

 M 2015



# REAL-TIME UNSUPERVISED MOTION LEARNING FOR AUTONOMOUS UNDERWATER VEHICLES USING GAUSSIAN MIXTURE MODELS

**ABDOLRAHMAN KHOSHROU**

DISSERTAÇÃO DE MESTRADO APRESENTADA  
À FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO EM  
ÁREA CIENTÍFICA

**FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO**



**FEUP** FACULDADE DE ENGENHARIA  
UNIVERSIDADE DO PORTO

# **Real-Time Unsupervised Motion Learning for Autonomous Underwater Vehicles using Gaussian Mixture Models**

**Abdolrahman Khoshrou**

WORKING VERSION

Master in Information Engineering

Supervisor: Professor António Pedro Aguiar

July 27, 2015



# Abstract

Over the last decade, there has been a flurry of activity in the development of autonomous marine robotic vehicles to improve the means available for ocean exploration and exploitation. A particular scenario where Autonomous Underwater Vehicles (AUVs) can play an important role is in the automatic acquisition of marine environmental data. In this case, one or more AUVs acting in cooperation are programmed to survey a given region. To this end, an important problem that has to be addressed is the sampling motion control strategy, that is, the high-level software algorithm that is implemented on a computer system at the AUV, that decides based on the on-board sensors where (and in some cases when) to acquire environmental data.

Motivated by the above, this thesis proposes a solution to solve the problem of real-time adaptive sampling using a coordinated fleet of AUVs. In the first part of the thesis, we address the on-line unsupervised learning problem of Gaussian mixture models (GMMs) in the presence of uncertain dynamic environments. In particular, we assume that the number of Gaussian components (clusters) is unknown and can change over time. We propose a multi-hypothesis adaptive algorithm that continuously updates the number of components and estimates the model parameters as the measurements (sample data) are being acquired. This is done by incrementally maximizing the likelihood probability associated to the estimated parameters and keeping/creating/removing in parallel a number of hypothesis models that are ranked according to the minimum description length (MDL), a well-known concept in information theory. The proposed algorithm has the additional feature that it relaxes “the sufficiently large data set” restriction by not requiring an initial batch of data. Simulation results illustrate the performance of the proposed algorithm.

In the second part of the thesis, we use the proposed unsupervised algorithm to develop an adaptive sampling strategy to obtain relevant conductivity, temperature and depth (CTD) information of a given area using a coordinated fleet of AUVs. In the proposed setup, a leader AUV is tasked to acquire CTD data by running a set of user-defined mission instructions like for example following a desired path profile. The rest of the fleet (the followers AUVs) will follow the leader closely with a desired formation that will adaptively change according to the CTD data that they are acquiring. More precisely, each AUV is in charge of running in real-time the proposed unsupervised learning algorithm for GMMs that is fed by the CTD data. At each time that the vehicles resurface (and this is done in a coordinated fashion), the leader AUV broadcast its currently estimated parameters of the GMM, and the followers based on this and their estimated GMM compute the variational distance error between these GMMs. This error, which provides a notion of how different is, from the leader, the CTD measurements that each follower is acquiring, is then used to update the next formation configuration, which typically scales the distance between the AUVs in the formation (making a zoom-in and zoom-out), in order to improve the efficiency of data acquisition in a given region. The simulation results show the feasibility and accuracy of the motion learning strategies in many uniform and complex environments.



# Acknowledgments

My deepest gratitude goes to my supervisor Professor António Pedro Aguiar. I am extremely thankful and indebted to him for sharing his expertise, and sincere and valuable guidance and encouragement extended to me. He has set an example of excellence as a researcher, mentor, instructor and role model.

Next, I need to thank all the people who created such a good atmosphere in the lab; Silvia, André, Luís, Hugo and Behzad. I would also like to thank the secretarial and academic staff for putting up with me and answering all my questions. A special thanks goes to Mónica Oliveira, Catarina Morais, Paulo Lopes and José António Nogueira who all deserve early retirement after I leave.

I am also very thankful to all friends I met in Portugal. I had a great time with you. A lot of thanks goes to Michael Silva, Luís Mendes, Carlos Almeida and Mario Alejandro Salgado. I hope to see you again as soon as possible.

And last but not least, my heartfelt thanks goes to my family. My mother Mina, my sisters Hannaneh and Samaneh and my brother Hamid. You are a constant source of understanding and support.



*“He who says he can and he who says he can’t are both usually right”*

Confucius



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Technologies for ocean sampling . . . . .	2
1.1.1	Research Vessels . . . . .	3
1.1.2	Moored Instrumentation and Ocean Observatories . . . . .	3
1.1.3	Towed Systems . . . . .	5
1.1.4	Remotely Operated Vehicles . . . . .	5
1.1.5	Autonomous Underwater Gliders . . . . .	6
1.1.6	Unmanned Surface Vessels: Wave Gliders . . . . .	7
1.1.7	Autonomous Underwater Vehicles . . . . .	7
1.1.8	The AOSN Concept . . . . .	8
1.2	Objectives . . . . .	9
1.3	Contributions and Scope . . . . .	10
1.4	Dissertation Outline . . . . .	11
<b>2</b>	<b>Unsupervised Learning of Gaussian Mixture Models</b>	<b>13</b>
2.1	Introduction . . . . .	13
2.2	Previous Works/Literature Review . . . . .	14
2.3	Problem Statement . . . . .	16
2.4	Preliminaries and basic results . . . . .	17
2.4.1	The Basic Expectation-Maximization (Off-line) Algorithm . . . . .	17
2.4.2	Titterington's On-line Algorithm for a Multivariate Normal Mixture . . . . .	23
2.4.3	The Minimum Description Length (MDL) Principle . . . . .	25
2.4.4	Gaussian Mixture Reduction . . . . .	28
2.5	The Proposed On-line Unsupervised Learning Algorithm . . . . .	33
2.6	Simulation Results . . . . .	36
2.6.1	A 2-d Gaussian Mixture . . . . .	36
2.6.2	The Iris Data Set . . . . .	36
2.6.3	A 1-d Gaussian Mixture . . . . .	37
2.7	Conclusion . . . . .	37
<b>3</b>	<b>Real-Time Unsupervised Motion Learning for Autonomous Underwater Vehicles</b>	<b>41</b>
3.1	Introduction . . . . .	41
3.2	Literature Review on Adaptive Sampling using AUVs . . . . .	42
3.3	Problem Formulation . . . . .	45
3.4	Preliminaries and Background . . . . .	45
3.4.1	Coordinate Frames . . . . .	46
3.4.2	Simplified Kinematic Equations . . . . .	47
3.4.3	CTD Sensors . . . . .	48

3.4.4	Navigation . . . . .	50
3.5	CTD adaptive sampling strategy . . . . .	55
3.5.1	Path Following . . . . .	56
3.5.2	Coordinated Formation . . . . .	57
3.5.3	Variational distance between GMMs . . . . .	57
3.6	Simulation Results . . . . .	58
3.6.1	The Leader AUV . . . . .	58
3.6.2	The follower AUV . . . . .	59
3.6.3	Uniform Environment Simulation . . . . .	61
3.6.4	Complex Environment Simulation . . . . .	63
3.7	Conclusion . . . . .	63
<b>4</b>	<b>Conclusions and FutureWork</b>	<b>65</b>
4.1	Conclusions . . . . .	65
4.2	Future Work . . . . .	66
<b>References</b>		<b>67</b>

# List of Figures

1.1 Underwater exploration (image sources: [3]). . . . .	1
1.2 A Lesson In Complexity (image source: [4]) . . . . .	2
1.3 Ocean expedition (image sources: Left [4], Right [2]). . . . .	3
1.4 Endurance array and slope base mooring (image source: [4]) . . . . .	4
1.5 Tethered Underwater Vehicles (image sources: Left [2], Right [4]). . . . .	5
1.6 Free-Swimming Autonomous Sampling Vehicles (image sources: Left [2], Right [12]). . . . .	6
1.7 Light Autonomous Underwater Vehicles (image source: [17]) . . . . .	8
1.8 The Autonomous Ocean Sampling Network (image source: [1]) . . . . .	9
2.1 An example of good initialization . . . . .	20
2.2 EM drawbacks . . . . .	21
2.3 An example of the execution behaviour of the proposed algorithm. . . . .	35
2.4 Iris Data Set Results . . . . .	37
2.5 Linearly separable mixture . . . . .	38
2.6 Fairly overlapped mixture . . . . .	38
2.7 Highly overlapped mixture . . . . .	38
3.1 The earth-fixed inertial frame $\{U\}$ and the body-fixed frame $\{B\}$ with position, orientation and linear and angular velocities . . . . .	46
3.2 Simplified kinematic model of an underwater vehicle that maintains in a horizontal plane. . . . .	47
3.3 A close up of the CTD sensor of the AUV (image source: [1]) . . . . .	48
3.4 Temperature Profiles (images source: [17]). . . . .	49
3.5 A two-dimensional kinematic model of an AUV. The angle $\theta$ of the resulting velocity $V$ does not necessarily correspond with the heading angle $\psi$ . . . . .	51
3.6 A typical DVL device (image source: [1]) . . . . .	52
3.7 Two categories of INSs (image sources: Left [4], Right [7]). . . . .	53
3.8 Overall visual servoing control scheme.(image source: [117]) . . . . .	54
3.9 Schematic representation of the setup . . . . .	56
3.10 Schematic representation of the leader . . . . .	58
3.11 Schematic representation of the follower . . . . .	59
3.12 Real CTD data . . . . .	60
3.13 Uniform environment. . . . .	61
3.14 Uneven environment. . . . .	62



# List of Tables

3.1 Simulation parameters . . . . .	57
-------------------------------------	----



# Abbreviations and Symbols

GMM	Gaussian Mixture Model
EM	Expectation Maximization
ML	Maximum Likelihood
MAP	Maximum A posteriori Probability
CTD	Conductivity, Temperature, Depth
AIC	Akaike's Information Criterion
BIC	Bayesian Inference Criterion
FIM	Fisher Information Matrix
MHT	Multi-Hypothesis Tracker
JPDAF	Joint Probabilistic Data Association Filter
KL	Kullback-Leibler
MDL	Minimum Description Length
MML	Minimum Message Length
SC	Stochastic Complexity
RCA	Robust Competitive clustering Algorithm
ART	Adaptive Resonance Theory
AUV	Autonomous Underwater Vehicle
ACM	Acoustic Current Meter
DSL	Digital Subscriber Line
USV	Unmanned Surface Vessel
ASC	Autonomous Surface Craft
AOSN	Autonomous Ocean Sampling Network
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
GPS	Global Positioning System
ITRF2000	International Terrestrial Reference Frame 2000
ROV	Remotely Operated Vehicle
DVL	Doppler Velocity Log
RPM	Round Per Minute
UVV	Unmanned Underwater Vehicle
ROMS	Regional Ocean Model System
COLREGs	International Regulations for Prevention of Collision
SONAR	Sound Navigation And Ranging
LBL	Long-Base-Line
USBL	Ultra-Short-Base-Line



# Chapter 1

## Introduction

The ocean is a crucial feature of Earth, but due to extreme conditions in deep beneath the surface, it still remains a nearly unexplored territory. Ocean expedition, methodical observations and documentation of different outlooks of the ocean, e.g., in biology, geochemistry, physics and archaeology can help us to reveal the mysteries of the deep ocean ecosystems. Recent discoveries reveal that largest mass of living things on earth are living near the ocean floor.

Figure 1.1(a) shows expanding of molten rock in the form of bubbles, under the pressure of magmatic gas. This magmatic gas is presumed to be mostly water because when the water is under the magmas and the magmatic temperature is a thousand degrees and when it is suddenly cooled by coming contact with the sea water, the bubble bursts. The ocean floors are formed exclusively by these volcanic activities.

Hydrothermal vents, which are chimney like structures with several stories high, spewing hot water geysers black with minerals and nutrients. The temperature around these deep sea events can scorch up to  $760^{\circ}F$ . Figure 1.1(b) shows an astonishing sight of an exotic garden of giant tube worms, that thriving without sunlight and feed on sulfur compounds erupting from the vents in this toxic water [1, 2]. Deep understanding of the ocean, demands novel research approaches to



(a) Hydrothermal vents



(b) An exotic marine garden

Figure 1.1: Underwater exploration (image sources: [3]).

study interactive ocean processes thoroughly and simultaneously to determine the linkage among



Figure 1.2: A Lesson In Complexity (image source: [4])

and between chemical, physical, biological and geological factors in a coherent temporal and spatial framework, see Figure 1.2. For example, the chemistry of water in different geographic areas, affect the organisms that live there and those in turn influence the geology of the ocean floor. With the recent advances in technology, ocean observatories are gaining the capacity to fulfil the expectations in expedition of vast, unstructured and dynamic environments of the ocean by analysing the dependency and correlation between ocean dynamics, climate and ecosystem responses at the local to basin scales in real-time. These powerful new approaches are named as the grand challenges in environmental and ocean sciences [4].

## 1.1 Technologies for ocean sampling

Our climate and weather, the water we drink and the food we eat, even the air we breathe, all of these are closely connected to the ocean that covers almost three quarters of our planet. Today, much of the ocean and how it works still remains a mystery, but in the way that the doctors constantly monitor the health of critical patients, scientists from around the world are now using the state-of-the-art technology to constantly take the pulse of our dynamic marine and coastal environments.

In the early days, exploring the ocean started by sending divers down into the water. Divers had to deal with a number of hazards such as decompression, dragging to the ship holes or pounding into the infrastructures, due to the currents. During the years, researchers use observatories wherever they can so they don't have to put themselves in those kind of situations.

Ocean observatories are collections of high-tech instruments above and below the waves that provide around-the-clock information about what is happening in the ocean in an systematic non-invasive way.

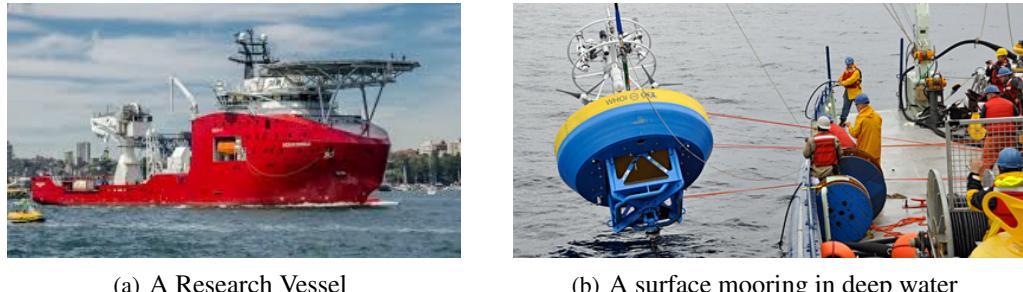


Figure 1.3: Ocean expedition (image sources: Left [4], Right [2]).

Sensors on satellites miles above the earth, look at both large and small areas of the ocean surface providing key data about the temperature, color and tide of the water. Radar towers on land, collect information about the movement of the water at the surface of the motion, including the speed and the direction of the surface currents. Sensors and instruments attached to the stationary buoys collect the information at the same location over long periods of time.

Autonomous underwater vehicles travel independently below the ocean surface collecting information about the water conditions. This data then sent back to the scientists on land or board ships. Instruments connected to the network of underwater hops called nodes continuously collect data and send it back to the land through cables. The same cables also provide electrical power to the nodes and other equipments. Using data collected through ocean observatories, scientists are now beginning to forecast ocean conditions, much like meteorologists do for the weather.

### 1.1.1 Research Vessels

A typical way to get ocean measurements is called shipboard hydrographic programme, which is the employment of in-situ sensors on cargo ships or ferry-boats. The ships are equipped with special tools and technology that allow scientists to collect samples and taking measurements in a different levels of the ocean. Annual cruises are launched to measure the water properties by lowering down the instruments all the way to the bottom of the ocean, step by step in different levels, to measure the water properties and the flow. This enables the scientists to get a snapshot of the ocean circulation yearly. In early design vessels, typical instruments such as temperature and conductivity recorders, current profilers and so on, capture the data and store them internally. It is only retrieved when the ship reaches the final destination [5]. Late research vessels, equipped with state-of-the-art technological gadgetry, provide stable platforms for the scientists to deploy divers and submersibles, Figure 1.3(a).

### 1.1.2 Moored Instrumentation and Ocean Observatories

Ocean processes are around-the-clock, so, the use of annual research vessel cruises is not practical to observe the ocean throughout the year. Scientists and engineers have come up with ways to

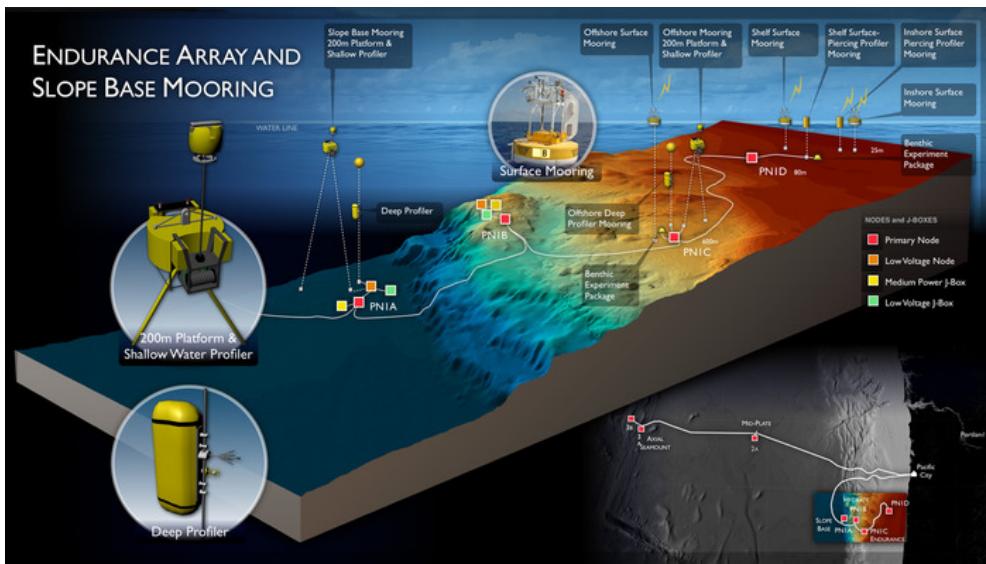


Figure 1.4: Endurance array and slope base mooring (image source: [4])

leave instruments out in the ocean for long-term monitoring.

A cost-effective solution could be implementing an array of moored profilers which are anchored to the bottom of the ocean. Moored observatories are platforms which can occupy a spot in the ocean and allow the researchers to observe a column of water years. They are important tools for monitoring the oceanic processes in large temporal scale and, due to advances in sensor technologies, the of them increasing in the last decade [6]. This technology can help the researchers to develop ocean-atmosphere models, and learn about air-sea interactions in seasonal and yearly scales.

The basic suite of instruments of a typical buoy is depicted in Figure 1.3(b). Above the water, moored buoys equipped with meteorological sensors and solar panes are able to measure the humidity and temperature of the air and solar and/or infra red radiation. Beneath the surface, buoys hold various instruments, such as: current meters, temperature and pressure sensors, sediment traps, chemical sensors, power supplies, data recorders, and an ACM (acoustic current meter). The captured data can be sent to a central repository in almost real-time through satellite or radio communication systems. Researchers are looking ahead to the prospect of sending data home in real-time via satellite [7]. It is quite challenging to get these instruments to operate for years in sea in very harsh environments, and only a very small portion of the ocean can be covered in this approach. As a result, the spatial resolution is typically poor due to the high cost. Furthermore, sea water generally does not get along well with electronic devices and, on the other hand, the surface waves can physically damage the instruments and the moorings.

With recent developments, the instrumentation and performance of the moored arrays can be improved by combining a profiling mechanism to a moored based system, see [8]. Nowadays, fibre cables are capable of supplying electric power for the mooring profilers and platforms and transferring the data from deep in the ocean to the land, see Figure 1.4. Also, battery-powered

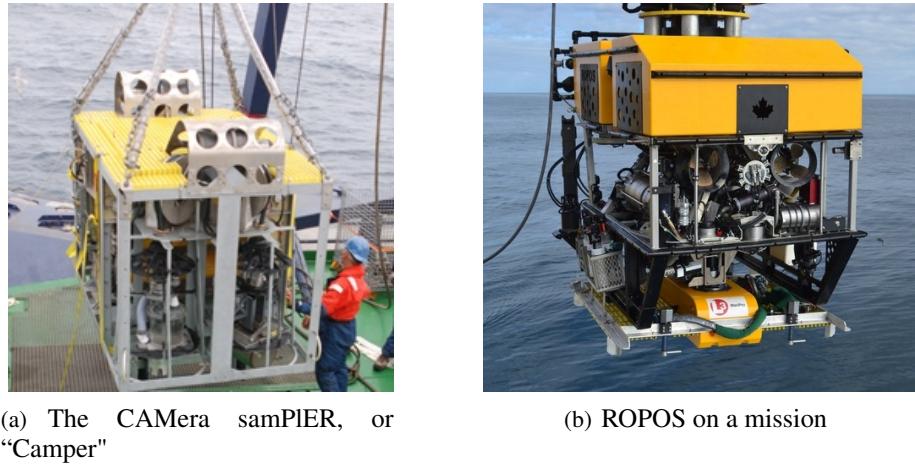


Figure 1.5: Tethered Underwater Vehicles (image sources: Left [2], Right [4]).

traction motor can help the profilers to climb up and down the mooring and provide unparalleled spatial-temporal resolution on biological and geochemical features, ocean acidification, and other oceanographic processes.

### 1.1.3 Towed Systems

To expand the spatial scale of the area under study, scientists and engineers have devised large sampling sledges that are tethered to research vessels. Towed systems lowered down to the sea floor to collect samples from the ocean bottom. With respect to the objective of the mission, towed sled can be equipped with different instruments to collect samples or take measurements. Chemical sensors, CTD sensor devices, lights and optical instrumentation, a navigation system, a video camera, and claws to quickly grab samples, can be mounted on these sledges, Figure 1.5(a). Recent towed systems, by using impeller-forced wings, are able to rotate and undulate to the upper level of water. A multi-conductor tow cable, or in sonar type a long fibre, is used to send the control signals from on-board ship control unit to a hydraulic unit inside the vehicle to modify the wings. Towed systems have limited maneuverability, are slow, costly and controlled from a ship via a cable or fibre, that provides electrical and data transfer possibility. Some undulating versions allow for complex vertical yo-yo paths, even so, during the last decade towed systems are being widely replaced by untethered submersibles. See [2, 9] for more details.

### 1.1.4 Remotely Operated Vehicles

Remotely Operated Vehicles (ROVs), see Figure 1.5(b), are tremendous powerful platforms tethered by the ships and typically known as underwater helicopters, because they can hover over a sea floor target and surf a vast area to explore unblemished ocean parts.

ROVs are tethered to the research vessel by a fiber-optic cable to be able to send high band-width data to a control console and receive electrical power and mission plan.

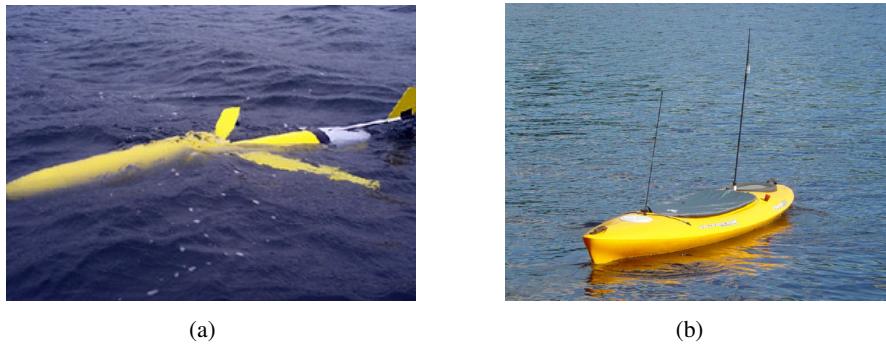


Figure 1.6: Free-Swimming Autonomous Sampling Vehicles (image sources: Left [2], Right [12]).

Unlike the towed vehicles in Section 1.1.3, here, a pilot can have a direct control over the vehicle's mission using a hand-controller to send the control commands and adjust the sensors. This, allows the researchers to carry out detailed experiments, take measurements, collect samples and observe the deep sea floor while controlling the vehicle from the surface. Another advantage is that, ROVs are less affected by the position and relative motion of the vessel and the current, so they get perfect sensor data, perfect video. ROVs, typically, are equipped with video cameras, lights, acoustic imaging sonars, which are used to "see" in zero visibility, velocity meters and CTD sensor devices. Using tracking and navigation systems, the exact location f them are known in real-time which increases the maneuverability of these vehicles. Recent ROVs are called smart flights which are semi-autonomy package that allow return to the target by pushing a button doing way-points navigation.

