estimation,” *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, March 2004.
- [46] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Journal of Fluids Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
- [47] B. Kalyan and M. Chitre, “A feasibility analysis on using bathymetry for navigation of autonomous underwater vehicles,” in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, ser. SAC '13. New York, NY, USA: ACM, 2013, pp. 229–231. [Online]. Available: <http://doi.acm.org/10.1145/2480362.2480411>
- [48] H. Kaprykowsky and X. Rodet, “Globally optimal short-time dynamic time warping application to score to audio alignment,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, 2006, pp. 249–252.
- [49] T. B. Koay, Y. T. Tan, Y. H. Eng, R. Gao, M. Chitre, J. L. Chew, N. Chandhavarkar, R. Khan, T. Taher, and J. Koh, “Starfish – a small team of autonomous robotic fish,” *Indian Journal of geo-Marine Science*, vol. 40, no. 2, 2011.

-
- [50] J. Kojima, Y. Kato, K. Asakawa, S. Matumoto, S. Takagi, and N. Kato, "Development of autonomous underwater vehicle 'AQUA EXPLORER 2' for inspection of underwater cables," in *Oceans '97. MTS/IEEE Conference Proceedings*, vol. 2, Oct 1997, pp. 1007–1012 vol.2.
- [51] H. Li, F. Nashashibi, and M. Yang, "Split covariance intersection filter: Theory and its application to vehicle localization," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 14, no. 4, pp. 1860–1871, 2013.
- [52] H. Li and F. Nashashibi, "Cooperative multi-vehicle localization using split covariance intersection filter," *Intelligent Transportation Systems Magazine, IEEE*, vol. 5, no. 2, pp. 33–44, 2013.
- [53] J. Liu, Z. Wang, Z. Peng, J. Cui, and L. Fiondella, "Suave: Swarm underwater autonomous vehicle localization," in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, April 2014, pp. 64–72.
- [54] A. Martins, J. M. Almeida, and E. Silva, "Coordinated maneuver for gradient search using multiple auvs," in *Oceans 2003. Celebrating the Past ... Teaming Toward the Future (IEEE Cat. No.03CH37492)*, vol. 1, Sept 2003, pp. 347–352 Vol.1.
- [55] D. K. Meduna, S. M. Rock, and R. S. McEwen, "Closed-loop terrain relative navigation for auvs with non-inertial grade navigation sensors," in *2010 IEEE/OES Autonomous Underwater Vehicles*, Sept 2010, pp. 1–8.
- [56] D. Meduna, S. Rock, and R. McEwen, "Closed-loop terrain relative navigation for auvs with non-inertial grade navigation sensors," in *Autonomous Underwater Vehicles (AUV), 2010 IEEE/OES*, Sept 2010, pp. 1–8.
- [57] P. Milne, *Underwater Acoustic Positioning Systems*. Gulf Publishing Company, 1983.

- [58] S. C. Nair, E. M. Coronado, M. T. Frye, and Y. Qin, “Swarm intelligence for the control of a group of robots,” in *2015 10th System of Systems Engineering Conference (SoSE)*, May 2015, pp. 205–207.
- [59] C. L. Ng, Q. Q. Chen, J. J. Chua, and H. F. Hemond, “A multi-platform optical sensor for in vivo and in vitro algae classification,” *Sensors (Basel)*, vol. 17, no. 4, 2017.
- [60] M. K. Nighot, V. H. Patil, and G. S. Mani, “Multi-robot hunting based on swarm intelligence,” in *2012 12th International Conference on Hybrid Intelligent Systems (HIS)*, Dec 2012, pp. 203–206.
- [61] D. R. Parhi, J. K. Pothal, and M. K. Singh, “Navigation of multiple mobile robots using swarm intelligence,” in *2009 World Congress on Nature Biologically Inspired Computing (NaBIC)*, Dec 2009, pp. 1145–1149.
- [62] D. Peng, T. Zhou, H. Li, and W. Zhang, “Terrain aided navigation for underwater vehicles using maximum likelihood method,” in *2016 IEEE/OES China Ocean Acoustics (COA)*, Jan 2016, pp. 1–6.
- [63] A. Phillips, D. Fenucci, A. Munafò, J. Neasham, N. Gold, J. Sitbon, I. Vincent, and T. Sloane, “Development of smart networks for navigation in dynamic underwater environments,” in *2018 IEEE/OES Autonomous Underwater Vehicles (AUV 2018)*, (Porto, Portugal), Nov 2018.
- [64] A. Phillips, G. Salavasidis, M. Kingsland, C. Harris, M. Pebody, D. Roper, R. Templeton, S. McPhail, T. Prampart, T. Wood, R. Taylor, and T. Jones, “Autonomous surface/subsurface survey system field trials,” in *2018 IEEE/OES Autonomous Underwater Vehicles (AUV 2018)*, (Porto, Portugal), Nov 2018.
- [65] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality (Wiley Series in Probability and Statistics)*. New York, NY, USA: Wiley-Interscience, 2007.