High cost, constrained mobility by the umbilical, relatively low speed the drag associated to the frame and the tether are the major challenges in using these vehicles [9].

### 1.1.5 Autonomous Underwater Gliders

An Autonomous Underwater Glider (AUG) is a buoyancy-propelled, fixed-wing vehicle that enables robotic exploration in large bodies of water, Figure 1.6(a). with the help of its wings and by means of internal mass redistribution, the vehicle goes up and down through the ocean in a pattern called a saw-tooth [10]. The wings actually take the vertical motion of the vehicle and translate that into forward motion. In oceanographic studies, these robots are being used to take scientific measurements of salinity, temperature, volcanic matters, oxygen contents in the water and so on. They sink and rise, on command, to send the high-resolution data back to the researchers. AUGs, while on the surface, can communicate with the operators via RF modems or Iridium satellite phones. As mentioned earlier, gliders sink and rise on command, i.e., during the mission the operators are able to program and change the current task or waypoints of the vehicle and send the commands via Iridium satellite links. Portability and scalability in infrastructure of the fleet are the main advantages of these small, inexpensive platforms to cover a vast area. On the other hand, the waves can cause a lot of damage to the robots or they may get hit by other vehicles during

resurfacing. The surface currents can become a serious issue, because it is not necessarily clear what the trajectories of the glider are. Therefore, the researchers may have to trace the AUG down using its last known location as well as the prediction of water currents to localize and recover the vehicle. Nevertheless, fleet of AUGs are one of the best at hand approaches to obtain subsurface spatial resolution necessary in oceanography [11].

### 1.1.6 Unmanned Surface Vessels: Wave Gliders

Unmanned surface vessels (USVs) are robotic float boats that typically use solar panels to power their electric systems for propulsion and operate without any tether link with an operator.

USVs have a variety of applications in oceanography science, ocean acidification, national security, phytoplankton studies and general robotics research, e.g. [2]. Figure 1.6(b) shows an Unmanned Wave Glider (UWG), which is a simple, cost effective platform for collecting ocean data that does not rely on expensive ships or buoys. These wave gliders are propelled through the water by underwater fins or wings that convert the wave energy into forward thrust. UWGs have revolutionized the cost of data gathering in oceanographic studies. They are the generation of marine vehicles that are persistent and wave-propelled systems, without any need to fuel or crew.

A wave glider is a two-part system, the float which stays on the surface the entire time and a set of wings or fins that are connected to the float by a cable. When the vehicle is deployed the set of wings are a few meter below the float. Whatever direction the float is get to move by the ocean waves, the glider wings goes up and down with the motion of the ocean surface and consequently the vehicle propelled itself. Affordable, long range and higher bandwidth satellite system, powered by solar panels, is used for communication and navigation in real-time. These systems, along with accurate and compact global positioning systems, have also increased the reliability and applicability of the UWGs.

While the technology potential of USVs is bright, looming policies in shipping and surface traffic rules restrict their applicability [13].

### 1.1.7 Autonomous Underwater Vehicles

Some of the most recent technological advances, particularly in the last decade, have fostered the development of Autonomous Underwater Vehicles [14]. An Autonomous Underwater Vehicle (AUV) is a robot designed to operate underwater, see Figure 1.7. It is typically a free body swimming and is not attached to the support vessel to which is launched. Equipped with a set of relevant sophisticated in-situ sensors, AUVs characterize the underwater environment without real-time control by human operators. This characteristic increases the maneuverability of AUVs in temporal and spatial scales in oceanography applications. AUVs have brought together specific complementary knowledge in computer science, electrical and mechanical engineering.

These vehicles, are a vital tool in gathering detailed ocean data at reasonable cost in order to targeting specific set of oceanography questions. The vehicle is programmed to complete a particular mission and uses its on-board sensors to estimate its state in order to complete those mission



Figure 1.7: Light Autonomous Underwater Vehicles (image source: [17])

objectives. AUVs are generally used in scientific expeditions to map new environments using a variety of sensors including sonar and/or vision. Scientific applications include geoscience to map particular features such as hydrothermal events or submarine volcanoes [4]. In oceanography for mapping the physical structure of the ocean [2]. In archeology for documenting shipwrecks and submerged cities [7]. In ecological applications for surveying the marine habitants and document their states to understand the changes through time [4]. In industry they are extensively used for conducting surveys for minerals and in oil and gas exploration. AUVs are also used in defence applications to fulfil dangerous roles such as mine counter measures and rapid environmental assessments. Conventional deployment of the AUVs generally involves following a predefined mission path in terms of a series of way points, [15]. Because of unstructured and harsh environments underwater, navigation and localization of the vehicle is an uneasy task. Thus, various control techniques needed to intelligently correct the position errors and also update the mission plan e.g. [16]. In Section 3.4, AUVs are presented in more detail.

### 1.1.8 The AOSN Concept

In the last couple of decades, with the advances in technology, the introduction of robotic systems are starting to play an important role. Robotic systems show up in several different ways. Numerous remotely operated vehicles as tele-operated systems, tethered by the ships and operated by the human operators are being used in oceanographic studies. But, another class of platforms are AUVs which usually are smaller than tethered platforms but can carry quite sophisticated payloads. Equipped with sonar scan and sub-bottom profilers, AUVs can look for ship wrecks or make map of the bottom of the ocean, probably with highest resolution maps of any other vehicles. AUV is a preferred platform, for doing certain classes of ocean observing but its endurance and battery power is tight comparing to the ship. Energy is a fundamental constraint in designing underwater systems. We can divide the energy used in an AUV in two general categories. One that depends on the velocity of the vehicle raised to the power of three, the other called *a hotel load* which in fact

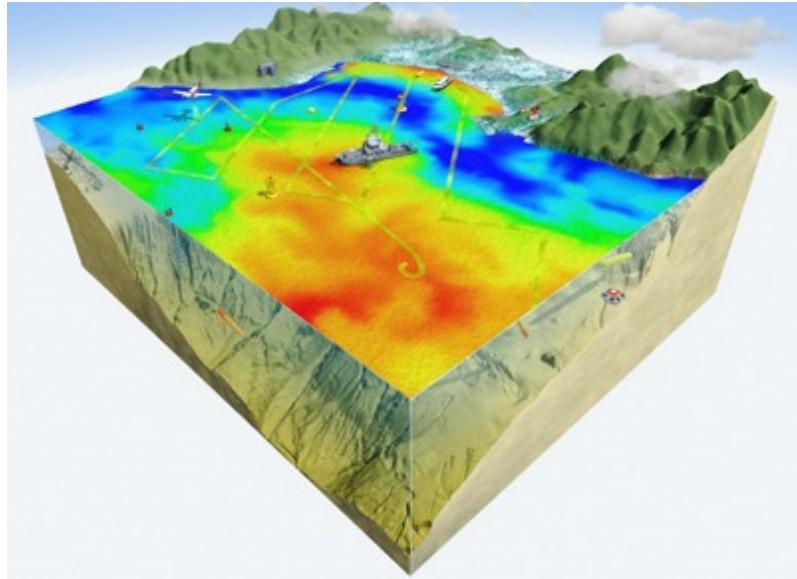


Figure 1.8: The Autonomous Ocean Sampling Network (image source: [1])

depends on everything else in the vehicle such as control, navigation systems and sensor devices. So, the trade off in designing AUVs is between endurance and the energy consumption. There are vehicles designed with low power sensors, very long endurance and comparatively low speeds. The other types have much higher power sensor systems, fairly short endurance and significantly higher speeds.

It can be mathematically proved that, using a couple of vehicles instead of one to do the same survey can increase the endurance of the vehicles and decrease the power consumption dramatically. For example, to do a survey in a given path in a certain area, consider the fact of using two vehicles instead of one. What happens then is that the path way that each vehicle goes drops by a factor of two, but if the time of experiment stays the same, then the speed drops by a factor of two and the power consumption of each vehicle drops by eight. As a result, the total energy consumed by the observation system is a factor of four lower. This realization can make the sensor systems very efficient.

The concept of Autonomous Ocean Sampling Networks (AOSN) is introduced in [18], which is a multi-platform approach to provide a framework to encompass a set of cooperative efforts to observe an area of the ocean. Figure 1.8 shows a field experiment involved numerous ships and dozens of floating, profilers, and autonomous oceanographic instruments operating simultaneously. AOSNs have enabled the observatories to cover larger areas and be able to persist for large amounts of time.

## 1.2 Objectives

The main motivation behind this thesis is to develop an adaptive sampling strategy to obtain relevant conductivity, temperature and depth (CTD) information of a given area using a coordinated

fleet of Autonomous Underwater Vehicles (AUVs), where one AUV plays the role of leader and the other(s) as follower(s). To this end, a real-time motion learning algorithm will be developed that allows to survey the ocean in close proximity, and provide suitable viewpoints for different applications. We assume that the mission plan of the leader is predefined, while as for the follower(s), they will be defined in real-time according to the motion algorithm that will be developed. In this work, we take into account the communication constraints and in particular we consider that the vehicles can not exchange data underwater at all. The follower(s), only on the surface can talk to the leader. The main objectives of the thesis are:

- Propose a new algorithm to solve the problem of real-time unsupervised learning of Gaussian Mixture Models (GMMs) in the presence of uncertain dynamic environments, i.e., we assume that the number of Gaussian components (clusters) is not only unknown but it can also change over time.
- Carry out a thorough review of prior work in adaptive sampling and real-time motion learning literature and identify requirements in this domain that have not been addressed in the literature.
- Use the algorithm developed in 1. and propose a novel real-time adaptive sampling motion learning for Autonomous Underwater Vehicles (AUVs) using Gaussian mixture models. The objective is to find strategies that are suitable for noisy environment and extremely restricted communication. We assumed that the communication constraints underwater are extreme, i.e., the vehicles can only communicate on the surface.
- Implement and evaluate the performance of the algorithm proposed through extensive computer simulations.

### 1.3 Contributions and Scope

A novel on-line unsupervised learning of GMMs in the presence of uncertain dynamic environments is proposed to be used in motion learning for AUVs. We assume that the complexity of the model not only is unknown, but it can also change over time. In other words, the number of Gaussian components is unknown and not fixed. Inspired by the work in [19], namely the use of the *minimum description length* (MDL) principle, we propose a multi-hypothesis adaptive algorithm that continuously updates the number of components and estimates the model parameters as the measurements (sample data) are being acquired. The proposed algorithm has the additional feature that it relaxes “the sufficiently large data set” restriction by not requiring in fact any initial batch of data. Simulation results illustrate the performance of the proposed algorithm where it shows that indeed it is able to continuously adapt to the dynamic changes of the number of clusters and estimate the parameters of the mixture model.

A second key contribution is the following: we address the problem of adaptive sampling using a coordinated fleet of AUVs. The system setup consist of one leader AUV and two or more follower

AUVs, where all the vehicles are equipped with a conductivity, temperature and depth (CTD) sensor devices. The CTD data that is being acquired by each AUV is modelled in real-time as a GMM by running locally the proposed unsupervised learning algorithm developed in the first part of the thesis. The task of the leader vehicle is to follow some desired, predefined path profile while acquiring CTD data. The aim of the follower AUVs is to follow the leader motion by keeping a desired formation that is a function of a particular error. This error is the *variational distance* between two GMMs, [20], that gives a notion of how different is the environment that each follower is travelling within from the leader. In other words, every follower has to explore the environments where in terms of CTD data are desirably, given by a *distance* value between two GMMs, different from the leader. The experimental results show the feasibility and accuracy of the motion learning strategies in many uniform and complex environments.

## 1.4 Dissertation Outline

The remainder of this dissertation is organized as follow. Chapter 2 is dedicated to the problem of Unsupervised Learning of Gaussian Mixture Models. First, we present an overview of the state of the art of the algorithms and methods for unsupervised learning of Gaussian mixture models. Some background knowledge of unsupervised learning of GMMs is also described. A particular point is the Titterington's on-line algorithm to update the GMM in time, which is highlighted in Section 2.4.2. Section 2.4.3 explains the minimum description length principle, which is used to determine the complexity of the GMM in our proposed algorithm. Gaussian mixture reduction methods, as another important feature of our algorithm, is presented in Section 2.4.4. Our proposed algorithm is fully described in detail in Section 2.5. The applicability of the algorithm over synthetic and real datasets are examined in Section 2.6.

Chapter 3 is devoted to the second contribution of the thesis, which is the development of an adaptive ocean sampling scheme for a formation of AUVs. This chapter starts with the motivation and an overview of strategies for ocean sampling. A brief introduction to coordination frames of AUVs is then brought in Section 3.4.1. Section 3.4.2 presents a simplified kinematic dynamic motion equation for an AUV. The CTD sensor devices are explained in Section 3.4.3. Section 3.6 contains our contribution for real-time motion learning for a fleet of autonomous underwater vehicles. In the proposed setup, the AUVs are required to keep a defined formation pattern while individually solving path following problems. As mentioned before, the objective is to find strategies that are suitable for noisy environment and restricted communication. Therefore data exchange within the vehicles are only carried out on the surface. Each AUV system contains a path generator, inter vehicle coordination, communication, path following control, inner low-level dynamic control (inner loop) and a position estimator. Computer simulations of complete closed loop motion control system for each AUV as well the overall proposed adaptive ocean sampling scheme were implemented in Matlab/Simulink ®to evaluate the performance in uniform and more complex environments. The simulations presented show that the results are sufficiently promising to be tested in real scenarios. The conclusion and future works are presented in Chapter 4.



# Chapter 2

## Unsupervised Learning of Gaussian Mixture Models

This chapter addresses the problem of unsupervised learning of Gaussian mixture models. We start with a brief introduction to the problem and present an overview of the state of the art of Gaussian mixture models. Section 2.3 contains the problem statement that we propose to solve. Some basic and background knowledge is then provided in Section 2.4. Our proposed algorithm is fully described in detail in Section 2.5. In Section 2.6, the viability of the proposed unsupervised learning of the GMMs algorithm is investigated over synthetic and real data. Section 2.7 wraps up this chapter.

### 2.1 Introduction

Data analysis plays an indispensable role for understanding various phenomena. Cluster analysis, primitive exploration with little or no prior knowledge, consists of research developed across a wide variety of communities. In order to learn a new object or understand a new phenomenon, people always try to seek the feature that can describe it, and further compare it with other known objects or phenomena. The comparison is based on some similarity or dissimilarity, generalized as proximity, according to some certain standards or rules. The goal of clustering is to distinguish and characterize a finite and discrete set of intrinsically similar or natural in a finite unlabelled dataset. Generally speaking, in clustering which also called exploratory data analysis, we are more interested in finding the hidden data structures, rather than provide an accurate characterization of unobserved samples generated from the same probability distribution [21, 22]. This can make the task of clustering fall outside of the framework of unsupervised predictive learning problems, such as probability density function estimation [23] and entropy maximization [24].

Over the years, research on finding the best complex model, identifying and classifying unknown number of components, to describe a random process has been an important topic in computer vision and pattern recognition communities. In particular, for data clustering, mixture models, where each component density of the mixture represents a given set of individuals in the total

community, has been applied in a widespread of applications. For example, the deployment of generic recognition or tracking systems with minimal prior knowledge to set-up the initialization and the ability to train the model over time. Generally speaking we can name some desired characteristics of clustering methods as below

- being flexible and capable of generating arbitrary shapes of clusters rather than be limited to some particular shape;
- handle big data as well as high-dimensional features with acceptable model complexity and in reasonable time;
- be able to find an automated solution with minimum reliance on the user-defined parameters;
- capable of detecting and excluding the outliers and noises in the data;
- have the capability in real-time applications without relearning from the scratch;
- be immune to the effects of order of incoming points;
- provide some insight to estimate the complexity of the model without prior knowledge;

The normal mixture is perhaps the most popular and widely studied type of the mixture families. The main reason could be its natural characteristics and flexibility to modelling the complex and non-linear pattern variations [25], which makes them an excellent choice for representing complex class-conditional pdfs (e.g. likelihood functions) in Bayesian supervised learning scenarios or prior probabilities for Bayesian parameter estimation [23]. It is simple and efficient in terms of memory, a principled model complexity selection is possible and importantly, there are theoretically guaranteed to converge algorithms for model parameter estimation [26]. In this work we propose a new unsupervised learning of Gaussian mixture models algorithm, that is flexible in terms of shape of the components, can deal with high dimensional data, is robust toward outliers and noises and with minimum dependency on initialization or prior knowledge.

## 2.2 Previous Works/Literature Review

In the probabilistic view, samples are assumed to be generated according to several probability distributions. Data points in different components are extracted from different probability distributions. They can be derived from different types of density functions, e.g., multivariate Gaussian or  $t$ -distribution, or the same families, but different parameters [27]. If the distributions are known, finding the clusters of a given dataset is equivalent to estimating the parameters of several underlying models. For off-line clustering, and more precisely to compute the parameters that define the mixture model given a finite data set, a widely used procedure is to apply the *expectation-maximization* (EM) algorithm that incrementally converges to a maximum likelihood estimate of the mixture model [26]. However in the basic EM algorithm, the complexity of the model has to be known a-priori and the whole dataset should be available. Therefore, the basic EM algorithm is

unable to deal with on-line data since it is an iterative algorithm that requires all the batch of data in each iteration. Another important restriction is the number of components of the mixture which can change under different circumstances.

Greggio et al. [28], presents a method to find the best fit in a batch of data. Starting from a fixed number of components, a split-and-merge approach together with a dissimilarity index concept is presented that adaptively updates the complexity of the mixture models. In [29], authors present an unsupervised algorithm for learning a finite mixture model from multivariate data based on the Dirichlet distribution. The proposed approach for estimating the parameters of a Dirichlet mixture is based on the maximum likelihood (ML) and Fisher scoring methods. Briefly, Zwolinski and Yang [30], and Figueiredo and Jain [19] overestimate the complexity of the model and reduce it by discarding “weak” components. Vlassis and Likas [31] use a weighted sample kurtoses of Gaussian kernels, while Verbeek et al. [32], introduces a heuristic greedy approach in which mixture components are added one at the time. A. Declercq and J. H. Piater et al. [26] presents a method to incrementally learning Gaussian mixture models (GMMs) based on a new fidelity criterion for splitting and merging mixture components. Ueda et al. proposes a split-and-merge EM algorithm to alleviate the problem of local convergence of the EM method [33]. Subsequently, Zhang et al. introduced another split-and-merge technique [34]. The split-and-merge equations show that the merge operation is a well-posed problem, whereas the split operation is ill-posed. Two methods for solving this problem are developed through singular value decomposition and Cholesky decomposition and then a new modified EM algorithm is constructed. Thrun et al. [35] presents a modified version of the EM capable of generating online 3D maps.

Essentially all previous works with GMM concentrated on non time critical applications, typically in which, historical data or some part of it, is available. Model fitting and determining the complexity of the model is performed in a full batch of data or using a relatively small training corpus. However, latest direction in robotics and computer vision is towards real-time applications (for example for human-computer interaction and on-the-fly model building) and modelling of dynamic, uncertain complex patterns which inherently involves large amounts of data. In both cases, the usual batch fitting becomes impractical and having a dynamic model is necessary. Incremental fitting of GMMs has already been addressed in the machine learning literature. Most of the existing methods assume that novel data arrives in *blocks* as opposed to a single datum at a time. Z. Zivkovic et al. [36] inspired by [19] proposed an on-line (recursive) algorithm that estimates the parameters of the mixture and simultaneously selects the number of components by starting with overestimating the complexity of the model in a small batch and searching for the *maximum a posteriori* (MAP) solution, and discarding the irrelevant components. Hall et al. [37] use a pair-wise merging of Gaussian components approach by considering volumes of the corresponding hyper-ellipsoids. Song and Wang et al. [38] propose a more principled method by using  $W$  statistic for covariance and Hotelling’s  $T^2$  statistic for mean equivalence. However, they do not fully exploit the available probabilistic information by not considering the weights of the Gaussian components. In other words, they fail to take into account the *evidence* for each component at the time of merging, as is suggested in, for example [19]. Hall, Marshal and Martin et al. [37] fail to

to make use of the existing model when the GMM corresponding to new data is fitted. This means that even if some of the new data is already explained well by the current model, the complexity of the model unnecessarily will change, in the context of the other novel data, affecting the accuracy of the fit as well as the subsequent component merging. Similar to that, in [38] the EM fitting fails to use the current model properly. The method of Hicks et al. [39] does not suffer from the same drawback. The authors propose to first “concatenate” two GMMs and then determine the complexity of the model by taking into account *all* low complexities and choosing the one that gives the largest penalized log-likelihood. A similar approach of combining Gaussian components was also described in [40, 41].

Arandjelovic et al. [42], addresses the problem of unsupervised learning of Gaussian Mixture Models (GMMs). Similar to our approach, his method works for the case when novel data points arrive one-by-one. This assumption is in contrary to many previous approaches which universally assume that new data comes in blocks representable by GMMs which are then merged with the current model estimate. In his method, two hypotheses are kept at each time and no historical data. The current fit is updated with the assumption that the number of components is fixed, which is increased or reduced when enough evidence for a new component is seen. This is deduced from the change from the oldest fit of the same complexity. This approach can be quite vulnerable to the order of arriving data. Our proposed method has tackled this problem by in fact keeping more than two hypotheses, no historical data likewise, and checking the Gaussian mixture reduction by receiving a new sample.

In this chapter, we address the on-line unsupervised learning problem of GMMs in the presence of uncertain dynamic environments, i.e., we assume that the number of Gaussian components (clusters) is not only unknown but it also can change over time. Inspired by the work in [19], namely the use of the *minimum description length* (MDL) concept, we propose a multi-hypothesis adaptive algorithm that continuously updates the number of components and estimates the model parameters as the measurements (sample data) are being acquired. The proposed algorithm has the additional feature that it relaxes “the sufficiently large data set” restriction by not requiring in fact any initial batch of data. Simulation results illustrate the performance of the proposed algorithm where it shows that indeed it is able to continuously adapt to the dynamic changes of the number of clusters and estimate the parameters of the mixture model.

## 2.3 Problem Statement

We consider the problem of determining the structure of clustered data, without prior knowledge of the number of clusters or any other information about their composition. Data are represented by a mixture model in which each component corresponds to a different cluster. Models with varying geometric properties are obtained through Gaussian components with different parametrizations and cross-cluster constraints. Unsupervised learning of Gaussian mixture models is a surprisingly difficult task. The main challenges of this problem is the model complexity selection which is required to be dynamic by the nature of the framework, without having access to historical data.

Intuitively, if the present GMM hypothesis at time  $t$ , can fully explain all the available and/or observed information up to time  $t$ , a single novel point *never* carries enough information to cause an increase in the number of Gaussian components. Another closely related difficulty lies in the *order* in which new data arrives [43]. If successive data points are always badly correlated, a considerably large number of samples needs to be seen to obtain a highly accurate model.

We now formulate the above ideas mathematically. Let  $\{\mathbf{Y}_n, n = 0, 1, 2, \dots\}$  be a discrete-time random process where for each particular time  $n$ ,  $\mathbf{Y}_n$  follows a  $K$ -component mixture of  $d$ -dimensional Gaussian with probability density function (pdf) given by

$$p(\mathbf{y}|\boldsymbol{\theta}) = \sum_{k=1}^K w^{[k]} p^{[k]}(\mathbf{y}|\boldsymbol{\theta}^{[k]}), \quad (2.1)$$

where  $\mathbf{y}$  represents one particular outcome of  $\mathbf{Y}_n$  and  $w = \{w^{[1]}, \dots, w^{[K]}\}$  is the mixing weight set that satisfies

$$\sum_{k=1}^K w^{[k]} = 1, \quad w^{[k]} > 0, \quad (2.2)$$

$K$  denotes the number of components of the mixture,  $\boldsymbol{\theta}^{[k]} = \{\mu^{[k]}, \Sigma^{[k]}\}$  is the mean and covariance matrix of the  $k^{th}$  component, with  $\boldsymbol{\theta} = \{\boldsymbol{\theta}^{[1]}, \dots, \boldsymbol{\theta}^{[K]}\}$ , and

$$p^{[k]}(\mathbf{y}|\boldsymbol{\theta}^{[k]}) = \frac{1}{(2\pi)^{d/2} |\Sigma^{[k]}|^{1/2}} \exp\left(-\frac{1}{2} (\mathbf{y} - \boldsymbol{\mu}^{[k]})^T (\Sigma^{[k]})^{-1} (\mathbf{y} - \boldsymbol{\mu}^{[k]})\right). \quad (2.3)$$

Note that for simplicity of notation we have omitted in the parameters  $K$ ,  $w$ ,  $\boldsymbol{\theta}$  their explicit dependence on the time  $n$ . As long as the parameter vector  $\boldsymbol{\theta}$  is decided, the posterior probability for assigning a data point to a cluster can be easily calculated with Bayes's theorem. Here, the mixture can be constructed with any types of components, but more commonly, multivariate Gaussian densities are used due to its complete theory and analytical tractability [44, 45].

We can now formulate the problem addressed in the paper:

*Given a sequence of observations  $Y_0, Y_1, \dots$ , find on-line, as the samples are arriving, a sequence of estimates for the parameters  $K$ ,  $w$ ,  $\boldsymbol{\theta}$  that is most likely to be in some sense close to the correct characterization of the random process  $\{\mathbf{Y}_n, n = 0, 1, 2, \dots\}$ .*

## 2.4 Preliminaries and basic results

This section presents several background results, starting with the EM algorithm, that are needed to understand the proposed on-line unsupervised learning algorithm.

### 2.4.1 The Basic Expectation-Maximization (Off-line) Algorithm

Finite mixtures presume that each observation is generated from one of a set of alternative random sources, and infer these sources and identify which source is most likely to have produced each observation. Finite mixtures are fundamental models in pattern recognition, classification

and clustering analysis applications, where the mixture model uses assumed component densities to represent individuals that have the same characteristics in the total population. Among the mixture families, the normal mixture is perhaps the most studied type, due to its naturalness and some interesting properties of the exponential family to which it belongs [25, 46]. The unsupervised learning of finite mixtures is usually realized via a maximum likelihood estimator. The Expectation Maximization (EM) algorithm, which was named by Dempster et al. in [26], is conveniently suitable for obtaining the Maximum Likelihood Estimation (MLE) of a finite mixture problem. Thus, it has found application in almost all statistical contexts and in almost all fields in which statistical techniques may be applied, such as economics, bioinformatics, medical imaging, etc [23].

The original EM algorithm works in a batch manner, that is, the parameters of mixtures are not updated until a thorough scan of available data samples is completed. In real-time learning systems, a stable dataset for off-line estimation does not exist. On-line algorithms are hence devised to deal with real-time applications. In contrast to the traditional version of EM, on-line EM variants can flexibly update the parameters of a finite mixture as soon as a new sample is observed, and meanwhile does not need to store the large number of data samples that is indispensable for batch learning algorithms. An important on-line EM variant elaborated in our work is proposed by Titterington in [47]. It is said to be closely related to Stochastic Approximation Theory [48]. A theoretical proof for utilizing this EM variant is provided in [46]. The EM algorithm is extremely suitable for deriving Maximum Likelihood Estimation (MLE) or Maximum A Posteriori (MAP) estimations. Moreover, it has long been recognized to be useful at dealing with incomplete data problems and mixtures of densities. Although the EM algorithm is reported to have only first-order convergence (slower than the methods of scoring or Newton-Raphson), it has some desirable features that can compensate for this drawback. For one thing, the EM algorithm automatically satisfies the probabilistic constraints of mixture problems [49]. For another, the EM algorithm has exhibited reliable global convergence behaviour under proper conditions [50]. Finally, EM is easy to realize and the computation per iteration is relatively low.

For finite mixture models, given a set of  $n$  independent and identically distributed samples  $Y = \{Y_1, \dots, Y_n\}$ , the log-likelihood corresponding to a  $K$ -component mixture where all the components are  $d$ -dimensional Gaussian is [23]

$$\ell = \log p(Y|\theta) = \log \prod_{i=1}^n p(Y_i|\theta) = \sum_{i=1}^n \log \sum_{k=1}^K w^{[k]} p(Y_i|\theta^{[k]}) \quad (2.4)$$

Maximum likelihood (ML) estimation is an important statistical approach for parameter estimation [51] and it considers the best estimate as one that maximizes the probability of generating all the observations, which is given by the joint density function, Equation (2.4). The best estimate can be achieved by solving the log-likelihood equations  $(\partial \ell(\theta)) / \partial \theta^{[k]} = 0$ . Unfortunately, since the solutions of the likelihood equations can not be obtained analytically in most circumstances [19, 52], iteratively suboptimal approaches are required to approximate the ML estimates. Among these methods, the expectation-maximization (EM) algorithm is the most popular [53].

EM regards the dataset as incomplete and divides each data point into two parts, observable features and missing data.

It is well-known that the *maximum likelihood* (ML) or *maximum a posteriori* (MAP) estimates can not be found analytically [23, Ch. 9]. An elegant and powerful method for finding ML or MAP solutions for models with latent variables is called the *expectation-maximization* or EM algorithm [26], [23, Ch. 9]. The EM is an easily implementable algorithm that iteratively increases the posterior density or likelihood function. In order to describe the EM, we need to introduce for each observation  $Y_i$ , a discrete unobserved indicator vector  $Z_i = [Z_i^{[1]}, \dots, Z_i^{[K]}]$ . This vector specifies from which component the observation  $Y_i$  was drawn, i.e., if  $Z_i^{[k]} = 1$  and  $Z_i^{[p]} = 0$  for  $k \neq p$ , then this means that the sample  $Y_i$  was produced by the  $k^{th}$  component. Hence, the complete log-likelihood function (i.e. the one from which we could estimate  $\theta, w$  if the *complete* data  $X = \{Y, Z\}$  was observed [26, 54]) can be written as a product

$$\log p(Y, Z|\theta) = \sum_{i=1}^n \sum_{k=1}^K Z_i^{[k]} \log [w^{[k]} p(Y_i|\theta^{[k]})] \quad (2.5)$$

The basic EM algorithm works with an incomplete batch sample set to estimate the missing or potential parameters. It runs over the whole data set  $Y$  and until some convergence criterion is met, iteratively produces a sequence of estimates  $\hat{\theta}_m, \hat{w}_m, m = 0, 1, 2, \dots$  by alternatively applying two steps:

**E-Step:** Computes the expected value of conditional probability of unobserved or underlying parameters given the observed data at the present parameter setting. Given  $Y$  and the current estimates  $\hat{\theta}_m, \hat{w}_m$  and by considering the fact that  $\log p(Y, Z|\theta)$  is linear with respect to the missing  $Z$ , the so-called  $Q$ -function computes the conditional expectation of the complete log-likelihood function as

$$Q(\theta, \hat{\theta}_m) = \mathbb{E} [\log p(Y, Z|\theta) | Y, \hat{\theta}_m] = \log p(Y, \Gamma|\theta), \quad (2.6)$$

where  $\Gamma \equiv \mathbb{E}[Z|Y, \hat{\theta}_m, \hat{w}_m]$  is the a conditional expectation that each observation is generated by which component. Since the elements of  $Z$  are binary, as mentioned in [19], their conditional expectations are given by

$$\Gamma_i^{[k]} = \mathbb{E} [Z_i^{[k]} | Y, \hat{\theta}_m] = \Pr[Z_i^{[k]} = 1 | Y_i, \hat{\theta}_m] = \frac{\hat{w}_m^{[k]} p(Y_i | \hat{\theta}_m^{[k]})}{\sum_{k=1}^K \hat{w}_m^{[k]} p(Y_i | \hat{\theta}_m^{[k]})}, \quad (2.7)$$

where  $\hat{w}_m^{[k]}$  corresponds to the a priori probability that  $Z_i^{[k]} = 1$  in the  $m$ -th iteration of the basic EM algorithm over  $Y$ , while  $\Gamma_i^{[k]}$  is the a posteriori probability that  $Z_i^{[k]} = 1$ , after observing  $Y_i$ .

**M-Step:** Maximizes the log-likelihood of the observed data on the basis of the result from

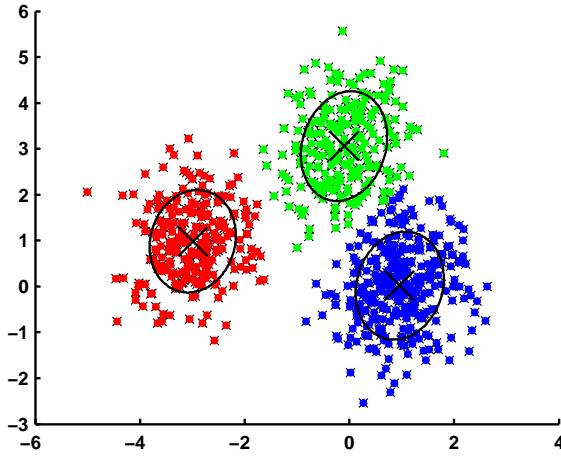


Figure 2.1: An example of good initialization

the former stage. Maximizing  $Q$  by constructing a Lagrangian function to update the parameter estimation for  $\theta$ .

$$\hat{\theta}_{m+1} = \arg \max_{\theta} Q(\theta, \hat{\theta}_m), \quad (2.8)$$

for the ML estimation. In the case of MAP criterion, instead of  $Q(\theta, \hat{\theta}_m)$ , we need to maximize  $\{Q(\theta, \hat{\theta}_m) + \log p(\theta)\}$ .

It is evident that each sample  $Y_i$  influences the parameter estimation by directly taking part in computation of the *a posteriori* probability in Equation (2.7), with all the samples doing the same thing simultaneously. Figure 2.1 shows a successful implementation of the standard EM algorithm over a synthetic dataset composed of three components.

#### 2.4.1.1 Drawbacks of the EM

Basically, all deterministic algorithms for fitting mixtures, especially when the true number of components is unknown [19], which use the EM algorithm have two major drawbacks:

- EM is highly dependent on initialization. It is a local method that is, it tends to converge to a local optimum and not necessarily the global one, thus it is sensitive to the initialization because the likelihood function of a mixture model is not uni-modal. Figure 2.2(a) shows a fail in finding the true component due to “bad” initialization.
- It may converge to the boundary of the parameter space (where the likelihood is unbounded) leading to meaningless estimates, Figure 2.2(b). For example, when fitting a Gaussian mixture with unconstrained covariance matrices, one of the  $w_{th}^{[k]}$  may approach zero and the corresponding covariance matrix may become arbitrary close to singular. When the number of components assumed is larger than the optimal/true one, this tends to happen frequently, thus being a serious problem for methods that require mixture estimates for various values of  $K$ .

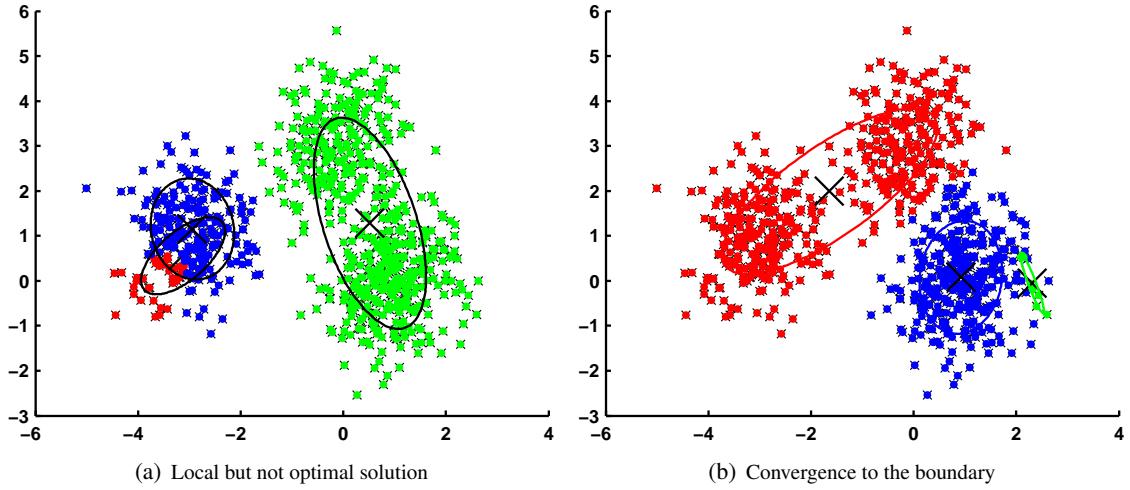


Figure 2.2: EM drawbacks

To tackle these problems a couple of time-consuming strategies can be done. Using multiple random starts and choosing the final model with the highest likelihood, or using the clustering algorithms such as K-means and feed the output of them as the initialization step in the EM algorithm.

Since in many real world applications, the complexity of the model is unknown and it may change over time, and also due to memory and time constraints, for these types of applications, a modified version of the EM algorithm should be used to accommodate those issues and also to be applicable in an on-line context.

#### 2.4.1.2 How Many Clusters?

The clustering process partitions data into an appropriate number of subsets. Although for some applications, users can determine the number of clusters,  $K$ , in terms of their expertise, under more circumstances, the value of  $K$  is unknown and needs to be estimated exclusively from the data themselves. Many clustering algorithms ask  $K$  to be provided as an input parameter, and it is obvious that the estimation of  $K$  can directly affect the likelihood value and the quality of the fitting model. Overestimating the complexity of the model can make it hard to interpret and analyse the results, while a division with too few components, also known as underestimating the model, may cause the loss of information and misleads the final decision. Dubes called the problem of determining the best  $K$  as “the fundamental problem of cluster validity” [55].

A large number of attempts have been made to estimate the appropriate  $K$  and some of representative examples are illustrated in the following

- Visualization of the dataset

For the data points that can be effectively projected onto two-dimensional Euclidean space, direct observation can give intuition to estimate the complexity of the model. A histogram or

scatter plot can be insightful in these cases. Although in many real world complex datasets, this method fails to give us enough evidence to determine the optimal  $K$ . As a result, the effectiveness of the strategy is only restricted to a small scope applications.

- Construction of certain indices or stopping rules

These indices usually emphasize the compactness of intra-cluster and isolation of inter-cluster. Mathematically it is easy to show that the minimizing of the intra-class covariance is the same as maximizing the inter-class covariance. Consider the comprehensive effects of several factors, including the defined squared error, the geometric or statistical properties of the data, the number of patterns, the dissimilarity (or similarity) and the number of clusters. Milligan and Cooper compared and ranked 30 indices according to their performance over a series of artificial datasets [56]. Among these indices, the Caliński and Harabasz index,  $CH(K)$ , achieved the best performance [55]. The  $K \in [1, \dots, M]$ , that maximizes the value of  $CH(K)$  is selected as the optimal. The drawback of this method is the dependency of  $CH(K)$  to the data. Therefore, the good performance of this criterion for a certain data does not guarantee the same behaviour with different data. As pointed out by Everitt, Landau and Leese, "it is advisable not to depend on a single rule for selecting the number of groups, but to synthesize the results of several techniques" [57].

- Optimization of some criterion functions under probabilistic mixture-model framework

In a statistical framework, finding the correct complexity for a model, is equivalent to fitting a model with observed data and optimizing some criterion [52]. As mentioned before, the EM algorithm is a widely used tool to estimate the model parameters for a given  $K$ , which goes through a predefined range of values. The optimal value of  $K$  is the one that maximizes (or minimizes) some defined criterion. Smyth et al. [58] proposes a Monte-Carlo cross validation method, which randomly separates the data into training and test sets  $M$  times according to a certain function  $\beta$  ( $\beta = 0.5$  works well from the empirical results). The  $K$  is selected either directly based on the criterion function or some prior knowledge. In literature, a large number of criteria with combination with information theory concepts have been proposed. Typical examples are as follow:

- Akaike's information criterion (AIC) where  $K$  is selected with the minimum value of  $AIC(K)$  [59, 60].
- Bayesian inference criterion (BIC) The optimal  $K$  is one that maximizes the value of  $BIC(K)$  [61, 62].

The BIC can be used to select an optimal model for the covariance matrices [63].

- Minimum description length criterion (MDL)

Is a widely used criterion, e.g. [42, 19, 64], and we focus on this in Section 2.4.3

- Other heuristic approaches based on a variety of techniques and theories

Girolami et al. [65], propose an eigenvalue decomposition on the kernel matrix in the high-dimensional feature space. The dominant  $K$  components were used in the decomposition

summation as an indication of the possible existence of  $K$  clusters. A scale-based method, in which the distance from a cluster centroid to other neighbouring clusters is considered in [66].

The idea behind these, is that we penalize the models according to the number of parameters that they have. Therefore, a model with more parameters, is a more complex model, thus it is penalized more.

AIC, BIC and MDL share a lot in common with maximum likelihood Bayesian approach, where we want to maximize the probability of a model given the data, e.g. [19].

Besides the previous methods, there are some adaptive methods, such as constructive clustering algorithms, that are capable of changing the complexity of a model in time. Adaptive Resonance Theory (ART) network is capable of continuous training in real time. ARTs increase the complexity of the model, only when the confidence level of the match between the input pattern and the expectation drops below some certain threshold [67]. The Robust Competitive Agglomeration (RCA) starts by overestimating the complexity of the model to reduce the sensitivity to initialization, and determines the actual number of clusters by a process of competitive agglomeration. It processes all the components in stages, and components that become “weak”, lose in the competition discarded and the weights of the others renormalized [68]. The generalization of this method is in [69], which the trade-off is between the complexity of the model and some fidelity index. Another example of this process is in [19], which attains the number of components by balancing the effect between the complexity and the minimum description length criterion. To review further methods, interested readers are referred to [27, 70].

#### 2.4.2 Titterington’s On-line Algorithm for a Multivariate Normal Mixture

Nowadays, huge datasets are routinely encountered in real applications, rendering a thorough scan of all the data costly and impractical. Moreover, real-time systems receive data streams from sensors and store them in such a manner that the oldest data are discarded to free up capacity for the newest one. Therefore, searching for an on-line version of EM has aroused considerable interest in both statistical and pattern recognition communities. Neal and Hinton [71] proposed an incremental EM algorithm by updating some sufficient statistics in the “E” step and then estimating parameters from those statistics in the “M” step. Although the method is still confined to batch dataset, the convergence is claimed to be faster than traditional EM. A step-wise on-line EM algorithm was proposed in [72] for normalized Gaussian networks with a discussion of the conditional equivalence of the on-line and batch EM. It has been long recognized that recursive EM shares much in common with stochastic approximation theory, both working favourably with finite mixture densities. The most important work in this respect is Titterington’s on-line algorithm [47] on which we are going to concentrate. Since the on-line algorithm is closely related to the basic EM algorithm, the notations of incomplete and complete data functions are considered carefully in the descriptions of their respective frameworks. We will first present some background about incomplete and complete data functions in the light of [26], before introducing the algorithm.

This section is mostly based on [46], where the application of an on-line EM variants for multi-variate normal mixture model in background learning and moving foreground detection was investigated.

**Definition 1** Consider two sample spaces  $\mathbf{X}$  and  $\mathbf{Y}$  and a many-to-one mapping from  $\mathbf{X}$  to  $\mathbf{Y}$ . All the observed data samples  $y$  are from  $\mathbf{Y}$ , while the actual data vector  $x \in \mathbf{X}$  is invisible and is only reached indirectly from  $y$ . In the mapping of  $y = y(x)$ , we refer to  $x$  as the complete data and  $y$  as the incomplete data. By postulating a mathematical representation of the complete data density  $f(x|\theta)$  dependent on  $\theta$ , a connection between  $f(x|\theta)$  and the incomplete date density  $p(y|\theta)$  is formed as

$$p(y|\theta) = \int f(x|\theta)dx \quad (2.9)$$

The choice of  $f(x|\theta)$  is generally not unique. Since in this work  $p(y|\theta)$  is a family of multivariate normal models defined by Equation (2.1), the complete data vector  $x$  is considered as a concatenation of  $y$  and a Kronecker Delta vector  $\delta = [\delta^{[1]} \dots \delta^{[K]}]^T$  in which exactly one element is 1 while others are all zeros. We specify  $f(x|\theta)$  as

$$f(x|\theta) = \prod_{k=1}^K [w^{[k]} p^{[k]}(y|\theta^{[k]})]^{\delta^{[k]}} \quad (2.10)$$

As mentioned earlier, the original EM algorithm works in a batch manner. In contrast to the traditional version of the EM, on-line EM variants can flexibly update the parameters of  $\mathbf{Y}_n$  as soon as a new sample is observed.

The on-line recursive parameter estimation proposed by Titterington [47] is a popular method for solving mixture estimation problems. The recursive procedure proposed takes the form

$$\theta_{n+1} = \theta_n + n^{-1} \mathbf{I}_c^{-1}(\theta_n) \mathbf{v}_p(Y_{n+1}, \theta_n), \quad (2.11)$$

where  $\mathbf{v}_p(Y_{n+1}, \theta_n)$  is the score function of  $p(y|\theta)$  given the newest observed value and the current parameter estimation, and it is defined as

$$\mathbf{v}_p = \nabla_\theta \log p(Y|\theta) \quad (2.12)$$

where  $\nabla_\theta$  denotes the gradient of  $\theta$ . In Equation (2.11),  $n$  is the time of the last received observation, and can be interpreted as the sequence of estimation  $m$  in the basic EM algorithm, since during the on-line task, parameter estimations are being updated by every new-comer point.  $\mathbf{I}_c(\theta)$  is the Fisher Information Matrix (FIM) for the complete likelihood:

$$\mathbf{I}_c(\theta) = \mathbb{E}_\theta[v_c v_c^T]. \quad (2.13)$$

where,  $v_c$  is the score function of the complete-data likelihood  $f(x|\theta)$  defined as

$$v_c = \nabla_\theta \log f(x|\theta) \quad (2.14)$$

under the constraint that all derivatives and expected values exist. For further details the reader is referred to [46].

**Titterington's On-line Algorithm:** The Titterington-type on-line recursive parameter estimation for multivariate normal mixtures are given by

$$\mu_{n+1}^{[k]} = \mu_n^{[k]} + \frac{1}{n} \frac{\Gamma_{n+1}^{[k]}}{w_n^{[k]}} (Y_{n+1} - \mu_n^{[k]}) \quad (2.15)$$

$$\Sigma_{n+1}^{[k]} = \Sigma_n^{[k]} + \frac{1}{n} \frac{\Gamma_{n+1}^{[k]}}{w_n^{[k]}} \left[ (Y_{n+1} - \mu_n^{[k]}) (Y_{n+1} - \mu_n^{[k]})^T - \Sigma_n^{[k]} \right] \quad (2.16)$$

$$w_{n+1}^{[k]} = w_n^{[k]} + \frac{1}{n} (\Gamma_{n+1}^{[k]} - w_n^{[k]}) \quad (2.17)$$

where  $Y_{n+1}$  is the new observation,  $\Gamma_{n+1}^{[k]}$  is the a posteriori probability described by Equation (2.7),  $n$  is the time,  $w_{n+1}^{[k]}$  is the mixing weight of  $k^{th}$  component at time  $n+1$ ,  $\mu_{n+1}^{[k]}$  is the updated mean of  $k^{th}$  component and  $\Sigma_{n+1}^{[k]}$  is the updated covariance of  $k^{th}$  component. For more details and the derivation of the formulas see [46]. Before we introduce the proposed algorithm, in the next section, first we briefly describe the criterion that is used in [19] in order to find the number of components in a batch of data. Later, we explain how to use this criterion in real time applications.

### 2.4.3 The Minimum Description Length (MDL) Principle

Two powerful method of *inductive inference*, the basis of statistical modelling, pattern recognition, and machine learning are the Minimum Description Length (MDL) and Minimum Message Length (MML) approaches.

The main difference between these two is the interpretation of prior probabilities. Used as a proxy for the complexity of the model, the MDL principle is a two-part code that maximizes the probability of a model given the data. On the other hand, the MML principle is a one-part code that do not determine the complexity of a model but instead provide a predictive model for the data [73, 74]. The task of inductive inference is to find patterns or regularities underlying some given set of data. These patterns are then used to understand the inner nature of the data or to classify or predict future data.

The MDL principle is rooted in the fact that any regularity in a given set of data can be used to compress it, i.e., using less symbols than the number of symbols required to describe the data literally. The more regularities the data have, the more it can be compressed. Formalizing this idea, which is just a version of famous Ockham's razor<sup>1</sup>, leads to a theory that is relevant to all kinds of high entropy reasoning under uncertainty, including inductive inference. In contrast to most other approaches in this field, MDL has its roots in theoretical computer science rather than statistics. The MDL principle has mainly been developed by Risannen [75], having important precursors in the works of Solomonoff and Wallace and Boulton [74].

---

<sup>1</sup>Ockham's razor says among equally valid alternatives, the best explanation is the simplest one

Here, we are concentrated on the application of MDL to *model selection*, the task of deciding which of several hypotheses best describes the data at hand [76, 77].