- [66] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.
- [67] S. Reece and S. Roberts, “Robust, low-bandwidth, multi-vehicle mapping,” in *Information Fusion, 2005 8th International Conference on*, vol. 2, July 2005, pp. 8 pp.–.
- [68] S. I. Roumeliotis and G. A. Bekey, “Collective localization: A distributed kalman filter approach to localization of groups of mobile robots,” in *the 2000 IEEE International Conference on Robotics & Automation, San Francisco, CA*, April 2000.
- [69] G. Rui and M. Chitre, “Cooperative multi-aUV localization using distributed extended information filter,” in *2016 IEEE/OES Autonomous Underwater Vehicles (AUV)*, Nov 2016, pp. 206–212.
- [70] —, “On distributed processing for underwater cooperative localization,” in *2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, June 2017, pp. 507–511.
- [71] —, “Path planning for bathymetry-aided underwater navigation,” in *2018 IEEE/OES Autonomous Underwater Vehicles (AUV 2018), (Porto, Portugal)*, Nov 2018.
- [72] —, “Cooperative positioning using range-only measurements between two auvs,” in *OCEANS’10 IEEE SYDNEY*, May 2010, pp. 1–6.
- [73] G. Salavasidis, C. Harris, S. McPhail, A. B. Phillips, and E. Rogers, “Terrain aided navigation for long range auv operations at arctic latitudes,” in *2016 IEEE/OES Autonomous Underwater Vehicles (AUV)*, Nov 2016, pp. 115–123.
- [74] Sam and N. Reid, “The self-driven particle model - an interactive tutorial,” <http://tutorials.siam.org/dsweb/particles/>, 2005.
- [75] J. Sijs, M. Lazar, and P. v.d.Bosch, “State fusion with unknown correlation: Ellipsoidal intersection,” in *American Control Conference (ACC), 2010*, June 2010, pp. 3992–3997.

- [76] T. L. Song, “Observability of target tracking with range-only measurements,” *IEEE Journal of Oceanic Engineering*, vol. 24, no. 3, pp. 383–387, July 1999.
- [77] S. Spires and S. Goldsmith, “Exhaustive geographic search with mobile robots along space-filling curves,” in *Collective Robotics*, ser. Lecture Notes in Computer Science, A. Drogoul, M. Tambe, and T. Fukuda, Eds. Springer Berlin Heidelberg, 1998, vol. 1456, pp. 1–12. [Online]. Available: <http://dx.doi.org/10.1007/BFb0033369>
- [78] M. Stipanov and S. Fioravanti, “Camelot - localization beacon system,” in *2018 IEEE/OES Autonomous Underwater Vehicles (AUV 2018)*, (Porto, Portugal), Nov 2018.
- [79] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, 1st ed. Cambridge, MA, USA: MIT Press, 1998.
- [80] Y. T. Tan, M. Chitre, and F. S. Hover, “Cooperative bathymetry-based localization using low-cost autonomous underwater vehicles,” *Autonomous Robots*, 2015.
- [81] Y. T. Tan, R. Gao, and M. Chitre, “Cooperative path planning for range-only localization using a single moving beacon,” *IEEE Journal of Oceanic Engineering*, vol. 39, no. 2, pp. 371–385, April 2014.
- [82] T. Tao, Y. Huang, J. Yuan, F. Sun, and X. Wu, “Cooperative simultaneous localization and mapping for multi-robot: Approach & experimental validation,” in *2010 8th World Congress on Intelligent Control and Automation*, July 2010, pp. 2888–2893.
- [83] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.
- [84] J. K. Uhlmann, “Covariance consistency methods for fault-tolerant distributed data fusion,” *Information Fusion*, vol. 4, no. 3, pp. 201–215, September 2003.

- [85] S. E. Webster, R. M. Eustice, H. Singh, and L. L. Whitcomb, “Advances in single-beacon one-way-travel-time acoustic navigation for underwater vehicles,” *I. J. Robotics Res.*, vol. 31, no. 8, pp. 935–950, 2012.
- [86] S. Webster, J. Walls, L. Whitcomb, and R. Eustice, “Decentralized extended information filter for single-beacon cooperative acoustic navigation: Theory and experiments,” *Robotics, IEEE Transactions on*, vol. 29, no. 4, pp. 957–974, 2013.
- [87] Y. Weiss and W. T. Freeman, “Correctness of belief propagation in gaussian graphical models of arbitrary topology,” *Neural Computation*, vol. 13, no. 10, pp. 2173–2200, Oct 2001.
- [88] G. Welch and G. Bishop, “An introduction to the kalman filter,” *University of North Carolina at Chapel Hill, Chapel Hill, NC*, 1995.
- [89] J. F. Wellmann, “Information theory for correlation analysis and estimation of uncertainty reduction in maps and models,” *Entropy*, vol. 15, no. 4, pp. 1464–1485, 2013. [Online]. Available: <http://www.mdpi.com/1099-4300/15/4/1464>
- [90] E. Wolbrecht, B. Gill, R. Borth, J. Canning, M. Anderson, and D. Edwards, “Hybrid baseline localization for autonomous underwater vehicles,” *Journal of Intelligent & Robotic Systems*, vol. 78, pp. 593–611, 2015.
- [91] X. S. Zhou and S. I. Roumeliotis, “Determining the robot-to-robot relative pose using range-only measurements,” in *Proceedings of IEEE International Conference on Robotics and Automation, Rome, Italy*, April 2007, pp. 4025–4031.
- [92] Y. Zhou and J. Li, “Data fusion of unknown correlations using internal ellipsoidal approximation,” in *Proceedings of the 17th World Congress, The International Federation of Automatic Control Seoul, Korea*, July 2008.