In the proposed method, dynamic model complexity estimation is based on the MDL criterion. Briefly, MDL assigns to a model a cost related to the amount of information necessary to encode the model and the data *given* the model (Equation (2.18)).

The rationale behind minimum encoding length criteria (like MDL and MML) is: if you can build a short code for your data, this means that you have a good data generation model. To formalize this idea, consider some dataset  $Y$ , known to have been generated according to  $p(Y|\theta)$ , which is to be encoded and transmitted. Following Shannon theory [78, 79], the shortest code length (measured in bits, if base-2 logarithm is used, or in *nats*, if natural logarithm is adapted [80]) for  $Y$  is  $[-\log p(Y|\theta)]$ , where  $\lceil a \rceil$  denotes “the smallest integer no less than  $a$ ”. Since for even moderately large datasets  $-\log p(Y|\theta) \gg 1$ , the  $\lceil \cdot \rceil$  operator is usually dropped. If  $p(Y|\theta)$  is fully known to both the transmitter and receiver, they can both build the same code and communication can proceed. However, if  $\theta$  is a priori unknown, the transmitter has to start by estimating and transmitting  $\theta$ . This leads to a two-part message, whose total length is given by

$$\text{Length}(\theta|Y) = \text{Length}(\theta) + \text{Length}(Y|\theta) \quad (2.18)$$

All minimum encoding length criteria (like MDL and MML) state that the parameter estimate is the one that minimizes  $\text{Length}(\theta, Y)$ .

A key issue of this approach, which the several flavours of the minimum encoding length principle (e.g., MDL and MML) address differently, is that since  $\theta$  is a vector of real parameters, a finite code-length can only be obtained by quantizing  $\theta$  to finite precision.

The central idea involves the following trade off. Let  $\tilde{\theta}$  be a quantized version of  $\theta$ . If a fine precision is used,  $\text{Length}(\tilde{\theta})$  is large, but  $\text{Length}(Y|\tilde{\theta})$  can be made small because  $\tilde{\theta}$  can be very close to the optimal value. Conversely, with a coarse precision,  $\text{Length}(\tilde{\theta})$  is small but  $\text{Length}(Y|\tilde{\theta})$  can be very far from the optimality. There are several ways to formalize and solve this trade off; see [81, 78] for a comprehensive review and pointers to the literature.

The fact that the data itself may also be real-valued does not cause any difficulty; simply truncate  $Y$  to some arbitrary fine precision  $\delta$  and replace the density  $p(Y|\theta)$  by the probability  $p(Y|\theta)\delta^d$  ( $d$  is the dimensionality of  $Y$ ). The resulting code-length is  $-\log p(Y|\theta) - d \log \delta$ , but  $-d \log \delta$  is an irrelevant additive constant. The particular form of MML approach herein adopted is derived in [19] and leads to the following criterion (where the minimization with respect to  $\theta$  is to be understood as simultaneously in  $\theta$  and  $c$ , the dimension of  $\theta$ ):

$$\hat{\theta} = \arg \min_{\theta} \left\{ -\log p(\theta) - \log p(Y|\theta) + \frac{1}{2} \log |\mathbf{I}(\theta)| + \frac{c}{2} \left( 1 + \log \frac{1}{12} \right) \right\} \quad (2.19)$$

where<sup>2</sup>  $\mathbf{I}(\theta) = -\mathbb{E}[D_{\theta}^2 \log p(Y|\theta)]$  is the expected Fisher information matrix, and  $|\mathbf{I}(\theta)|$  denotes its determinant. The MDL criterion (which formally, though not conceptually, coincides with

---

<sup>2</sup>Here,  $D_{\theta}^2$  denotes the matrix of second derivatives, or also known as the Hessian matrix

BIC) can be obtained as an approximation to Equation (2.19). Start by assuming a flat prior  $p(\theta)$  and drop it. Then, since  $\mathbf{I}(\theta) = n\mathbf{I}^{(1)}(\theta)$  (where  $\mathbf{I}^{(1)}(\theta)$  is the Fisher information corresponding to a single observation),  $\log|\mathbf{I}(\theta)| = c \log n + \log|\mathbf{I}^{(1)}(\theta)|$ . For large  $n$ , drop the order 1 terms  $\log|\mathbf{I}^{(1)}(\theta)|$  and  $\frac{c}{2}(1 + \log \frac{1}{12})$ . Finally, for a given  $c$ , take  $-\log p(Y|\theta) \simeq -\log p(Y|\hat{\theta}(c))$ , where  $\hat{\theta}(c)$  is the corresponding ML estimate. The result is the well-known MDL criterion,

$$\hat{c}_{MDL} = \arg \min_c \left\{ -\log p(Y|\hat{\theta}(c)) + \frac{c}{2} \log n \right\} \quad (2.20)$$

whose two-part code interpretation is clear: the data code-length is  $-\log p(Y|\hat{\theta}(c))$ , while each of the  $c$  components of  $\hat{\theta}(c)$  requires a code-length proportional to  $(1/2)\log n$ . Intuitively, this means that the encoding precision of the parameter estimates is made inversely proportional to the estimation error standard deviation, which, under regularity conditions, decreases with  $\sqrt{n}$ , leading to the  $(1/2)\log n$  term [82].

#### 2.4.3.1 The Proposed Criterion for Mixtures

For mixtures,  $\mathbf{I}(\theta)$  cannot, in general, be obtained analytically [52, 83, 84]. To side-step this difficulty, we replace  $\mathbf{I}(\theta)$  by the complete-data Fisher information matrix  $\mathbf{I}_c(\theta) = \mathbb{E}[D_\theta^2 \log p(Y, Z|\theta)]$ , which upper-bound  $\mathbf{I}(\theta)$ , see [52].  $\mathbf{I}_c(\theta)$  has block-diagonal structure

$$\mathbf{I}_c(\theta) = n\text{block} - \text{diag} \left\{ w^{[1]}\mathbf{I}^{(1)}(\theta^{[1]}), \dots, w^{[K]}\mathbf{I}^{(1)}(\theta^{[K]}), \mathbf{M} \right\} \quad (2.21)$$

where  $\mathbf{I}^{(1)}(\theta^{[m]})$  is the Fisher matrix for a single observation known to have been produced by the  $m$ th component, and  $\mathbf{M}$  is the Fisher matrix of a multinomial distribution (recall that  $|\mathbf{M}| = (w^{[1]}w^{[2]}\dots w^{[K]})^{-1}$ ) [52]. The approximation of  $\mathbf{I}(\theta)$  by  $\mathbf{I}_c(\theta)$  becomes exact in the limit of non overlapping components. We adopt a prior expressing lack of knowledge about the mixture parameters. Naturally, we model the parameters of different components as a priori independent and also independent from the mixing probabilities, i.e.,

$$p(\theta) = p(w^{[1]}, \dots, w^{[K]}) \prod_{m=1}^K p(\theta^{[m]}) \quad (2.22)$$

For each factor  $p(\theta^{[m]})$  and  $p(w^{[1]}, \dots, w^{[K]})$ , we adopt the standard non informative Jeffreys' prior (see, for example [85])

$$p(\theta^{[m]}) \propto \sqrt{|\mathbf{I}^{(1)}(\theta^{[m]})|} \quad (2.23)$$

$$p(w^{[1]}, \dots, w^{[K]}) \propto \sqrt{|\mathbf{M}|} = (w^{[1]}w^{[2]}\dots w^{[K]})^{-1/2} \quad (2.24)$$

for  $0 < w^{[1]}, w^{[2]}, \dots, w^{[K]} < 1$  and  $w^{[1]} + w^{[2]} + \dots + w^{[K]} = 1$ . With these choices and noticing that for a  $K$ -component mixture,  $c = NK + K$ , where  $N$  is the number of parameters specifying each

component, i.e., the dimensionality of  $\theta^{[m]}$ , Equation (2.19) becomes

$$\hat{\theta} = \arg \min_{\theta} \mathcal{L}(\theta, Y) \quad (2.25)$$

Now we can define the MDL intuition as follow:

Given a set of hypotheses  $\mathcal{H} = \{\mathcal{H}_1, \mathcal{H}_2, \dots\}$  and a data set  $Y$ , the goal is to find the hypothesis or combination of hypotheses in  $\mathcal{H}$  that most compress  $Y$ . For the particular case of a data set  $Y = \{Y_1, \dots, Y_n\}$ , that has been generated according to Equations (2.1-2.3), which has to be encoded and transmitted, the description length can be obtained as follow [19, 86]:

$$\mathcal{L}(\theta, Y) = \frac{N}{2} \sum_{m=1}^K \log\left(\frac{nw^{[m]}}{12}\right) + \frac{K}{2} \log \frac{n}{12} + \frac{K(N+1)}{2} - \log p(Y|\theta) \quad (2.26)$$

Apart from the order-1 term  $\frac{K(N+1)}{2}(1 - \log 12)$ , this criterion has the following intuitively appealing interpretation in the spirit of the standard two-part code formulation of MDL and MML:

- As usual,  $-\log p(Y|\theta)$  is the code-length of the data.
- The expected number of data points generated by the  $m$ th component of the mixture is  $nw^{[m]}$ ; this can be seen as an effective sample size from which  $\theta^{[m]}$  is estimated; thus, the “optimal” (in the MDL sense) code length for each  $\theta^{[m]}$  is  $(N/2) \log(nw^{[m]})$ .
- According to [19], we assume that the minimum number of points needed to support a  $d$ -dimensional Gaussian is  $N/2$  where  $N$  is a constant that grows quadratically with the dimension  $d$  of the data and for a case of free covariance matrix equals  $(d + d(d+1)/2)$ .
- The  $w^{[m]}$ s are estimated from all the  $n$  observations, giving rise to the  $(K/2) \log(n)$  term. As mentioned before, all  $w^{[m]}$ s in Equation (2.26) are considered to be non zero. Otherwise the objective function does not make sense if any one of them becomes zero (it becomes  $-\infty$ ). This is the final cost function, whose minimization with respect to  $\theta$  will constitute our mixture estimate.

#### 2.4.4 Gaussian Mixture Reduction

Gaussian mixture reduction algorithms are useful in many areas, including in error correction codes, supervised learning of multimedia data, distributed data fusion, and pattern recognition, to name a few. In many target tracking problems, data association uncertainty manifests itself as multiple hypotheses, each corresponding to a component in a multivariate Gaussian mixture posterior distribution. It is naturally of interest to reduce their number with minimal loss of fidelity. The simplest method of managing the complexity is to eliminate Gaussian components having low probabilities, as used, for example, in the track or hypothesis-oriented Multi-Hypothesis Tracker (MHT) [87]. Alternatively, instead of pruning components of the Gaussian mixture, one could merge components according to a similarity measure. Hypothesis merging can be considered more

attractive than (MHT-style) pruning, since the information lost in removing mixture elements is, in some sense, preserved in the uncertainty (covariances) of those retained. Salmond is among the first to consider a tracker based on merging hypotheses according to an ad-hoc similarity measure [88]. Such trackers, which are based upon hypothesis merging, may be considered to be variants of the hypothesis-oriented MHT or as multimodal generalizations of the Joint Probabilistic Data Association Filter (JPDAF), as described, for example, in [89]. More advanced techniques for Gaussian mixture reduction optimize the parameters of the reduced mixture according to a global optimality criterion. One of the first uses of optimization-based mixture reduction was by Scott and Szewczyk [90], who introduced the so-called  $L_2E$  measure of similarity as well as a correlation measure of similarity. The  $L_2E$  measure is more commonly known as the Integral Squared Error (ISE). Next, we introduce of a number of greedy initialization algorithms for mixture reduction. Most greedy algorithms for Gaussian mixture reduction try to decide which components of a Gaussian mixture should be merged or pruned in order to reduce the mixture to the desired number of components.

#### 2.4.4.1 Component extinction

Pruning is the simplest approach to mixture reduction. Given a Gaussian mixture consisting of  $K$  Gaussian components, one can discard the  $K - L$  components having the lowest cost (according to some measure), and then renormalizing the weights of the remaining components.

An important feature of Algorithm 1 (which is the proposed Unsupervised Learning of Gaussian Mixture Models algorithm that will be described in Section 2.5) is that it performs component annihilation, thus being an explicit rule for reducing the complexity of the GMM hypothesis. According to [19], we assume that the minimum number of points needed to support a  $d$ -dimensional Gaussian is  $N/2$  where  $N$  is a constant that grows quadratically with the dimension  $d$  of the data and for a case of free covariance matrix equals  $(d + d(d + 1)/2)$ . Notice that this approach can tackle overfitting issue and prevents the algorithm from approaching to the boundary of the parameter space: When one of the components does not have enough *evidence*, meaning that it is not supported by the data, it is simply annihilated and the weights of other components are normalized to satisfy the Equation (2.2).

However, this pruning has proven inferior to more sophisticated greedy methods [91]. Instead of pruning, one can perform mixture reduction utilizing merging, whereby the merging of the components comes from taking an expected value across the set of components that are to be merged, as is described in the sequel.

#### 2.4.4.2 Merging two components

From Equation (2.4) it can be seen that the log-likelihood increases by choosing a more complex model, i.e. with higher number of components to estimate  $\mathbf{Y}$  in time. On the other hand, Equation (2.26) shows that more complex model demands higher description length in turn.

Therefore at each time we check if we can reduce the complexity of the best hypothesis  $\mathcal{H}_1$  by

merging two most similar components in that hypothesis (in case the number of components is more than one). This can be interpreted as the over-fitting problem because instead of describing the random process  $\mathbf{Y}$ , the over-fitted result will describe the set of observations  $Y$ .

One way to tackle this problem is to employ a model complexity reduction scheme by reducing the GMM to lower number of components. Diverse algorithms in components reduction can be found in the literature. The goal is to maintain the mean and the variance of the original mixture or at least with a negligible deviation so that the resulting GMM should properly represent the structure of the original mixture [92]. A common approach is successively to merge pairs of components, replacing the pair with a single Gaussian component whose moments up to second order match those of the merged pair. Salmond [88] and Williams [93, 94] have each proposed algorithms along these lines, but using different criteria for selecting the pair to be merged at each stage. Salmond [88], proposed an algorithm in which the mixture reduction happens by repeatedly merging the two most similar components. His criterion of similarity, based on concepts from the statistical analysis of variance, seeks to minimise the increase in “within-component” variance resulting from merging two chosen components. The proposed dissimilarity measure has two properties that may be considered undesirable. First, the measure depends on the mean of the components, but not on their individual covariance matrices, leading to merging the pair of components with the closest means even if their covariance matrices are very different. The second drawback arises from the fact that inclusion of a new component, especially if it is far remote from the existing components, can greatly affect the merging candidates by increasing the overall covariance in the mixture [95].

Williams [93, 94], proposed an integrated squared difference (ISD) dissimilarity measure to evaluate the difference between two arbitrary Gaussian mixture in closed form. The ISD criterion circumvents both of the drawbacks of the Salmond’s method. On the other hand, it is more time consuming than the Salmond method. The Williams algorithm has its own puzzling behaviour, specially when the components are radially symmetric or when the system state vector has high dimensionality [95].

**Kullback-Leibler** The Kullback-Leibler (KL) dissimilarity measure  $B$ , is a non-symmetric measure of the difference between two probability distributions  $g_1$  and  $g_2$ .

At each iteration of the algorithm outlined in Algorithm 1, we wish to choose two components from the mixture for merging. Our ultimate objective is to find a weighted mixture of  $K - 1$  Gaussian components in such a way as to keep the KL discrimination of the post-merged mixture from the original pre-merged  $K$ -component mixture as small as possible, subject to being able to accomplish this with a method that is computationally reasonably fast. A reasonable criterion, therefore, is to choose two components in such a way as to minimise the KL discrimination of the mixture after the merge from the mixture before the merge. As mentioned in [95], unfortunately there appears to be no closed-form expression for the KL discrimination of one (nontrivial) Gaussian mixture from another. (This fact deterred Williams [93] from pursuing a cost measure based on KL discrimination; were it not for this, he says it would be the “ideal cost function” for Gaussian

mixture reduction). However, Runnals et al. [95] provided an upper bound on the discrimination of the mixture after the merge from the mixture before the merge.

The dissimilarity measure  $B((\mu^{[g_1]}, \Sigma^{[g_1]}, w^{[g_1]}), (\mu^{[g_2]}, \Sigma^{[g_2]}, w^{[g_2]}))$  is defined as

$$\begin{aligned} 2B\left((\mu^{[g_1]}, \Sigma^{[g_1]}, w^{[g_1]}), (\mu^{[g_2]}, \Sigma^{[g_2]}, w^{[g_2]})\right) &= \text{tr}(\Sigma^{[g_12]}^{-1} \check{\Sigma}^{[g_12]}) \\ &+ (w^{[g_1]} + w^{[g_2]}) \log \det(\Sigma^{[g_12]}) - w^{[m]} \log \det(\Sigma^{[g_1]}) - w^{[g_2]} \log \det(\Sigma^{[g_2]}) \end{aligned} \quad (2.27)$$

where

$$\check{\Sigma}^{[g_12]} = w^{[g_1]} \Sigma^{[g_1]} + w^{[g_2]} \Sigma^{[g_2]} - (w^{[g_1]} + w^{[g_2]}) \Sigma^{[g_12]} + \frac{w^{[g_1]} w^{[g_2]}}{w^{[g_1]} + w^{[g_2]}} (\mu^{[g_1]} - \mu^{[g_2]})(\mu^{[g_1]} - \mu^{[g_2]})^T$$

We propose that, in each iteration of Algorithm 1, we select for merging two components  $g_1$  and  $g_2$ ,  $g_1 \neq g_2$ , such that  $B(g_1, g_2)$  is minimized. The dissimilarity measure  $B(g_1, g_2)$  as given in Equation (2.27) is reasonably easy to compute, with computational complexity at most  $\mathcal{O}(d^3)$ . Consequently, if our task is to reduce a mixture of  $K$  components to a mixture of  $K - 1$ , this will have total computational complexity of  $\mathcal{O}(K^3 d^3)$ . This criterion has qualitatively the right properties. Roughly speaking, it will tend to select for merging:

- 1) components with low weights. Note how the weights appear outside the logarithm in Equation (2.27), and so can have a dominant effect,
- 2) components whose means are close together in relation to their variances,
- 3) components whose covariance matrices are similar.

For further details see [95]. The fact that  $B(g_1, g_2)$  is merely an upper bound on the KL discrimination, rather than an exact value, is admittedly a drawback. Moreover, since KL discrimination does not satisfy the triangle inequality, there is no simple way of bounding the discrimination that arises over the course of two or more iterations of the algorithm.

However, obtaining a direct estimate of the KL bound would appear to require a numerical method, e.g. numerical integration. Worse, this integration would need to be carried out multiple times:  $\mathcal{O}K^3$  times if, as above, our task is to reduce  $K$  components to  $K - 1$  components. In many applications this will be computationally prohibitive. A possible compromise approach would be to use the  $B(g_1, g_2)$  criterion to compile a shortlist of possible component merges, selection from within this shortlist being by direct numerical integration. In this work, we chose a pairwise merging of components method that measures the dissimilarity between the post-merge mixture with respect to the pre-merge mixture based on an easily-computed upper bound of the *Kullback-Leibler* (KL) discrimination measure presented in [95].

Suppose we are given a mixture of two Gaussian components  $g_1, g_2$  with the parameters  $\theta$  and  $w$ , where  $\theta^{[i]} \equiv \{\mu^{[i]}, \Sigma^{[i]}\}$ ,  $i \in \{g_1, g_2\}$  and  $w^{[g_1]} + w^{[g_2]} = 1$ , and that we wish to approximate this mixture as a single Gaussian. A strong candidate is the Gaussian whose zeroth, first- and second-order moments match these two components, i.e., the Gaussian with mean vector  $\mu^{[g_{12}]}$  and covariance

matrix  $\Sigma^{[g_{12}]}$  as follows

$$\begin{aligned}\boldsymbol{\mu}^{[g_{12}]} &= w^{[g_1]}\boldsymbol{\mu}^{[g_1]} + w^{[g_2]}\boldsymbol{\mu}^{[g_2]} \\ \Sigma^{[g_{12}]} &= w^{[g_1]}\Sigma^{[g_1]} + w^{[g_2]}\Sigma^{[g_2]} + w^{[g_1]}w^{[g_2]}(\boldsymbol{\mu}^{[g_1]} - \boldsymbol{\mu}^{[g_2]})(\boldsymbol{\mu}^{[g_1]} - \boldsymbol{\mu}^{[g_2]})^T\end{aligned}\quad (2.28)$$

which is referred to as the moment-preserving merge of two Gaussian components  $g_1, g_2$ . Suppose that we are given a mixture with  $K$  components, and we wish to approximate it by a mixture of  $K - 1$  components, where  $K > 1$ . In this work, we choose the two components that in a sense to be defined are least dissimilar, based on Kullback-Leibler criterion, and replace them by their moment-preserving merge.

---

#### Algorithm 1 On-line Unsupervised Learning of GMM

---

**Input:** Sample data:  $Y_0, Y_1, \dots$ ,  
 Covariance matrix for postulated component  $\Sigma^{[0]}$ ,  
 Maximum number of hypotheses:  $\aleph_{max}$ ,  
 Dissimilarity measure threshold:  $B_{max}$   
**Output:** Number of the components:  $K$ ,  
 Mean and covariance of the components:  $\{\theta^{[1]}, \dots, \theta^{[K]}\}$ ,  
 Mixing weights:  $\{w^{[1]}, \dots, w^{[K]}\}$

**Initialization:** Postulate the first component

**Steps:**

**1. Update the current components in each hypothesis  $\mathcal{H}_1, \dots, \mathcal{H}_{\aleph}$ :**

Find the a posteriori probabilities  $\Gamma_{n+1}$  as Equation (2.7)

Update the current components by Equations (2.15-2.17)

Update the log-likelihood  $\ell$  as in Equation (2.30)

Update the description length  $\mathcal{L}$  as in Equation (2.26)

**2. Add a new hypothesis:  $\mathcal{H}_{\aleph+1}$**

Create a new component according to Equation (2.29)

Define the log-likelihood of this new hypothesis:  $\ell_{\aleph+1} = \ell_1$

Obtain the description length of this new hypothesis:  $\mathcal{L}_{\aleph+1}$  in Equation (2.26)

**3. Check if we can add another hypothesis:  $\mathcal{H}_{\aleph+2}$**

Find  $B$  for every pair of components in  $\mathcal{H}_1$  according to Equation (2.27)

**if**  $\min(B) < B_{max}$  **then**

Merge two components according to Equation (2.28)

Estimate the expected log-likelihood  $\ell_{\aleph+2}$  based on Equation (2.30)

Obtain the description length for this new hypothesis  $\mathcal{L}_{\aleph+2}$  in Equation (2.26)

**end if**

**4. Check if we can add another hypothesis:  $\mathcal{H}_{\aleph+3}$**

Component annihilation, Section 2.4.4.1

Estimate the expected log-likelihood  $\ell_{\aleph+3}$  based on Equation (2.30)

Obtain the description length for this new hypothesis  $\mathcal{L}_{\aleph+3}$  in Equation (2.26)

**5. Refresh the model**

Re-order incrementally the hypotheses with given number of components according to their description length  $\mathcal{L}$

Keep the first  $\aleph_{max}$  hypotheses

Acquire the next sample and go to 1

---

## 2.5 The Proposed On-line Unsupervised Learning Algorithm

In this section we describe the proposed on-line unsupervised learning algorithm, which is composed of several models as it will become clear later. Algorithm 1 describes the pseudo-code for one model and its rational is as follows:

- Start with one single observation and build the first hypothesis  $\mathcal{H}_1$  described by a single Gaussian distribution with mean  $\mu^{[0]}$  at the point itself, and some predefined covariance  $\Sigma^{[0]}$ . Then, calculate the log-likelihood of this hypothesis  $\ell_1 = -\log \sqrt{(2\pi)^d |\Sigma|}$  and find the corresponding description length  $\mathcal{L}_1$  according to Equation (2.26).
- The second acquired sample, updates the first hypothesis  $\mathcal{H}_1$  according to Equations (2.15-2.17), and builds the second hypothesis  $\mathcal{H}_2$  which contains two components: the first updated component and a second component with mean  $\mu^{[K+1]}$  at the point itself, with some predefined covariance  $\Sigma^{[K+1]}$

$$\mu^{[K+1]} = Y_{n+1} \quad w^{[K+1]} = \frac{1}{n+1} \quad (2.29)$$

where  $K$  is the number of components at the time (being  $K = 1$  for the case of the second sample). The selection of the covariance matrix for this new born component is optional and can be done with respect to the type of data that we are dealing with. We used a diagonal matrix where each element was a fraction of the standard deviation of the features of  $Y_{n+1}$ .

- The third point will update the two current hypotheses and build another one by adding a new component to  $\mathcal{H}_1$  and so on and so forth. For the sake of computational speed and memory, the number of hypotheses has to be bounded. Thus, after reaching the limit of maximum hypotheses  $\aleph_{max}$ , we rank the hypotheses in an increasing order according to their description length and keep only the first  $\aleph_{max}$  hypotheses and discard the rest.
- As explained above, in each iteration we add a new hypothesis by assuming that the new arriving point is a new component, according to Equation (2.29), beside the current Gaussian mixture in  $\mathcal{H}_1$ .

Thus, it is likely that we face the very common problem of over fitting, i.e., there is a tendency for the number of components of the mixture to grow without bound; indeed, if the algorithm were simply to follow the statistical model on which the method is based, the number of components would increase exponentially over time. To combat this, various pragmatic measures must be taken to keep the number of components in check. Typically this will be achieved either by discarding components with low probability, as suggested in [19], and/or by merging components which represent similar state hypotheses.

Thus, in each iteration after updating the current components in all hypotheses, we check the possibility of adding another hypothesis by merging two most similar components in  $\mathcal{H}$  (the hypothesis with minimum description length), according to the dissimilarity measure

$B_{max}$ . For example at time  $n$ , if there were 5 components in  $\mathcal{H}_1$ , by receiving a new point  $Y_{n+1}$ , first we would update the components in  $\mathcal{H}_1$  as we do in all other hypotheses; then if there were two similar components according to a threshold in  $\mathcal{H}_1$ , we would merge them and add another hypothesis  $\mathcal{H}_{K+2}$  (see Algorithm 1) composed by the post-merge mixture. For this new hypothesis the log-likelihood is set to be the same as the log-likelihood of the pre-merge mixture in  $\mathcal{H}_1$ , since it is assumed that the two components were very similar to each other.

- The dissimilarity measure threshold  $B_{max}$  is an important quantity since a very small value would not be helpful in tackling the over-fitting problem and setting a very high threshold can cause under-fitting of the components. To address this problem, we separate the hypotheses with given complexity at each time, then resort incrementally the hypotheses in each , set of hypotheses with given  $K$ .

Basically, in each hypothesis we have the mean, covariance and weights of the mixture beside the number of points in each Gaussian and the description length of the model.

Another point that needs to be taken into consideration is the fact that the computation of the log-likelihood has to be done in a recursive on-line format. Thus, after updating  $\theta_n$  and  $w_n$ , For some practical reasons, in Equations (2.15-2.17), we changed the *learning rate*  $\frac{1}{n}$  to a faster decaying envelope, i.e. we added a sufficiently large enough constant to  $n$  in order to reduce the problem of instability as proposed in [36].

One of the greatest challenges of unsupervised learning of GMMs is the dynamic *model order* selection. In the last stage of our proposed method, all the current hypotheses are being sorted incrementally based on their description length value and the first,  $\mathcal{N}_{max}$  hypotheses are being stored and the rest discarded.

As mentioned before, in the third and forth stages, we may add a new hypothesis by merging two candidate components or annihilating some *weak* components. In order to calculate the description length of these hypotheses, we need to have the log-likelihood of them (Equation (2.4)). The problem is that for the computation of  $p(Y|\theta)$  historical data  $Y$  is needed, which is not available. As proposed in [42], instead of  $p(Y|\theta)$ , we use the expected likelihood of the same number of data points and, hence, the expected description length for the new born hypotheses is then computed. More precisely, consider two Gaussian components  $g_1(Y) \sim \mathcal{N}(Y; \theta^{[g_1]})$ ,  $g_2(Y) \sim \mathcal{N}(Y; \theta^{[g_2]})$  with the parameters  $\theta$  and  $w$ , where  $\theta^{[i]} \equiv \{\mu^{[i]}, \Sigma^{[i]}\}$ ,  $i \in \{g_1, g_2\}$  and  $w^{[g_1]} + w^{[g_2]} = 1$ .

Let  $g_{12}$  be the moment preserving merge of these two components can be obtained by Equations 2.28.

The expected likelihood of  $N_1$  points drawn from the former and  $N_2$  points from the latter given model  $w^{[g_1]}g_1(Y) + w^{[g_2]}g_2(Y)$  is

$$\mathbb{E}[p(Y|\theta^{[g_{12}]})] = \left( \int g_1(Y)(w^{[g_1]}g_1(Y) + w^{[g_2]}g_2(Y))dY \right)^{N_1} \left( \int g_2(Y)(w^{[g_1]}g_1(Y) + w^{[g_2]}g_2(Y))dY \right)^{N_2} \quad (2.30)$$

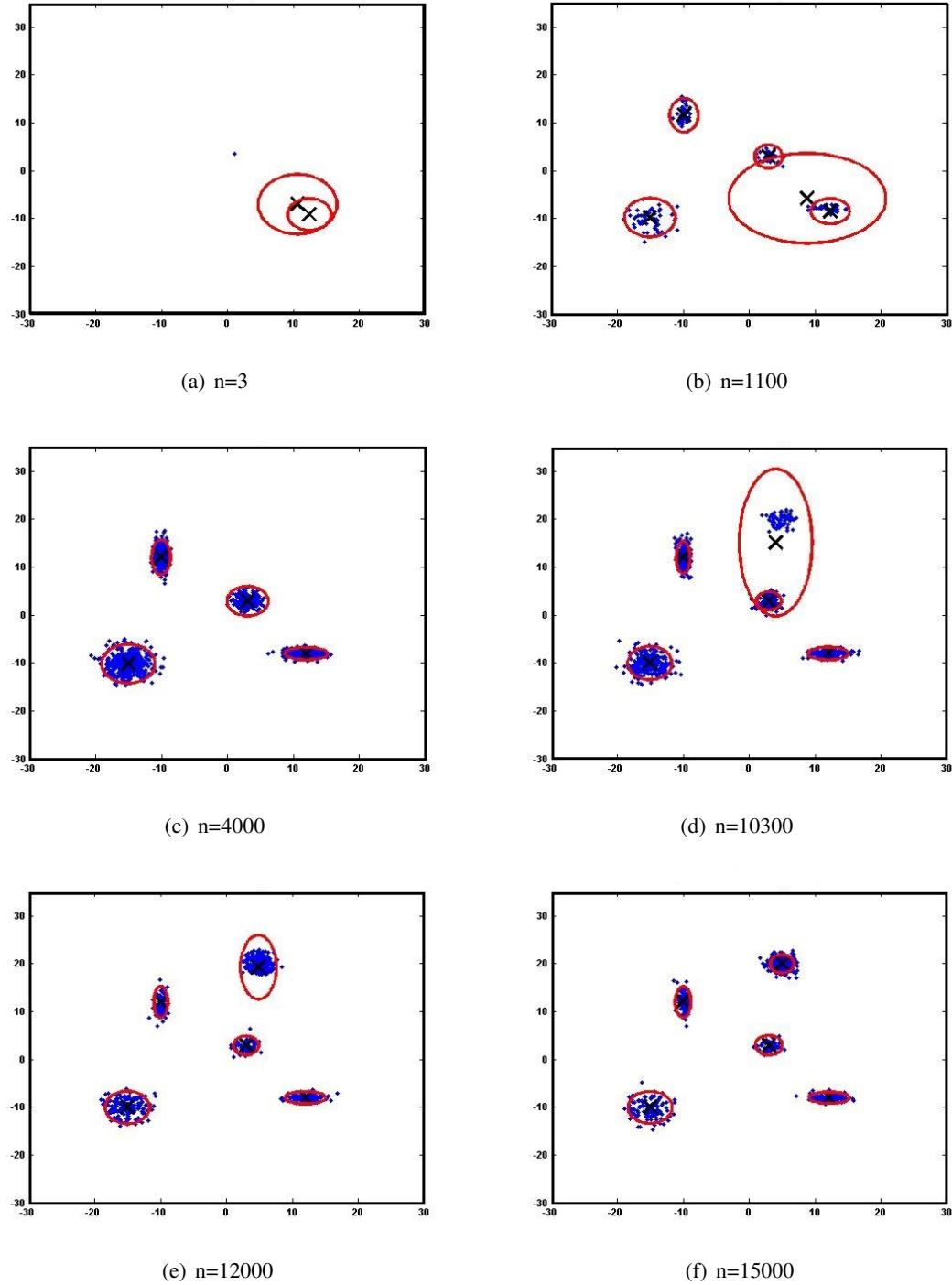


Figure 2.3: An example of the execution behaviour of the proposed algorithm.

where integrals of type  $\int p^i(y)p^j(y)dy$  are recognized as the Bhattacharyya distance, which for Gaussian distributions can easily be computed as

$$d_B(p^i, p^j) = \int p^i(y)p^j(y)dy = \frac{\exp(-C/2)}{(2\pi)^{(d/2)} |\Sigma^i \Sigma^j \Sigma^{ij}|^{1/2}} \quad (2.31)$$

where

$$\begin{aligned}\Sigma^{ij} &= (\Sigma^{i-1} + \Sigma^{j-1})^{-1}, \quad \mu^{ij} = \Sigma^{ij}(\Sigma^{i-1}\mu^i + \Sigma^{j-1}\mu^j) \\ C &= \mu^i\Sigma^{i-1}\mu^{iT} + \mu^j\Sigma^{j-1}\mu^{jT} - \mu^{ij}\Sigma^{ij-1}\mu^{ijT}\end{aligned}$$

After calculating the expected log-likelihood by Equation (2.30), the description length can be obtained by Equation (2.26).

## 2.6 Simulation Results

This section illustrates the behaviour of the proposed algorithm for two types of experiments: a synthetic Gaussian mixture data set and the Iris data set.

### 2.6.1 A 2-d Gaussian Mixture

Figure 2.3 shows an example of 3 models running in parallel in order to find a mixture of well separated synthetic Gaussian components in real time starting with one single observation. The maximum number of hypotheses was set to  $\mathcal{N}_{max} = 10$  and the merging threshold  $B_{max}$  to 0.008, 0.08, and 0.8, respectively. This experiment can be split in two steps. For  $n < 10000$ , the observed data were randomly extracted, according to Equation (2.1) with 4 components ( $K=4$ ) and the mixing weights of the components from left to right are  $w = [0.35, 0.25, 0.15, 0.25]$ . It can be seen, after some transient situation, that the algorithm merged the two most similar components and were able to correctly determine the 4 components. Then, for  $n \geq 10000$ , we started to extract data from another component beside those previous ones. The algorithm was able to converge to the solution rapidly.

### 2.6.2 The Iris Data Set

We used the well-known 3-component 4-dimensional “Iris” data set [96]. This data set has only 150 samples, and therefore we had to randomize and repeat them 60 times. We set the maximum number of hypotheses  $\mathcal{N}_{max} = 50$  in 10 different models with the merging threshold starting from  $B_{max} = 0.002$ . Figure 2.4(a) shows that in 64 out of 100 trials the 3 components were correctly identified. By visual inspection we could observe that the linearly separated component (iris setosa) could almost perfectly be identified. On the other hand, the proper identification of the other two non-linear separable components (iris versicolor and iris virginica) was more challenging since the order in which the data is presented can influence the recursive solution. The typical solution is shown in Figure 2.4(b) by projecting the 4-dimensional data to the first two principal components.

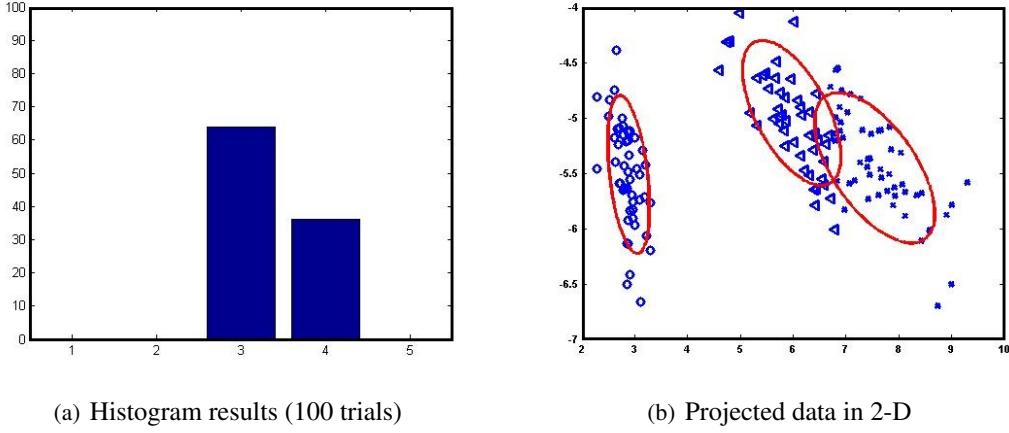


Figure 2.4: Iris Data Set Results

### 2.6.3 A 1-d Gaussian Mixture

In this experiment, the data is a mixture of three linearly separable Gaussian components, as is shown in Figure 2.5(a). The weights of the components are equal. The means of the components are  $[-40, 0, 40]$  and the variance values are  $[15, 10, 20]$  respectively. Figure 2.5(b) illustrates the output of our algorithm at time  $n = 5000$  for a hundred trials.

Next, we did the same experiment with data shown in Figure 2.6(a) that represents a mixture where the components are closer together, i.e., they are not linearly separable any more. The means are  $[-20, 2, 25]$  and variances are  $[4, 60, 20]$ . Figure 2.6(b) shows the results of 100 trials that the algorithm succeeded to find the components correctly. A more complex case is represented in Figure 2.7(a), where the components are highly overlapped. The weights of the components are the same. The mean and variance values are  $[-10, 0, 10, 20]$  and  $[6, 2, 8, 4]$  respectively. Figure 2.7(b) illustrates the output of 100 trials. There is no surprise to witness *over fitting* in some cases, due to the nature of the mixture.

## 2.7 Conclusion

This chapter proposed an on-line unsupervised learning of GMMs algorithm in the presence of uncertain dynamic environments. The algorithm relies on a multi-hypothesis adaptive scheme that continuously updates the number of components and estimates the model parameters as the measurements (sample data) are being acquired. The hypothesis models are ranked according to the MDL. In this work we propose an unsupervised learning of Gaussian mixture models algorithm, that is flexible in terms of shape of the components, can deal with high dimensional data, is robust toward outliers and noises and with minimum dependency on initialization or prior knowledge. In general, we could conclude that the algorithm has a good performance specially when the components are well separated. However, it is worth to mention that a critical issue is the initial selection of the covariance when a new component is created. This has to be done carefully

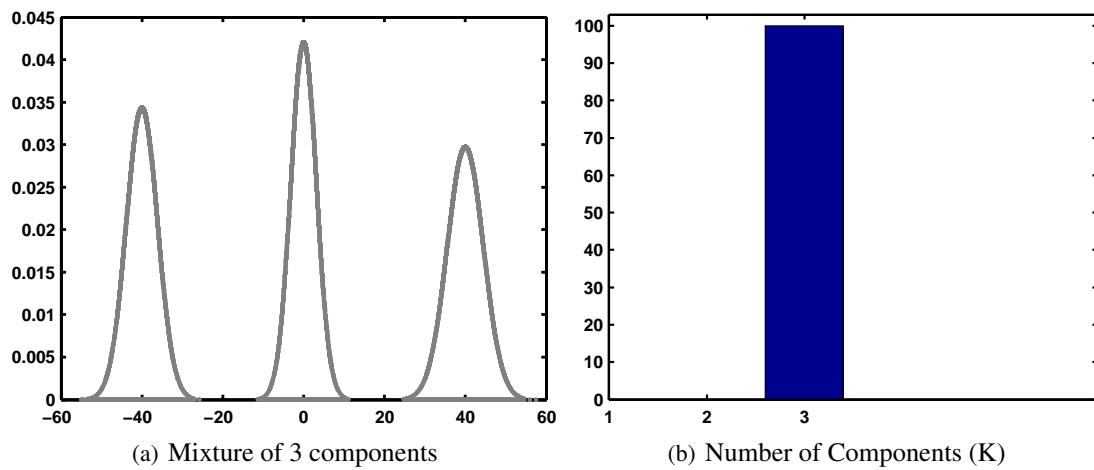


Figure 2.5: Linearly separable mixture

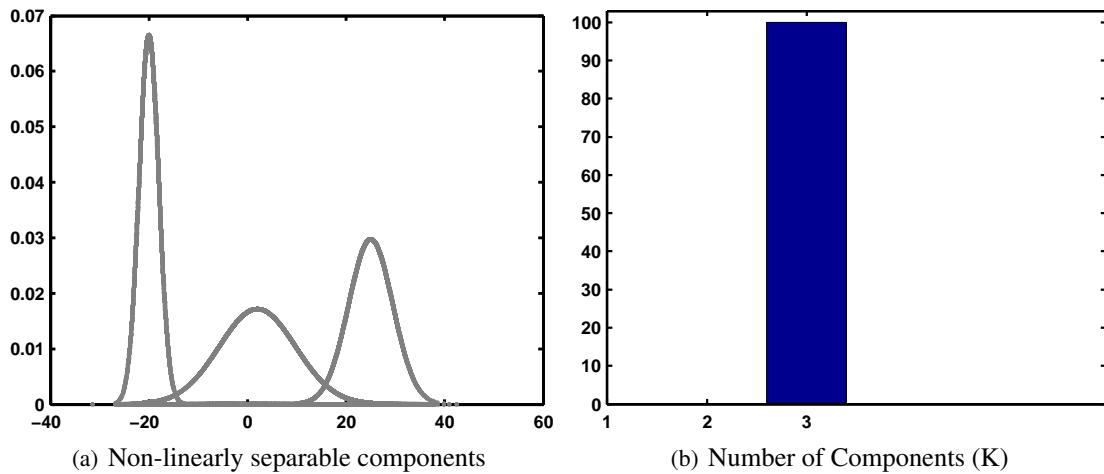


Figure 2.6: Fairly overlapped mixture

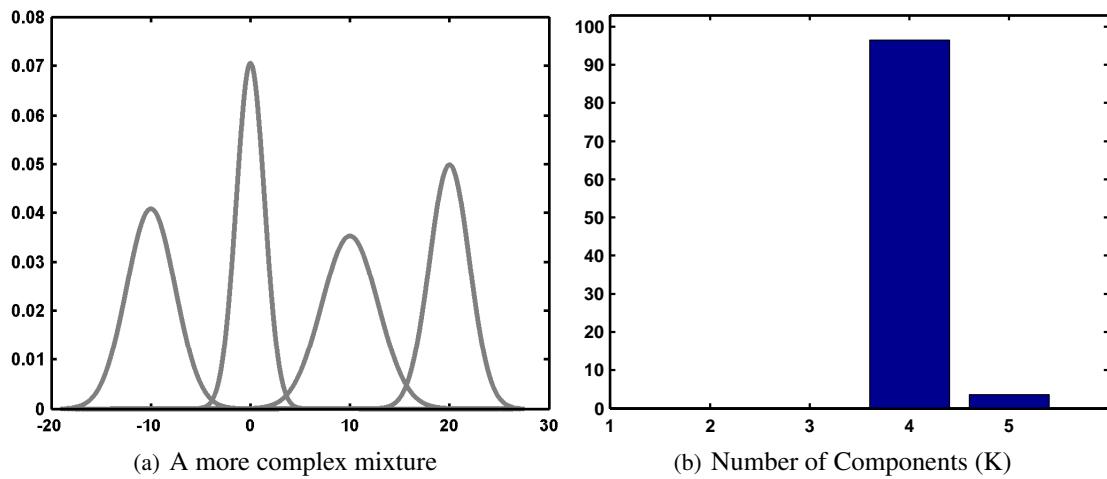


Figure 2.7: Highly overlapped mixture

because choosing a very small covariance can be experimentally problematic since in the process of calculating the a posteriori probability in Equation (2.7), the result in Equation (2.3) could be zero due to finite precision. On the other hand, choosing an extremely large covariance can lead to the “under-fitting” problem. This is something that deserves further investigation.



## Chapter 3

# Real-Time Unsupervised Motion Learning for Autonomous Underwater Vehicles

### 3.1 Introduction

An Autonomous Underwater Vehicle (AUV) is a robot designed to operate underwater. It is typically a free swimming body and is not attached to the support vessel to which it is launched. Autonomous Underwater Vehicles (AUVs), by taking advantage of sophisticated, smart, small and inexpensive on-board sensors and marine robotic platforms, have brought together specific complementary knowledge in computer science, electrical and mechanical engineering.

Nowadays, we are able to carry out ocean expeditions thoroughly without actually being there. These vehicles, are a vital tool in gathering detailed ocean data with much higher resolution and at reasonable cost in order to target specific set of oceanography questions.

Further, the idea of using inexpensive multiple AUVs, acting in cooperation to perform specific tasks, like for example ocean sampling, mapping, mine detection and neutralization, offers potentially significant advantages in performance and efficiency as well as redundancy in case of failure, compared with the use of a single AUV, see [97] and [98]. Scientific applications include geoscience to map particular features such as hydrothermal events or submarine volcanoes [3]. In archeology for documenting shipwrecks and submerged cities [4]. In oceanography for mapping the physical structure of the ocean. In ecological applications for surveying the marine habitants and document their states to understand the changes through time. In industry they are used extensively used for conducting surveys for minerals and in oil and gas exploration. AUVs are also used in defence applications to fulfil dangerous roles such as mine counter measures and rapid environmental assessments.

The rest of this chapter is organized as follows. Section 3.2 describes the state of the art of adaptive sampling and motion learning for AUVs. Section 3.3 explains our proposed real-time motion learning strategy for multiple AUVs . The applicability of the proposed algorithm presented in

Chapter 2, is investigated in Section 3.6. Two different scenarios are defined: Real-time unsupervised motion learning for AUVs in an uniform environment, and a more complex scenario, where the environment is non-uniform. This chapter ends with some conclusions.

## 3.2 Literature Review on Adaptive Sampling using AUVs

In oceanographic sampling the goal is to produce a map of a given environmental quantity (e.g., temperature, or sound speed, or bottom morphology, depending on the specific payload) accurately and in the minimum amount of time. When a hundred percent coverage is not strictly required, or it is not possible because the payload can only make a point-wise measurement (as in the temperature/salinity case), the produced map is an estimate of the true map based on the available samples. To meet the accuracy specification while minimizing the number of sampled points, one intuitive idea is that of increasing the spatial sampling rate where the environmental map is rapidly changing, while decreasing it when the environmental map is almost constant. As sample progress, the smoothness (or the spatial correlation length) of the map can be estimated from the data themselves; the next sampling location can then be established on-line on the basis of the previous measurements. This approach is called “adaptive sampling”. If the sampling is performed by a team of AUVs, then the sampling strategy must be chosen in order to exploit the availability of multiple vehicles through some coordination strategy, possibly implemented in a distributed fashion.

To this end, the following questions need to be addressed: How to program the individual robots so that they have their own on-board fast decision making ability? How to collect data in most efficient way? How to increase the robustness of the group to uncertainty or disturbances in the environment? How to conveniently use the network of sensors available to robustly collect data to reveal desired features of the environment? How to stabilize the formation of the vehicles so that they could change the resolution of the formation, shape of the formation of individual agents moving around?

In the literature, it is possible to find several interesting works that have proposed adaptive sampling strategies for marine systems. Inspired by natural systems, for example fish schools that are able to climb the gradients even through noisy fields, Leonard et al. in [99] presents a framework, relatively simple at the individual level but with greater functionality and intelligence at the group level, for coordinated and distributed control of multiple autonomous vehicles using artificial potentials and *virtual leaders*. Each of these potentials is a function of the relative distance between a pair of neighbours. Artificial potentials define interaction control forces between neighbouring vehicles and are designed to manipulate the formation and geometry of the fleet. A virtual leader is a moving reference point, at the center of the formation, that attracts or repels the neighbouring vehicles by means of additional artificial potentials. In the reported work, the closed-loop stability using the system kinetic energy and the artificial potential energy is guaranteed by constructing a Lyapunov function. By considering no ordering of the vehicles in fleet, the adapted setup is more robust in comparison with a single vehicle usage. In [100], following the approach proposed

in [99], a robotic ocean sampling network is developed where AUVs move around on their own, collecting data on the ocean physics and biology to gain a better understanding of ecosystem and ocean climate change. This idea in simple words can be described as follow. Having a group of sensors, maybe in a formation measuring a scalar like temperature, by themselves each individual can not know about the gradient but collectively, if they act like fish, they can pass the information and somehow get an estimation of the gradient and climb that gradient. The algorithm used in that work is based on artificial potential. Three vehicles at the corners of a triangle, with this ability that the center of the triangle can track a line. The ability to control the formation to change the resolution or one edge of the triangle be normal to the path. They have built some other related things based on this idea of dynamic gradient climbing by extending the approach to estimating the second derivative and understand the curvature of the level sets, and also by changing the resolution of the fleet so that minimize the estimation error. The agents not only do sampling while tracking, but they can reconfigure as they go. This is a significant advantage of using mobile sensor networks. Ogren et al. in [101], present a stable control strategy for a group of vehicles to direct and reconfigure cooperatively based on changes in acquired measurements. In the reported work, the vehicles follow the thermal gradient direction by sending the sampled data, during each resurfacing, to a central unit to update the mission plan and send it for each vehicle. In [102], the authors propose a cooperation algorithm for adaptive oceanographic sampling, taking into account range communication constraints. In a fleet of AUVs with static topology, a distributed dynamic programming algorithm is applied to solve the global optimization problem of maximizing the oceanographic sampling area coverage. Worth to be mentioned that the solution (added to a cost function) is computed in a distributed fashion, since each vehicle makes the cost computation for its own candidate points. In the reported work, for practical issues and to preserve underwater communication possibilities, the maximum distance between adjacent agents is limited by an upper bound. Every vehicle submerges vertically to capture the measurements and then resurface while carrying a payload of Conductivity, Temperature and Depth (CTD) data. The data is transmitted to a measuring station on the surface of the sea. When on the surface, the vehicle navigates with the help of GPS. A land-station link enables almost real-time data transmission and on-line modification of the mission plan. In [103], a map is used to decrease the uncertainty of the sampling by considering the correlations among the ocean values. In [104] an  $A^*$  approach to trajectory planning and finding feasible paths through a detailed map of known obstacles and unsafe regions is presented. Significant computation times for  $A^*$  path planning (order 10-100s CPU time with currents sampled only once every nautical mile) is stressed by the authors. Smith et al. in [105], present a near-real time path planning solution for tracking and sampling an evolving ocean feature using one or more glider(s). In the reported work, a Regional Ocean Model System (ROMS), which is known a priori, generates a sampling plan that steers deployed gliders to regions of scientific interest based upon the given feature. Throughout the execution of the sampling plan, the collected data are transmitted and assimilated into the ocean model. We believe that the performance of the proposed approach strongly depends on ROMS as it predicts the behaviour of the objective feature and steer the vehicles toward new trajectories. Since the existence of outliers

in sampling, specially in highly unstructured underwater environment, can affect the 2D gradient direction dramatically, outlier detection in sampling is an open study. In [106], the authors propose an adaptive control of autonomous mobile sensor platforms approach for oceanographic sampling purposes. To investigate the viability of their approach, an experiment with the goal of adaptively following the ocean thermal gradient was conducted. Considering the fact that the optimum sampling path was not predefined, the sensor platform should have adaptively manoeuvre itself based on the real-time measurements. The experiment consisted in 4 segments, in each, the CTD autonomy sensor could go down and rise in a certain direction and in a predefined zigzag path to measure the temperature profile of the ocean. The authors in this work, developed an autonomy architecture to support adaptive sampling. With the aim of reducing the model uncertainty, and considering the fact that the optimum sampling path is not predefined, the sensor platform must adaptively manoeuvre itself based on the real-time measurements. The sampling process, records the ocean temperature at each proper depth. The direction of any segment, except the first segment that was predefined, was the 2D thermal gradient of the batch of data captured during the previous segment. The authors used autonomous surface craft as a mobile sampling platform, which in comparison with AUV, has the advantage of better navigation and communication. As the authors stated, higher sampling density could be achieved in shorter time and with less power consumption, if AUV had been used instead of an autonomous surface craft. It seems that, this method is less resilient to outlier and noise in sampling which is very likely in unstructured underwater environment. Note that the existence of outlier can change the direction of the 2D gradient dramatically. By spreading the use of autonomous vehicles, congestion and highly dynamic traffic on the surface is becoming more challenging. Svec et al. in [107] tackles this problem by developing a model-predictive, local path planning algorithm for an unmanned surface vehicle (USV). The goal is to map the spatio-temporal obstacle regions, with increasing resolution and focused sampling, by introducing the *velocity obstacle* concept to the systems with nonlinear dynamics, nonholonomic constraints, and any form of low level feedback control. In the reported work, to grant the International Regulations for Prevention of Collisions at Sea (COLREGs), the sampling of motion goals is constrained.

Our proposed approach consist in one leader AUV and two or more follower AUVs, all equipped with conductivity, temperature and depth (CTD) sensor devices. The CTD data is modelled as a Gaussian mixture model (GMM) by running the proposed algorithm in Chapter 2 in each vehicle separately. In the setup adopted, a leader AUV is tasked to acquire CTD data by running a set of user-defined mission instructions like for example following a desired path profile. The aim of each follower AUVs is to follow the leader closely with a desired formation that will adaptively change according to the CTD data that they are acquiring. More precisely, each AUV is in charge of running in real-time the unsupervised learning algorithm for GMMs that is fed by the CTD data. To make the scheme robust to fault of underwater communications, which is very prone to happen, in this approach we assume that the followers only on the surface can receive the GMM hypothesis of the leader. In other words, each time the vehicles resurface (and this is done in a coordinated fashion), the leader AUV broadcast its currently estimated parameters of the GMM,

and the followers based on this and their own estimated GMM compute the variational distance error between these GMMs. This error is the *variational distance* between two GMMs that gives a notion of how different is the CTD environment of each follower with respect to the leader. Thus, during each resurfacing, the calculated error and the geographical position of the leader is used to guide the next formation configuration, which typically scales the distance between the AUVs in the formation (making a zoom-in and zoom-out), in order to improve the efficiency of data acquisition in the given region. Therefore, every follower has to explore the environments where, in terms of CTD data, are desirably different from the leader.

Comparing with the related literature described above, in our approach we have the following advantages: The vehicles do not need to have communication underwater. No prior knowledge of the dynamics of the environment is required in unsupervised motion learning scenario. Moreover, no significant payload to save data is also needed since each vehicle only saves a GMM profile of the captured CTD data. Furthermore, our proposed GMM based method is much more resilient toward outlier and noise by nature, because outliers do not carry enough *evidence* to manipulate the GMM significantly. Also, it is more efficient in terms of resources since followers can cover a larger area and expand the formation with respect to the position and GMM of sampling profile of the leader.

### 3.3 Problem Formulation

We address the problem of real-time adaptive sampling using a coordinated fleet of AUVs. To this end, we provide a motion control algorithm that includes the unsupervised learning of Gaussian Mixture Models (GMMs) described in the previous chapter. The system set up consist of one leader AUV and two or more followers AUVs. All vehicles are equipped with CTD sensor devices. The leader moves around, in a predefined path, while acquiring CTD data. Meanwhile, it constructs a GMM profile by running the unsupervised learning of GMM algorithm described in Chapter 2, see also in [64]. The objective is to execute a coordinated formation maneuver, by driving and maintaining the other vehicles, henceforth known as followers, at a desired position with respect to the leader that is a function of the dissimilarity of the CTD profiles of the leader and the followers. This dissimilarity measure is the *variational distance* between two GMM profiles. To make the scheme robust to fault of underwater communications, which is very prone to happen, in this approach we consider that follower only on the surface can learn the GMM hypothesis of the leader to calculate the error (dissimilarity measure). while on the surface, the vehicles can also use GPS to correct their navigation errors.

### 3.4 Preliminaries and Background

AUVs are equipped with a variety of sensor systems and communication devices. There are some concepts and devices found in nearly every AUV and others, which are only used for specific

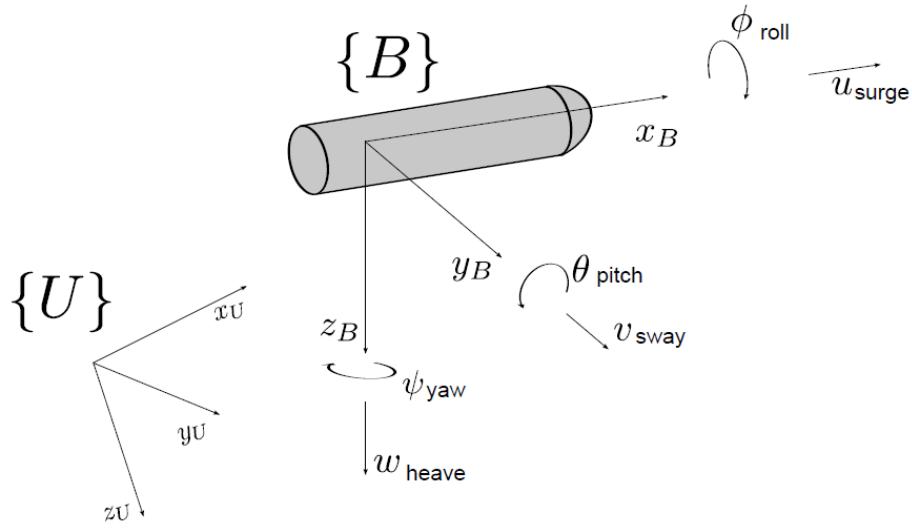


Figure 3.1: The earth-fixed inertial frame  $\{U\}$  and the body-fixed frame  $\{B\}$  with position, orientation and linear and angular velocities

mission tasks. The following section gives an overview of sensor and communication systems that are used in AUVs.

### 3.4.1 Coordinate Frames

The equations of motion for an AUV require the definition of an earth-fixed inertial frame  $\{U\}$  with the orthogonal axes  $\{x_U, y_U, z_U\}$  and a body-fixed frame  $\{B\}$  with the orthogonal axes  $\{x_B, y_B, z_B\}$ . The position and orientation of the vehicle is usually given with respect to the inertial frame  $\{U\}$ . The linear and angular velocities are expressed in the body-fixed frame. The body-fixed frame usually coincides with the center of gravity of the vehicle. The body axes are defined as follows [108]:

- $x_B$  is the axis from aft to fore
- $y_B$  is the axis from port to starboard
- $z_B$  is the axis from top to the bottom

Figure 3.1 shows the two frames. For the sake of more clarity, we define the following entities:

- $\eta_1 = [x, y, z]^T$ , the origin of  $\{B\}$  with respect to  $\{U\}$
- $\eta_2 = [\phi, \theta, \psi]^T$ , the angles of orientation of  $\{B\}$  with respect to  $\{U\}$ .  $\phi$  is called roll,  $\theta$  pitch and  $\psi$  yaw.
- $v_1 = [u, v, w]^T$ , the linear velocities of the origin of  $\{B\}$  relative to  $\{U\}$  expressed in  $\{B\}$ .  $u$  is called surge,  $v$  is called sway and  $w$  is called heave.

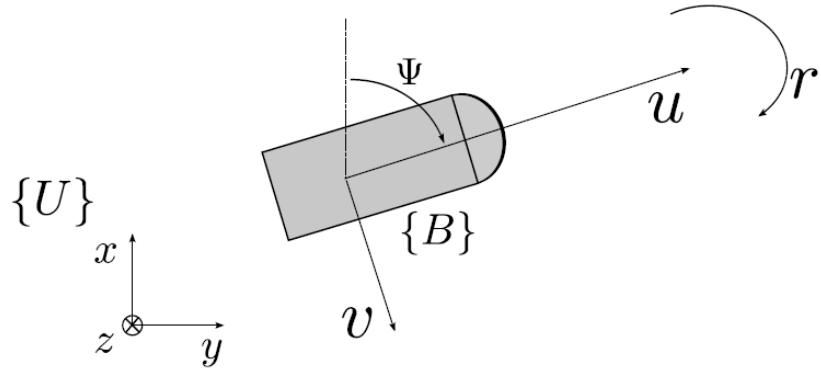


Figure 3.2: Simplified kinematic model of an underwater vehicle that maintains in a horizontal plane.

- $v_2 = [p, q, r]^T$ , the angular velocities of the origin of  $\{B\}$  relative to  $\{U\}$  expressed in  $\{B\}$ .
- $\tau_1 = [X, Y, Z]^T$ , the actuating forces expressed in  $\{B\}$
- $\tau_2 = [K, M, N]^T$ , the actuating torques expressed in  $\{B\}$

The following vectors are introduced for convenience:  $\eta = [\eta_1^T, \eta_2^T]^T$ ,  $v = [v_1^T, v_2^T]^T$ ,  $\tau = [\tau_1^T, \tau_2^T]^T$ .

### 3.4.2 Simplified Kinematic Equations

In this thesis, we consider the control of the vehicles in horizontal plane. The kinematic equations relate the position and orientation vector  $\eta$ , which is expressed in the inertial frame  $\{U\}$ , to the velocity vector  $v$ , which is expressed in the body-fixed frame  $\{B\}$ . The dynamic equations, which are not considered in here, relate the velocity vector  $v$ , which is expressed in the body-fixed frame  $\{B\}$ , to the force and torque vector  $\tau$ , which is also expressed in the body-fixed frame  $\{B\}$ . Figure 3.2 shows a simplified version of the kinematic model of an underwater vehicle. Interested readers are encouraged to see [109, 108]. This assumption leads to simplified kinematic equations

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} u \cos \psi - v \sin \psi \\ u \sin \psi - v \cos \psi \\ r \end{bmatrix} \quad (3.1)$$

with  $p = [x, y, \psi]^T$  denoting the position and orientation of the vehicle on the two-dimensional plane and  $v = [u, v, r]^T$  denoting the velocity. Equation (3.4) can be written in the form

$$\dot{p} = R(\psi)v, \quad (3.2)$$



Figure 3.3: A close up of the CTD sensor of the AUV (image source: [1])

with the rotational matrix  $R(\psi)$

$$R(\psi) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

### 3.4.3 CTD Sensors

A CTD, an acronym for Conductivity, Temperature, and Depth, is a package of electronic instruments and a primary tool for determining essential physical properties of sea water, see Figure 3.3. The depth of the vehicle from the surface is determined by means of a pressure sensor. The primary function of a CTD device is to detect how the conductivity and temperature of the water column changes relative to depth. Figure 3.4 represents the 2-D and 3-D temperature profile in a trial. Conductivity is a measure of how well a fluid conducts electricity. Figure Conductivity is directly related to salinity, which is the concentration of salt and other inorganic compounds in seawater. Salinity is one of the most basic measurements used by ocean scientists. When combined with temperature data, salinity measurements can be used to determine seawater density which is a primary driving force for major ocean currents.

Ocean explorers often use CTD measurements to detect evidence of volcanoes, hydrothermal vents, and other deep-sea features that cause changes to the physical and chemical properties of seawater. CTDs can provide profiles of chemical and physical parameters through the entire water column. By analysing these parameters, scientists can make inferences about the occurrence of certain biological processes, such as the growth of algae. Knowledge obtained from CTD devices can, in turn, lead scientists to a better understanding of such factors as species distribution and abundance in particular areas of the ocean. Sudden changes or *anomalies*, in one or more of the properties being measured may alert scientists to an unusual occurrence, such as an active hydrothermal vent [1]. The main advantages of CTDs are accuracy and light weight. On the flip-side the calibration of the devices especially for long term missions can be challenging. Until recently, once an AUV was launched it was completely isolated from its human operators until it returned

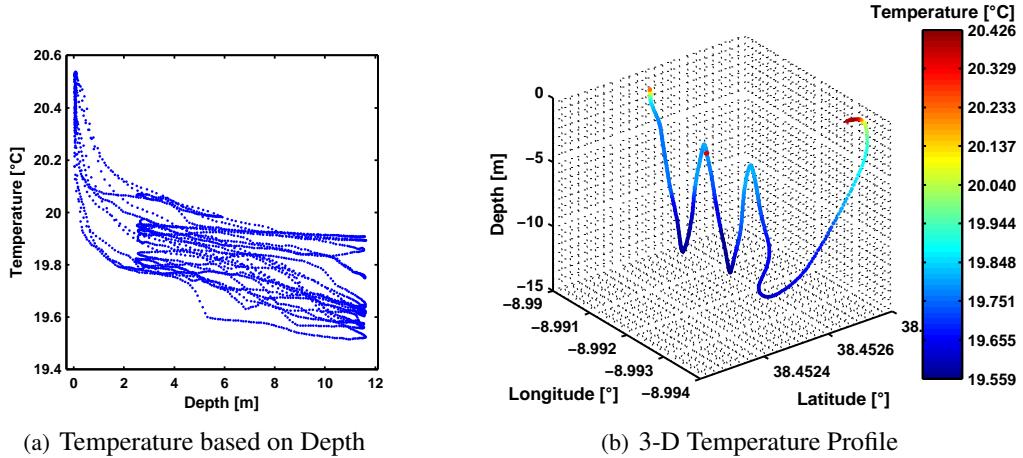


Figure 3.4: Temperature Profiles (images source: [17]).

from its mission. Because there was no effective means for communicating with a submerged AUV, everything depended upon instructions programmed into the AUV’s on-board computer. Today, it is possible for AUV operators to send instructions and receive data with acoustic communication systems that use sound waves with frequencies ranging roughly between 50 Hz and 50 kHz [7]. These systems allow greater interaction between AUVs and their operators, but basic functions are still controlled by the computer and software on-board the AUV.

Basic systems found on most AUVs include: propulsion, usually propellers or thrusters (water jets); power sources such as batteries or fuel cells; environmental sensors such as video and devices for measuring water chemistry; computer to control the robot’s movement and data gathering functions; and a navigation system.

Navigation has been one of the biggest challenges for AUV engineers. Today, everyone from backpackers to ocean freighters use global positioning systems (GPS) to find their location on Earth’s surface. But GPS signals do not penetrate into the ocean. One way to overcome this problem is to estimate an AUV’s position from its compass course, speed through the water, and depth. This method of navigation is called “dead reckoning”, and was used for centuries before GPS was available. Dead reckoning positions are only estimates however, and are subject to a variety of errors that can become serious over long distances and extended time periods.

In a confined area, the position of an AUV can be determined using acoustic transmitters that are set around the perimeter of the operating area, see Section 3.4.4.3. These transmitters may be moored to the seafloor, or installed in buoys. Some buoy systems also include GPS receivers, so the buoys’ positions are constantly updated. Signals from at least three appropriately positioned transmitters can be used to accurately calculate the AUV’s position. Although this approach can be very accurate, AUV operators must install the transmitters, and the AUV must remain within a rather small area [109, 110].

A more sophisticated approach uses Inertial Navigation Systems (INS) that measure the AUV’s

acceleration and angular velocity in all directions. These systems provide highly accurate position estimates, but require periodic position data from another source for greatest accuracy. On surface vessels and aircraft equipped with INS, additional position data are often obtained from GPS. On underwater vessels, the accuracy of INS position estimates is greatly improved by using a Doppler Velocity Logger (DVL) to measure velocity of the speed of the vessel. On some AUVs, several of these systems are combined to improve the overall accuracy of on-board navigation.

### **3.4.4 Navigation**

With recent advances in battery capacity and the development of hydrogen fuel cells, autonomous underwater vehicles (AUVs) are being used to undertake longer missions that were previously performed by manned or tethered vehicles. Navigation is one of the most critical factors in determining the operational suitability of any unmanned vehicle for its designated environment. In fully autonomous vehicles, due to the lack of a human operator to perform the navigation task, there is a fundamental requirement to incorporate estimation techniques that can provide the desired information necessary for navigation. Such information includes position, attitude, and velocity of the vehicle. Unlike unmanned underwater vehicles (UUVs) which are usually operated remotely by an acoustic modem link, AUVs present a uniquely challenging navigational problem because they operate autonomously in a highly unstructured environment where satellite-based navigation is not directly available, see e.g. [111]. As a result, more advanced navigation systems are needed to maintain an accurate position over a larger operational area.

Generally speaking, all underwater navigation systems problems fall into the 2D position localization (i.e. longitude and latitude). It is due to the fact that all submersible vehicles are outfitted with a pressure sensor and are capable of determining their absolute depth with high accuracy and a high update rate.

The accuracy of a navigation system is critical for the oceanographers to more fully exploit quantitative data from high resolution sensors such as high-frequency bathymetric sonar sensors and optical cameras. Furthermore, the closed-loop feedback control of underwater robotic vehicles and improvement of the quality of collected data during survey missions are eminently dependent on the accuracy of the navigation system.

In this work, the need for precise navigation is more evident because the trajectories of the followers are a function of the GMM hypothesis of the leader which in turn, depends on how accurate the leader moves in a predefined path. If the AUV does not follow the path accurately during the mission, critical features may not be recorded and the position of any features recorded during the mission will be uncertain. In the literature, many different methods for navigation in different under-water environments have been proposed, see e.g., [110, 111].

Most AUV navigation breaks down into five major categories: GPS, inertial, visual, dead-reckoning and acoustic.

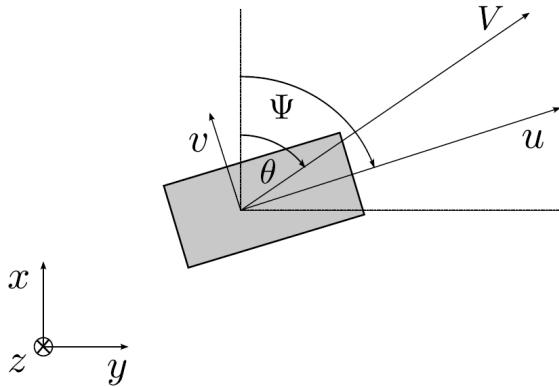


Figure 3.5: A two-dimensional kinematic model of an AUV. The angle  $\theta$  of the resulting velocity  $V$  does not necessarily correspond with the heading angle  $\psi$

#### 3.4.4.1 Global Positioning System (GPS)

An easily accessible global frame is the International Terrestrial Reference Frame 2000 (ITRF2000) available by GPS positioning [112]. The GPS can be used for navigation with bounded-error when the vehicle is operating on the surface. The GPS system's radio-frequency signals are blocked by sea water, thus GPS signals cannot be directly received by deeply submerged ocean vehicles. However, GPS commonly aides a variety of underwater vehicle navigation techniques, including surveying of acoustic transponders, alignment calibration of Doppler sonar systems [113].

#### 3.4.4.2 Magnetic Compass

Digital compass plays a vital role in autonomous vehicles. It provides the 3D-vector of the local magnetic field. It can not only determine bearing, but when used as an element for dead reckoning, can determine location. Placement in a vehicle, however, exposes compasses to a variety of hard to quantify influences. Digital compasses are subject to hard and soft iron errors, acceleration errors, and severe inclinations, which can affect a heading calculation and increase error.

Therefore, before computing the heading angle  $\psi$  of the vehicle from the magnetic field vector it is necessary to carefully calibrate the compass each time we have a mission in a new environment, as the “variation”, difference between the orientation of the 3D magnetic field vector and the direction of true north, changes in new geographic location.

In the field, if something goes wrong, it can be difficult to determine what has happened; by testing compass accuracy in a lab one can better understand the limitations of a given device. The two-dimensional heading angle  $\psi$  of the vehicle can be measured by a magnetic compass. Such a compass is found in almost every underwater vehicle. Furthermore, the compass suffers from local magnetic anomalies and magnetic fields induced by the vehicle itself [109]. All mentioned aspects lead to noisy measurements, which can hardly assumed to be Gaussian. In contrary to most land vehicles, for marine vehicles the heading angle  $\psi$  does not necessarily correspond with the angle  $\theta$  of the resulting velocity  $V$  of the vehicle (see Figure 3.5).



Figure 3.6: A typical DVL device (image source: [1])

#### **3.4.4.3 Doppler velocity log (DVL)**

The Doppler velocity log (DVL) is a device that is used to obtain the three-dimensional speed vector of the vehicle. A typical DVL device (see Figure 3.6) consists of 4 transceiver units. This instrument, mounted on the vehicle, sends out a sound signal and the acoustic pulses are reflected from the seafloor or from the surface. Then, the vehicle measures the Doppler shift of its return. From this shift, the vehicle's velocity is calculated and is used to determine position and depth. The measurement is called "bottom-lock" if the measured reflection is from the seafloor or "surface-lock" if the reflection from the surface is measured. DVL can be used to obtain a three-dimensional linear velocity vector as well as a three dimensional angular velocity vector in the body-fixed frame of the vehicle. They have become increasingly popular in AUVs and ROVs, which is a result of the significantly decreased size of available devices [109, 2].

#### **3.4.4.4 Dead-reckoning Navigation**

A Dead-Reckoning system is a basic requirement for any AUV. It involves simple measurements of bearing or magnetic heading, depth, ground velocity, and travelled distance. The AUV uses an on-board digital compass for bearing and a pressure sensor to measure the depth, Section 3.4.4.2. It also calculates the velocity and displacement based on thruster commands, with this assumption that there is a certain linear dependency between propeller RPM and forward vehicle speed. Ease of use and affordability are the main advantages of this approach. When using dead reckoning, due to the integrative nature of the position estimate, the error of the position estimate will grow unbounded. The growth in this error is caused by errors in the velocity and attitude estimates which are in turn affected by the accuracy of the navigational filter and the accuracies of the measurements observing these states.

#### **3.4.4.5 Inertial Navigation System (INS)**

Inertial navigation involves the detection of acceleration of the vehicle with the use of three orthogonally-mounted gyroscopic sensors.

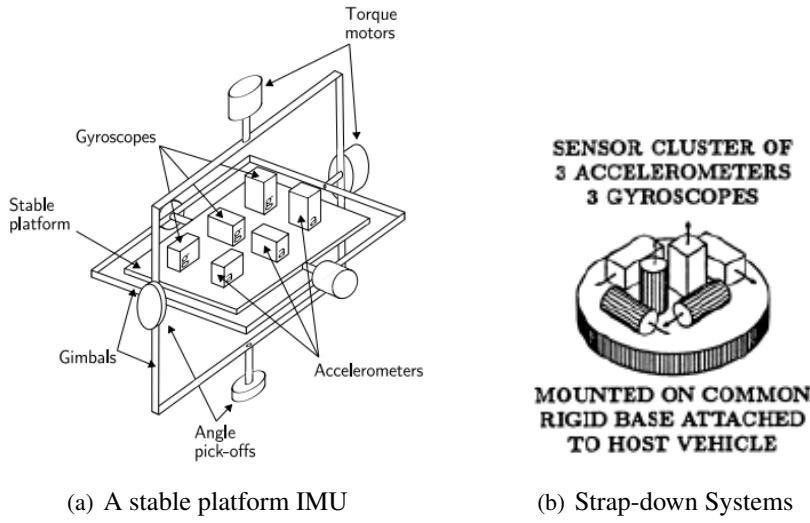


Figure 3.7: Two categories of INSs (image sources: Left [4], Right [7]).

Inertial navigation is a self-contained navigation technique in which measurements provided by accelerometers and gyroscopes are used to track the position and orientation of an object relative to a known starting point, orientation and velocity. Inertial Measurement Units (IMUs) typically contain three orthogonal rate-gyroscopes and three orthogonal accelerometers. This allows the system to measure linear and rotational accelerations in 3-dimensions, thereby track the position and orientation of the device and characterizing vehicle motion in all 6-Degree Of Freedom, see [111, 114, 115]. Nearly all IMUs fall into one of the two categories outlined below. The difference between the two categories is the frame of reference in which the rate-gyroscopes and accelerometers operate.

- **Stable Platform Systems** In stable platform type systems the inertial sensors are mounted on a platform which is isolated from any external rotational motion. In other words the platform is held in alignment with the global frame, Figure 3.7(a). This is achieved by mounting the platform using frames which allow the platform freedom in all three axes, The platform mounted gyroscopes detect any platform rotations. These signals are fed back to torque motors which rotate the gimbals in order to cancel out such rotations, hence keeping the platform aligned with the global frame.
- **Strap-down Systems** In strap-down systems the inertial sensors are mounted rigidly onto the device, and therefore output quantities measured in the body frame rather than the global frame, Figure 3.7(b). To keep track of orientation the signals from the rate gyroscopes are “integrated”. To track position the three accelerometer signals are resolved into global co-ordinates using the known orientation, as determined by the integration of the gyroscope signals. The global acceleration signals are then integrated as in the stable platform algorithm.

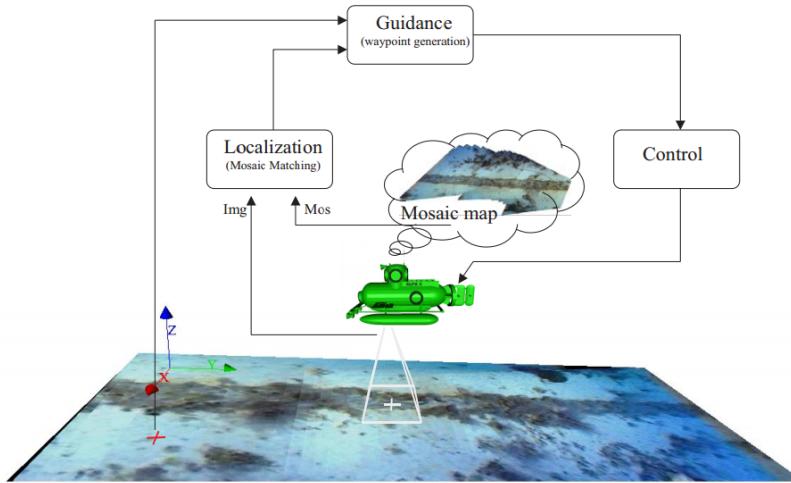


Figure 3.8: Overall visual servoing control scheme.(image source: [117])

Stable platform and strap-down systems are both based on the same underlying principles. Strap-down systems have reduced mechanical complexity and tend to be physically smaller than stable platform systems. These benefits are achieved at the cost of increased computational complexity. As the cost of computation has decreased strap-down systems have become the dominant type of INS.

However, the problem with this approach is that the accuracy of navigation can deteriorate rapidly due to accumulated error in integrating of these accelerations over time. This error can be reduced through state-estimation methods such as Kalman Filtering, and/or completely bounded by periodic resurfacing and using absolute position fixes, through the use of Differential GPS (see [111] and [116]). This is a significant improvement over dead reckoning and is often combined with a Doppler velocity log (DVL) that can measure the vehicle relative velocity [111].

#### 3.4.4.6 Visual Navigation

Visual navigation systems are mainly used for object identification, following, and collision avoidance. Due to attenuation of light underwater, this approach only is suitable for short distance. In literature, a set of algorithms for the creation of underwater mosaics and their use as visual maps for underwater vehicle navigation are presented, Figure 3.8. An automatic video mosaics is being created, which deals with the problem of image motion estimation in a robust and automatic way. The motion estimation is based on a initial matching of corresponding areas over pairs of images, followed by the use of a robust matching technique, which can cope with a high percentage of incorrect matches. Several motion models, established under the projective geometry framework, allow for the creation of high quality mosaics where no assumptions are made about the camera motion [118, 117].

#### 3.4.4.7 Acoustic Navigation

Acoustic navigation systems can be used for navigation over long distances, due to high speed of sound underwater (avg. 1500 m/s in seawater). This method can be used in environments that are opaque to the radio-frequencies upon which the GPS relies, as in subsurface, and are more affordable than conventional INSs. Several types of acoustic systems are available for AUVs, including Long-Base-Line (LBL), Ultra-Short-Base-Line (USBL), Sound Navigation And Ranging (SONAR), and Acoustic Doppler (ADCPs & DVLs) technologies (see [119], [120] and [121]). Deep-sea research got a boost with the invention of sonar in 1914. Developed to detect icebergs at night or in fog, sonar quickly proved to be an excellent depth-finder, much faster than the old sounding line. Sonar is based on the principle that sound travels through water at a rapid and fairly constant rate. A “pinger” mounted underwater on a ship’s hull sends out periodic bursts of sound. The sound waves bounce off obstacles and get reflected back to the ship. By measuring how long it takes for the reflected waves to return to the ship, the distance to an obstacle can be calculated. Finally, it is concluded that only geophysically referenced methods will enable AUVs to navigate accurately over large areas and that advances in underwater feature recognition are required before these methods can be implemented in operational AUVs.

## 3.5 CTD adaptive sampling strategy

As mentioned before, this work deals with a cooperative coordinated formation of AUVs. We assume that one vehicle always plays the role of leader and the rest as follower(s). The task of the leader vehicle is to follow some predefined path profile while is acquiring CTD data. The aim of the each follower is to keep a coordinated formation with respect to the leader by keeping a desired distance from it. This distance is a function of a dissimilarity measure, which is the *variational* distance between its own GMM hypothesis and the leader. This measure,  $\zeta$ , provides a quantity of how different or similar is the CTD data between the leader and each follower at the time of resurfacing. As mentioned before, we assume that all followers, adjust their position during each resurfacing with respect to the leader. A control strategy is adopted to generate speed and heading commands so as to drive suitably defined along track and cross track errors to zero. The commands are used as input to local inner loops for yaw and speed control.

In practice, executing this type of mission without expensive inertial sensor suites requires the follower vehicles to manoeuvre into formation by relying on measurements of their distances to the leading vehicles and exchanging complementary data. This entails considerable difficulties underwater, as conventional communication and localization systems (like GPS) are unavailable and usually replaced by acoustic devices: acoustic modems that allow the exchange of data, and ranging devices that estimate distances by measuring time-of-flight of acoustic signals. These devices exhibit a number of constraints that are inherent to the medium, such as temporary communication losses, outliers in the range measurements, and low bandwidth of the acoustic communication systems. In practice, an important consequence of these limitations is the inability to measure

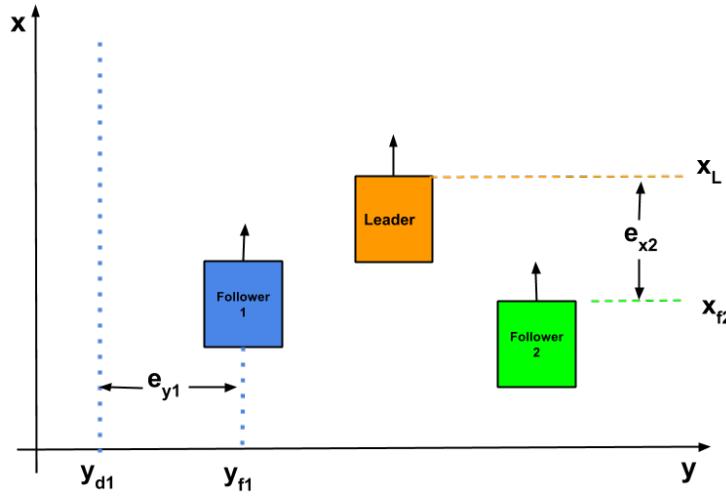


Figure 3.9: Schematic representation of the setup

or communicate frequently, with inter-sample times often in the range of seconds, making the problem of underwater range-based multiple vehicle formation keeping very challenging.

### 3.5.1 Path Following

As it can be seen in Figure 3.9, the vehicles are required to follow segments of line, which for simplicity and without loss of generality are perpendicular to the  $y$  axis. During each resurfacing, every follower separately regulates the desired linear velocity  $u_d$  and heading  $\psi_d$ . These are then fed to inner loop controllers specific to the vehicle. A simplified version of the kinematic model of an underwater vehicle can be described as follows

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} u \cos \psi - v \sin \psi \\ u \sin \psi - v \cos \psi \\ r \end{bmatrix} \quad (3.4)$$

where  $u$  is the surge velocity,  $v$  is the sway velocity,  $r$  is the angular velocity and  $\psi$  is the heading. Using the path-following algorithm in [124], we have

$$e_y = y - y_d \Rightarrow \dot{e}_y = \dot{y} - \dot{y}_d = u \sin \psi - 0 = uU \quad (3.5)$$

where  $U = \sin \psi$ , and  $y_d$  define the desired position of the line to be followed by the AUV, which will correspond to the desired distance from the leader AUV (see Figure 3.9). Therefore, if  $U = \sin \psi$  we obtain  $\dot{e}_y = -k_1 e_y$ , which implies that the error  $e_y$  will converge exponentially to zero. The desired heading angle can be obtained as

$$\psi_d = \sin^{-1}(\text{sat}(U)) \quad (3.6)$$

Table 3.1: Simulation parameters

$\varepsilon$	$k_1$	$k_\psi$	$k_u$	$k_2$
0.8	1	1.5	1.2	1

where

$$sat(U) = \begin{cases} U & \text{if } |U| < \varepsilon \\ \varepsilon & \text{if } U > \varepsilon \\ -\varepsilon & \text{if } U < -\varepsilon \end{cases} \quad (3.7)$$

and  $\varepsilon \in (0, 1)$ . A possibility for the inner-loop in heading can be derived by noting that

$$\tilde{\psi} = \psi - \psi_d \Rightarrow \dot{\tilde{\psi}} = r - \dot{\psi}_d \quad (3.8)$$

Thus, if  $r = -k_\psi \tilde{\psi} = -k_\psi(\psi - \psi_d)$ ,  $k_\psi > 0$  it can be concluded that  $\psi$  will converge to  $\psi_d$  for  $\psi_d$  constant. Note that it is possible to show, using Lyapunov-based analysis tools, that the above nonlinear control law yields convergence of the cross track error to zero if the actual vehicle heading equals the desired heading reference  $\psi_d$ . The work in [124] also shows that “identical behaviour” is obtained when the dynamics of the heading autopilot (inner loop) and the sideslip of the vehicle are taken into account. Table 3.1 contains the parameters of this controller that are fixed throughout the simulation experiment.

### 3.5.2 Coordinated Formation

For simplicity we consider a formation in line parallel with the  $y$  axis as illustrated in Figure 3.9. To keep the coordinated formation, a follower during each resurfacing needs to correct its position with respect to the leader. To achieve this goal, a PI controller is used

$$\dot{\xi} = e_x, u_f = -k_u e_x - k_2 \xi \quad (3.9)$$

where  $e_x$  is the difference between the position of the leader and each follower along the  $x$  axis (see Figure 3.9).

### 3.5.3 Variational distance between GMMs

The Gaussian mixture model defined in this work reflects the CTD feature distributions of an environment by a linear combination of Gaussian densities. In this way, the density of a single feature or the probability of regions in the feature space depends on all components of the Gaussian mixture model, Equation (2.1). Thus, this model corresponds to the soft assignment approach of modelling image contents [23]. Kullback Leibler (KL) divergence, which is a widely used tool in

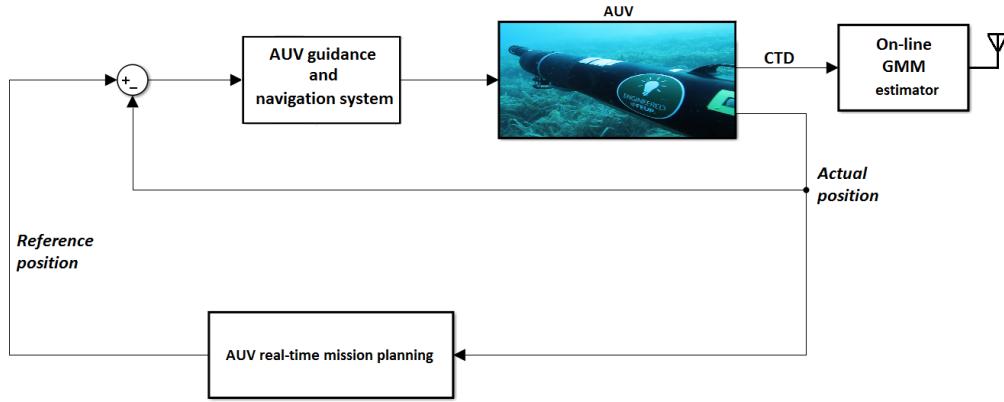


Figure 3.10: Schematic representation of the leader

statistics and pattern recognition, to measure the similarity between them as follows:

$$D(f\|g) = \int f(x) \log \frac{f(x)}{g(x)} dx \quad (3.10)$$

Note that unlike two single Gaussian distributions, there is no closed form expression to measure the similarity between two GMMs. Thus, in this work we use the *variational* approximation [20] to measure the dissimilarity between GMM hypotheses, that is,

$$\zeta = D_{\text{variational}}(f\|g) = \sum_a w_a \log \frac{\sum_{a'} w_{a'} e^{-D(f_a\|f_{a'})}}{\sum_b w_b e^{-D(f_a\|g_b)}} \quad (3.11)$$

where  $w_a$  and  $w_{a'}$  are the mixing weights of the components in  $f(x)$  and  $w_b$  is the mixing weights of the components in  $g(x)$ .

## 3.6 Simulation Results

In this section, the performance of the proposed algorithm is demonstrated in uniform and changeable environments which were reconstructed based on real CTD data.

### 3.6.1 The Leader AUV

Figure 3.10 shows a schematic representation of a leader AUV. As mentioned above, its task is to follow some desired predefined path, reference position, while acquiring CTD data. This data, is fed to the on-line GMM estimator to create a CTD profile for the leader. The AUV real-time mission planner is a high-level decision maker on-board the AUV. It monitors all system states and issues the speed and steering commands to the low-level thruster controllers (AUV guidance and navigation block). In literature, the problem of position trajectory-tracking and path-following for AUVs have been addressed repeatedly, e.g. see [122], and [123]. The low-level AUV controller set consists of three controllers, depth, speed and heading, which are dedicated to receiving commands

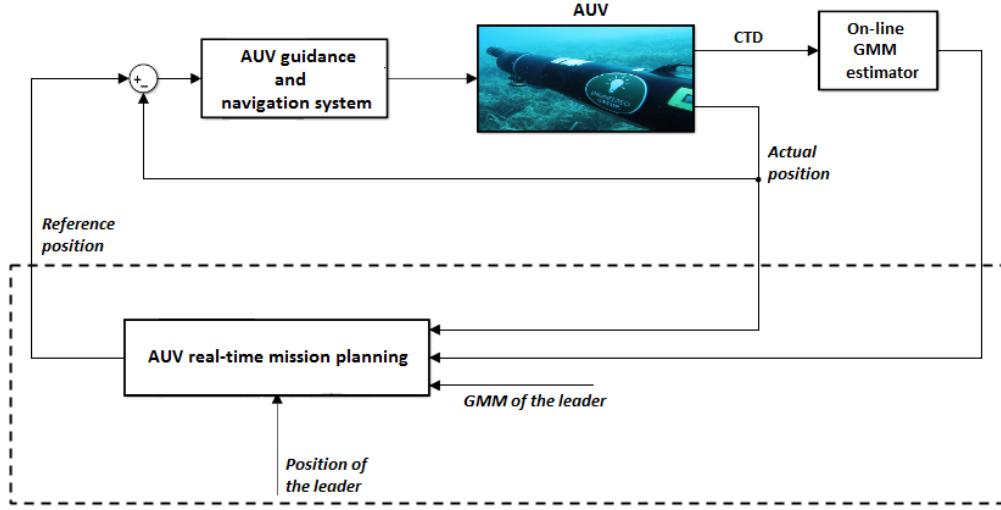


Figure 3.11: Schematic representation of the follower

from the reference controller and generating thruster outputs based on error between commanded and actual depth and heading.

### 3.6.2 The follower AUV

A scheme of a follower AUV, is represented in Figure 3.11. Similar to the leader, each follower AUV captures CTD data in real-time and construct its own GMM hypothesis. Unlike the leader, the desired path is not predefined and has to be updated in each resurfacing (adaptive sampling). In fact, the task of the follower, can be divided into on-surface and subsurface.

In each resurfacing, using the available GPS signals, the follower can correct its own navigation errors as suggested in [123] and learn the position of the leader. Suitable velocity command can be issued to have a coordinated formation with respect to the last seen position of the leader. Furthermore, on surface, the follower can learn the CTD profile, GMM hypothesis, of the leader and measure the dissimilarity measure,  $\zeta$ , according to Equation (3.11). If the dissimilarity is high, the follower continues the current formation. On the other hand, too small dissimilarity measure means that expanding of the formation, to sample new environment is needed. In subsurface, the vehicle has to follow the last updated mission plan as in [122].

Worth to be mentioned that, some limits were put for the acceleration of the vehicles and expansion of the formation, to address the various operational constraints associated with real world applications. To maintaining the group structure: 1) attraction to the leader up to a maximum distance 2) repulsion from the leader when it is too close 3) alignment or velocity matching with the leader. Every follower makes its own decision based on the difference or distance between its GMM and the leader. In particular, for the case of one single follower (which means 2 AUVs), the task would consist in keeping the follower AUV at a convenient distance from the leader, whose distance depends on the CTD that is being measured. Figure 3.12, shows a real CTD data that

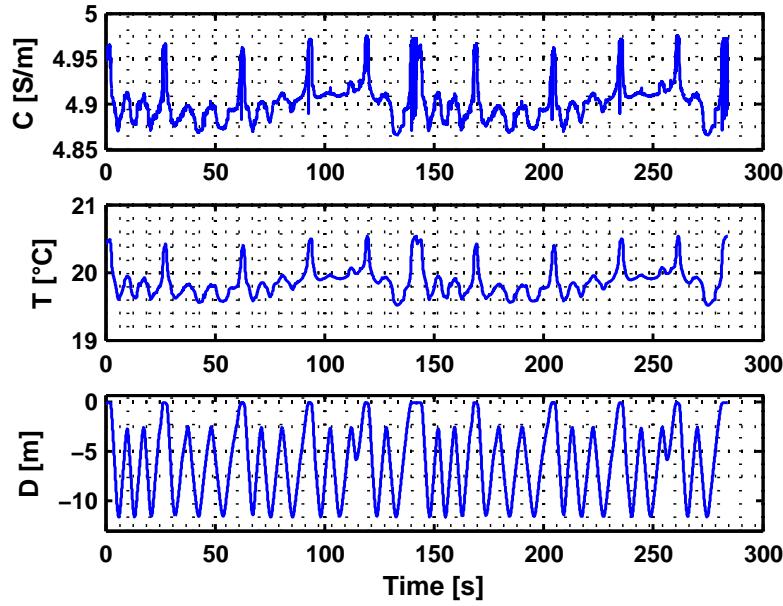


Figure 3.12: Real CTD data

was used in the simulations. A typical scenario is to make the follower vehicle close to the leader when the difference of the CTDs is large, and on the contrary, increase the distance when the difference is small. To implement this strategy, we propose the following steps to compute the error or dissimilarity measure:

- The CTD data is modelled as a Gaussian Mixture Model (GMM) and each vehicle runs an on-line unsupervised learning algorithm to estimate the GMM parameters. As mentioned in Chapter 2, the number of components of the GMM is not fixed and can change over time, which provides more flexibility to deal with uncertain dynamic environments.
- During each resurfacing, every follower learn the last seen position of the leader and its GMM hypothesis. Then, together with its own estimated GMM parameters, compute an error that measures the dissimilarity of the two GMM probability density functions given by a closed-form formula (the *variational* approximation) that measures the Kullback Leibler (KL) divergence, see [20].

The desired position  $y_d$  for each follower corresponds to the case when variational distance  $\zeta$  becomes equal to some reference value  $\zeta_r$ , and it is given by

$$\begin{aligned} e_\zeta &= (\zeta - \zeta_r) / \zeta_r \\ y_d &= y_f + \lambda e_\zeta \end{aligned} \tag{3.12}$$

where  $\lambda$  is a constant value and can be positive or negative, depending on which direction we want to guide the follower.  $y_d$  is the desired position in  $x - y$  plane and  $y_f$  is the current position of the follower. The intuition behind the normalization of  $e_\zeta$  between  $[0, 1]$  is to moderate the changes in

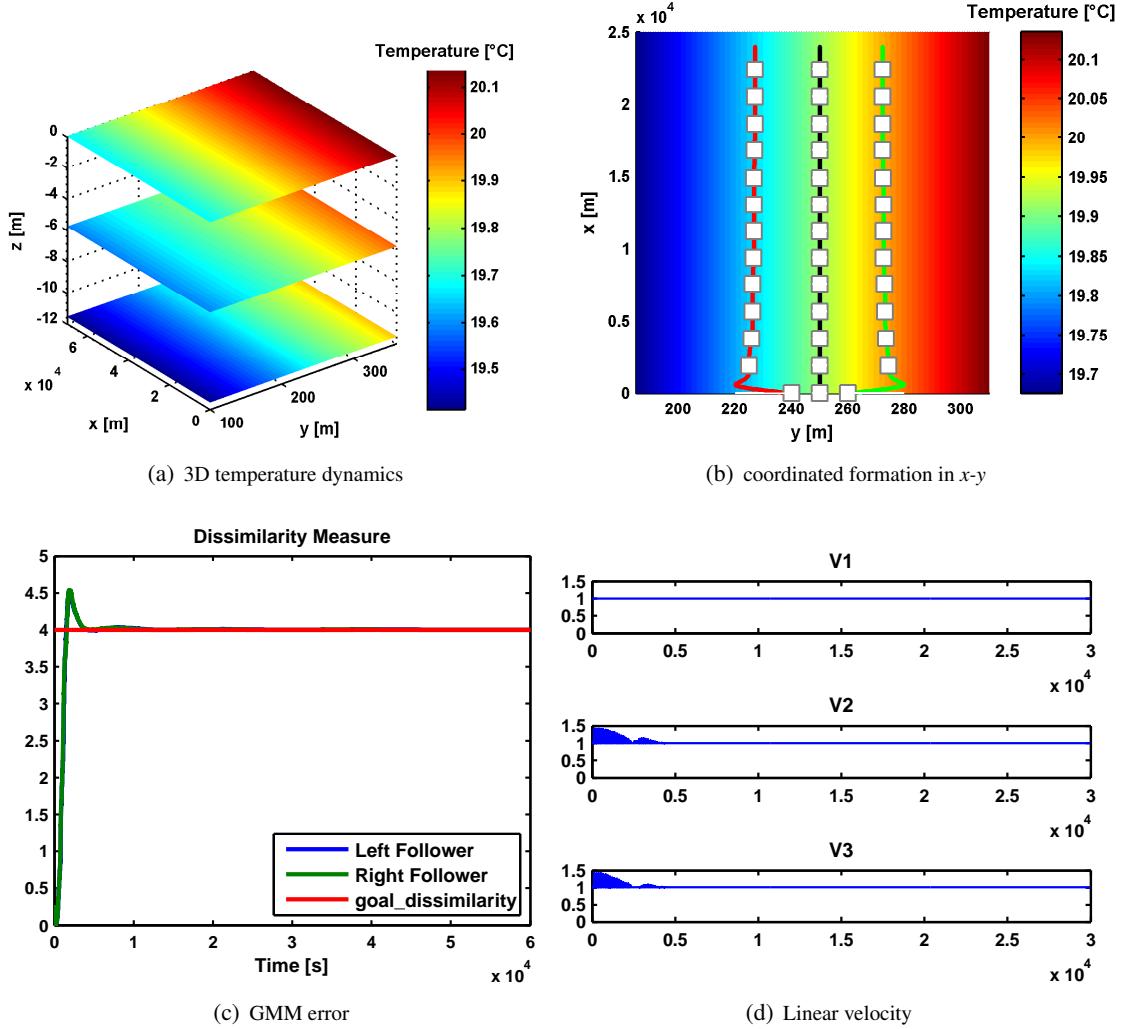


Figure 3.13: Uniform environment.

the formation with respect to the value of  $\zeta$  specially at the beginning of the experiment (zoom-in and zoom-out of the formation be more dependent on user-defined value of  $\lambda$  than unknown value of  $\zeta$ ).

### 3.6.3 Uniform Environment Simulation

Figure 3.13(a) shows the temperature field of a reconstructed environment, where the top layer represents the near surface and the lowest one is  $12m$  beneath the surface. In this experiment, we assume that the environment is uniform along the  $x$  axis. As expected, we see that the average temperature on the surface is higher than in depth. We assumed a fleet of three vehicles, and the leader is in the middle. Every vehicle could go down to  $12m$  below the surface, in a saw-tooth

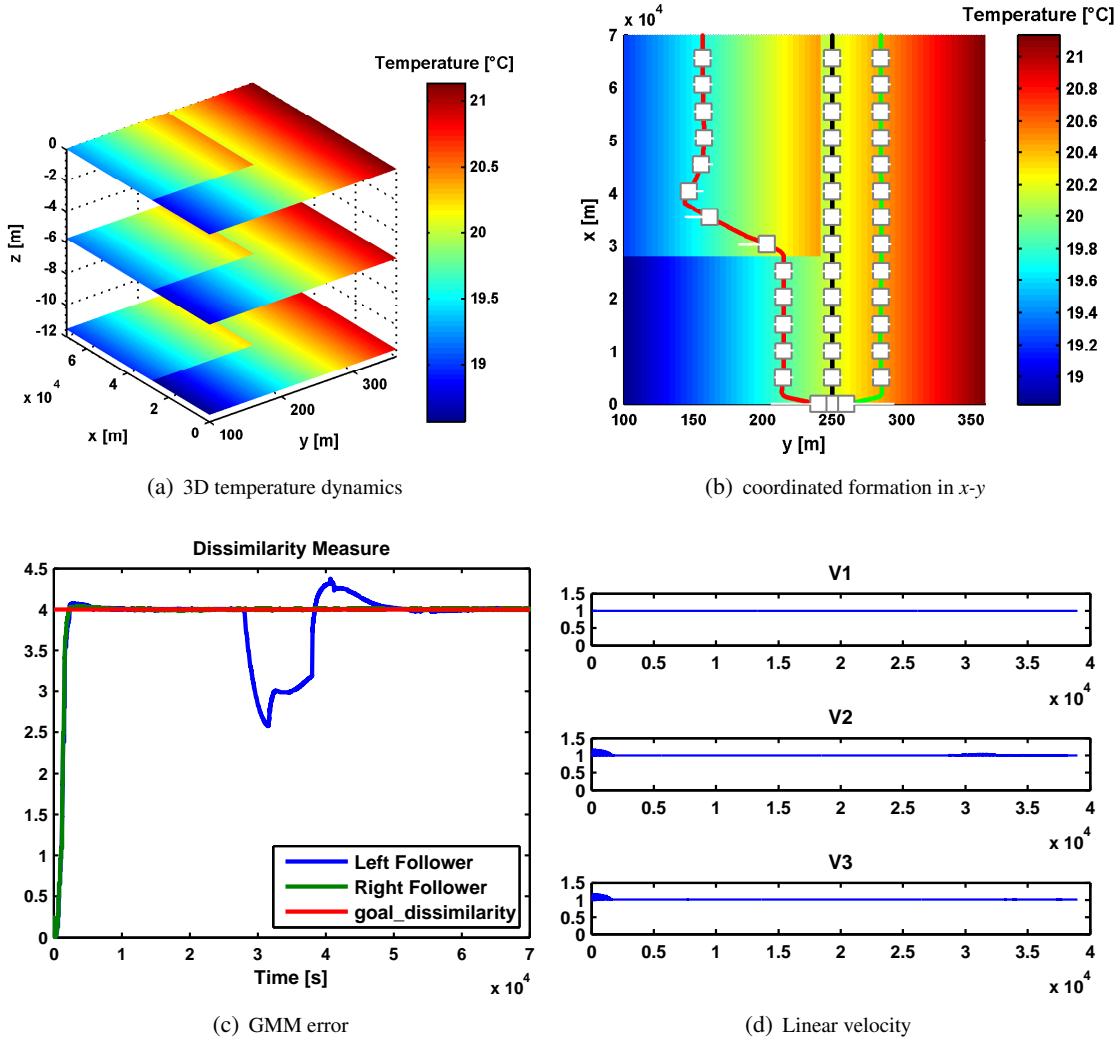


Figure 3.14: Uneven environment.

pattern. The mission plan for the leader is to follow a line, in parallel with the  $x$  axis, with a constant speed. The followers need to adapt their speed and distance from the leader to maintain the coordinated formation of the fleet. In other words, every follower has to expand the formation to find the regions where the CTD data is desirably different from the leader. Figure 3.13(b) shows the overall performance of the fleet, from the top view, and the position of the vehicles during resurfacing intervals in  $x$ – $y$  plane. Figure 3.13(c), compatible with Figure 3.13(b) and based on Equation (3.12), shows by expanding the formation at the beginning of the experiment, the followers find the desirable environment. After moving to that position, the fleet goes on as a coordinated group along the  $x$  axis. The objective is that during each resurfacing, each follower learn the last seen position of the leader and its GMM hypothesis to update its own mission plan. Figure 3.13(d) shows the linear velocity changes of each vehicle at the beginning of the experiment. As mentioned before, the leader travels with a constant speed. And each follower, adjust its velocity in

each resurfacing to keep the coordinated formation.

### 3.6.4 Complex Environment Simulation

For the second task, we assume that the environments on the left side of the leader is not uniform and changes at some point in  $x - y$  frame, see Figure 3.14(a). Like the previous case, the followers learn their path toward desired position according to Equation (3.12). Figure 3.14(b) shows that at some point the left follower enters a new environment which leads to changes in dissimilarity measure, see Figure 3.14(c). Therefore, after some time the left AUV successfully relearn the desired path. Figure 3.14(d) confirms the results.

## 3.7 Conclusion

This Chapter demonstrated the performance of the unsupervised learning of GMM algorithm proposed in the previous chapter, qualitatively and quantitatively in CTD adaptive sampling for a fleet of Autonomous Underwater Vehicles (AUVs). The practical implementation requires a careful tuning of the gain parameters. The manoeuvrability of the vehicles, e.g. how fast the vehicles can place themselves in the desired position or the greatness of the displacement based on the dissimilarity measure can be decisive in the performance of the model. For example with very slow linear velocity the followers would resurface before reaching the desired position which was learned during the last resurfacing and this could jeopardise the overall performance of the model, especially convergence to the desired position in reasonable time. The simulation results show the feasibility and accuracy of the motion learning strategies in many uniform and complex environments.



# Chapter 4

## Conclusions and Future Work

### 4.1 Conclusions

This thesis addressed the problem of motion learning and adaptive sampling for a fleet of autonomous underwater vehicles. An overview of sensor systems and navigation techniques is given as well as a brief description of the main motion control system formed with a path following controller, a path generator and coordination control. A novel unsupervised learning of Gaussian mixture models in the presence of dynamic environments is presented. The algorithm relies on a multi-hypothesis adaptive scheme that continuously updates the number of components and estimates the model parameters as the measurements (sample data) are being acquired. The hypothesis models are ranked according to the minimum description length. The virtue of this approach is the fact that it addresses explicitly the case that the complexity of the GMM is not only unknown, but it also can change over time. The proposed algorithm has the additional feature that it relaxes “the sufficiently large data set” restriction by not requiring in fact any initial batch of data.

The simulations that are presented here led to a variety of conclusions and assumptions. A sparingly important assumption is that cooperative unsupervised motion learning for the followers is reasonable, only if the navigation of an individual vehicle is reliable. The performance of the proposed method is demonstrated qualitatively and quantitatively in CTD adaptive sampling for a fleet of Autonomous Underwater Vehicles (AUVs).

The practical implementation requires a careful tuning of the gain parameters that has been successfully done as demonstrated with experimental data. It opens up the possibility of oceanographic missions conducted by a team of vehicles. In the simulations a Gaussian noise is considered for the pressure sensor. The communication constraints underwater are extreme, i.e., the followers only on the surface can “talk” to the leader and learn its position and GMM hypothesis. The manoeuvrability of the vehicles, e.g. how fast the vehicles can place themselves in the desired position or the greatness of the displacement based on the dissimilarity measure can be decisive in the performance of the model. For example, if the followers do not move fast enough in the horizontal plane (linear velocity be too low), the vehicles can resurface before reaching to the desired position that was learned during the last resurfacing and this could jeopardise the overall

performance of the model and the convergence to the desired position in reasonable time.

A further important property of the proposed unsupervised motion learning strategy is that, the leader can distribute its individual errors to the whole formation. Because the position of the followers is a function of the position of the leader and its sampling profile in terms of Gaussian mixture models. As a results, high navigation errors and sampling noises can disturb and aggravate the coordinated formation dramatically.

This suggests that a higher number of vehicles, more than one leader, can improve the quality of the estimations. The simulations have shown that all strategies are very sensitive towards the values of parameters.

## 4.2 Future Work

This thesis showed the strengths and the weaknesses of a novel real-time unsupervised motion learning of AUVs strategy based on Gaussian mixture models. The coordinated formation strategy was evaluated through computer simulations but using real CTD data. In order to implement the proposed algorithms in real systems, further steps are necessary.

Defining a more complex scenario, where the number of leader is more than one or the follower AUVs can communicate and exchange GMM hypotheses can help to improve the overall performance of the model. This would make the overall system more robust and less vulnerable to navigation and sampling error could be the direct result of that.

The most important next step is the validation of this strategy with real data and experiments with real AUVs in a test laboratory and later in an ocean environment. These steps are necessary to study the behaviour of the algorithms under non Gaussian noise and the effect of outliers.

# References

- [1] Monterey bay aquarium research institute. <http://www.mbari.org/default.htm>.
- [2] Woods hole oceanographic institution. <http://www.whoi.edu/>.
- [3] John r. delaney. <http://ooi.washington.edu/rsn/jrd/index.html>.
- [4] Interactive oceans. <http://www.interactiveoceans.washington.edu/>.
- [5] Douglas A Stow, Allen Hope, David McGuire, David Verbyla, John Gamon, Fred Huemmrich, Stan Houston, Charles Racine, Matthew Sturm, Kenneth Tape, et al. Remote sensing of vegetation and land-cover change in arctic tundra ecosystems. *Remote Sensing of Environment*, 89(3):281–308, 2004.
- [6] NN Soreide, CE Woody, and SM Holt. Overview of ocean based buoys and drifters: Present applications and future needs. In *OCEANS, 2001. MTS/IEEE Conference and Exhibition*, volume 4, pages 2470–2472. IEEE, 2001.
- [7] Ocean explorer. <http://oceandiscovery.noaa.gov/welcome.html>.
- [8] Steven S Brown, Harald Stark, Steven J Ciciora, and AR Ravishankara. In-situ measurement of atmospheric no<sub>3</sub> and n<sub>2</sub>o<sub>5</sub> via cavity ring-down spectroscopy. *Geophysical research letters*, 28(17):3227–3230, 2001.
- [9] Nuno Cruz, Aníbal Matos, Sérgio Cunha, Sérgio Silva, and RDR Frias. Zarco-an autonomous craft for underwater surveys. *Proceedings of the 7th Geomatic Week, Barcelona, Spain*, 2007.
- [10] Charles C Eriksen, T James Osse, Russell D Light, Timothy Wen, Thomas W Lehman, Peter L Sabin, John W Ballard, and Andrew M Chiodi. Seaglider: A long-range autonomous underwater vehicle for oceanographic research. *Oceanic Engineering, IEEE Journal of*, 26(4):424–436, 2001.
- [11] Daniel L Rudnick, Russ E Davis, Charles C Eriksen, David M Fratantoni, and Mary Jane Perry. Underwater gliders for ocean research. *Marine Technology Society Journal*, 38(2):73–84, 2004.
- [12] Joseph Curcio, John Leonard, and Andrew Patrikalakis. Scout-a low cost autonomous surface platform for research in cooperative autonomy. In *OCEANS, 2005. Proceedings of MTS/IEEE*, pages 725–729. IEEE, 2005.
- [13] Justin E Manley. Unmanned surface vehicles, 15 years of development. In *OCEANS 2008*, pages 1–4. IEEE, 2008.

- [14] Gwyn Griffiths and Ia Edwards. Auvs: designing and operating next generation vehicles. *Elsevier Oceanography Series*, 69:229–236, 2003.
- [15] Elgar Desa, R Madhan, and P Maurya. Potential of autonomous underwater vehicles as new generation ocean data platforms. *Current science*, 90(9):1202–1209, 2006.
- [16] Charles W Warren. A technique for autonomous underwater vehicle route planning. *Oceanic Engineering, IEEE Journal of*, 15(3):199–204, 1990.
- [17] Laboratório de Sistemas e Tecnologia Subaquática. Light autonomous underwater vehicles. <http://lsts.fe.up.pt/>.
- [18] Thomas B Curtin, James G Bellingham, Josko Catipovic, and Doug Webb. Autonomous oceanographic sampling networks. *Oceanography*, 6(3):86–94, 1993.
- [19] Mario A. T. Figueiredo and Anil K. Jain. Unsupervised learning of finite mixture models. volume 24, pages 381–396, 2000.
- [20] John R Hershey and Peder A Olsen. Approximating the kullback-leibler divergence between gaussian mixture models. In *ICASSP (4)*, pages 317–320, 2007.
- [21] Andrea Baraldi and Ethem Alpaydin. Constructive feedforward art clustering networks. i. *Neural Networks, IEEE Transactions on*, 13(3):645–661, 2002.
- [22] Vladimir Cherkassky and Filip M Mulier. *Learning from data: concepts, theory, and methods*. John Wiley & Sons, 2007.
- [23] Christopher M. Bishop. Pattern recognition and machine learning. Springer, 2006. URL: </bib/bishop/Bishop2006/PatternRecognitionAndMachineLearningBishop-.pdf>.
- [24] Bernd Fritzke. Some competitive learning methods. *Artificial Intelligence Institute, Dresden University of Technology*, 1997.
- [25] Ralph Gross, Jie Yang, and Alex Waibel. Growing gaussian mixture models for pose invariant face recognition. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 1, pages 1088–1091. IEEE, 2000.
- [26] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.
- [27] Rui Xu, Donald Wunsch, et al. Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, 16(3):645–678, 2005.
- [28] Nicola Greggio, Alexandre Bernardino, and José Santos-Victor. A practical method for self-adapting gaussian expectation maximization. In *ICINCO (1)*, pages 36–44, 2010.
- [29] Nizar Bouguila, Djemel Ziou, and Jean Vaillancourt. Unsupervised learning of a finite mixture model based on the dirichlet distribution and its application. *Image Processing, IEEE Transactions on*, 13(11):1533–1543, 2004.
- [30] Zheng Rong Yang and Mark Zwolinski. Mutual information theory for adaptive mixture models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(4):396–403, 2001.

- [31] Nikos Vlassis and Aristidis Likas. A kurtosis-based dynamic approach to gaussian mixture modeling. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 29(4):393–399, 1999.
- [32] Jakob J Verbeek, Nikos Vlassis, and B Kröse. Efficient greedy learning of gaussian mixture models. *Neural computation*, 15(2):469–485, 2003.
- [33] Naonori Ueda, Ryohei Nakano, Zoubin Ghahramani, and Geoffrey E Hinton. Smem algorithm for mixture models. *Neural computation*, 12(9):2109–2128, 2000.
- [34] Zhihua Zhang, Chibiao Chen, Jian Sun, and Kap Luk Chan. Em algorithms for gaussian mixtures with split-and-merge operation. *Pattern recognition*, 36(9):1973–1983, 2003.
- [35] Sebastian Thrun, Christian Martin, Yufeng Liu, Dirk Hahnel, Rosemary Emery-Montemerlo, Deepayan Chakrabarti, and Wolfram Burgard. A real-time expectation-maximization algorithm for acquiring multiplanar maps of indoor environments with mobile robots. *Robotics and Automation, IEEE Transactions on*, 20(3):433–443, 2004.
- [36] Zoran Zivkovic and Ferdinand van der Heijden. Recursive unsupervised learning of finite mixture models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(5):651–656, 2004.
- [37] P. Hall, D. Marshall, and R. Martin. Merging and splitting eigenspace models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(9):1042–1049, Sep 2000. doi:[10.1109/34.877525](https://doi.org/10.1109/34.877525).
- [38] Mingzhou Song and Hongbin Wang. Highly efficient incremental estimation of gaussian mixture models for online data stream clustering. In *Defense and Security*, pages 174–183. International Society for Optics and Photonics, 2005.
- [39] Yulia Alexandrovna Hicks, Peter M Hall, and Andrew David Marshall. A method to add hidden markov models with application to learning articulated motion. 2003.
- [40] Peter M Hall, Yulia Hicks, and Tony Robinson. A method to add gaussian mixture models. 2005.
- [41] Nuno Vasconcelos and Andrew Lippman. Learning mixture hierarchies. In *NIPS*, pages 606–612. Citeseer, 1998.
- [42] Ognjen Arandjelovic and Roberto Cipolla. Incremental learning of temporally-coherent gaussian mixture models. *Society of Manufacturing Engineers (SME) Technical Papers*, pages 1–1, 2006.
- [43] P M Hall and Y Hicks. A method to add gaussian mixture models. Other, April 2004. ID number: CSBU-2004-03. URL: <http://opus.bath.ac.uk/16849/>.
- [44] Michael R Anderberg. Cluster analysis for applications. Technical report, DTIC Document, 1973.
- [45] S Zhuang, Yan Huang, Kannappan Palaniappan, and Yunxin Zhao. Gaussian mixture density modeling, decomposition, and applications. *Image Processing, IEEE Transactions on*, 5(9):1293–1302, 1996.

- [46] Dawei Li, Lihong Xu, and Erik Goodman. On-line em variants for multivariate normal mixture model in background learning and moving foreground detection. *Journal of mathematical imaging and vision*, 48(1):114–133, 2014.
- [47] D Michael Titterington. Recursive parameter estimation using incomplete data. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 257–267, 1984.
- [48] Vaclav Fabian. On asymptotically efficient recursive estimation. *The Annals of Statistics*, pages 854–866, 1978.
- [49] Lei Xu and Michael I Jordan. On convergence properties of the em algorithm for gaussian mixtures. *Neural computation*, 8(1):129–151, 1996.
- [50] Richard A Redner and Homer F Walker. Mixture densities, maximum likelihood and the em algorithm. *SIAM review*, 26(2):195–239, 1984.
- [51] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2012.
- [52] Geoffrey McLachlan and David Peel. *Finite mixture models*. John Wiley & Sons, 2004.
- [53] Geoffrey McLachlan and Thriyambakam Krishnan. *The EM algorithm and extensions*, volume 382. John Wiley & Sons, 2007.
- [54] G.J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. John Wiley & Sons, New York, 1997.
- [55] Chi-hau Chen, Louis-François Pau, and Patrick Shen-pei Wang. *Handbook of pattern recognition and computer vision*. World Scientific, 2010.
- [56] Glenn W Milligan and Martha C Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2):159–179, 1985.
- [57] B.S. Everitt, S. Landau, M. Leese, and D. Stahl. *Cluster Analysis*. Wiley series in probability and statistics. Wiley, 2011. URL: <http://books.google.pt/books?id=w3bE1kqd-48C>.
- [58] Padhraic Smyth. Clustering using monte carlo cross-validation. In *KDD*, pages 126–133, 1996.
- [59] Hirotugu Akaike. Likelihood of a model and information criteria. *Journal of econometrics*, 16(1):3–14, 1981.
- [60] Michael P Windham and Adele Cutler. Information ratios for validating mixture analyses. *Journal of the American Statistical Association*, 87(420):1188–1192, 1992.
- [61] Dan Pelleg and Andrew W. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 - July 2, 2000*, pages 727–734, 2000.
- [62] Gideon Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

- [63] Chris Fraley and Adrian E Raftery. Bayesian regularization for normal mixture estimation and model-based clustering. *Journal of Classification*, 24(2):155–181, 2007.
- [64] Abdolrahman Khoshrou and António Pedro Aguiar. Unsupervised learning of gaussian mixture models in the presence of dynamic environments. In *CONTROLO'2014—Proceedings of the 11th Portuguese Conference on Automatic Control*, pages 387–396. Springer International Publishing, 2015.
- [65] Mark Girolami. Mercer kernel-based clustering in feature space. *Neural Networks, IEEE Transactions on*, 13(3):780–784, 2002.
- [66] Ravi Kothari and Dax Pitts. On finding the number of clusters. *Pattern Recognition Letters*, 20(4):405 – 416, 1999. URL: <http://www.sciencedirect.com/science/article/pii/S0167865599000082>, doi:[http://dx.doi.org/10.1016/S0167-8655\(99\)00008-2](http://dx.doi.org/10.1016/S0167-8655(99)00008-2).
- [67] Gail A Carpenter and Stephen Grossberg. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer vision, graphics, and image processing*, 37(1):54–115, 1987.
- [68] Hichem Frigui and Raghu Krishnapuram. A robust competitive clustering algorithm with applications in computer vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(5):450–465, 1999.
- [69] Nozha Boujemaa. Generalized competitive clustering for image segmentation. In *Fuzzy Information Processing Society, 2000. NAFIPS. 19th International Conference of the North American*, pages 133–137. IEEE, 2000.
- [70] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.
- [71] Radford M Neal and Geoffrey E Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Springer, 1998.
- [72] Masa-Aki Sato and Shin Ishii. On-line em algorithm for the normalized gaussian network. *Neural computation*, 12(2):407–432, 2000.
- [73] Rohan A Baxter and Jonathan J Oliver. Mdl and mml: Similarities and differences. *Dept. Comput. Sci. Monash Univ., Clayton, Victoria, Australia, Tech. Rep*, 207, 1994.
- [74] Peter Grünwald. Model selection based on minimum description length. *Journal of Mathematical Psychology*, 44(1):133–152, 2000.
- [75] Jorma Rissanen. Stochastic complexity and modeling. *The Annals of Statistics*, pages 1080–1100, 1986.
- [76] Peter D Grünwald. *The minimum description length principle*. The MIT Press, 2007.
- [77] Aaron D Lanterman. Schwarz, wallace, and rissanen: Intertwining themes in theories of model selection. *International statistical review*, 69(2):185–212, 2001.
- [78] Chris S. Wallace and David L. Dowe. Minimum message length and kolmogorov complexity. *The Computer Journal*, 42(4):270–283, 1999.

- [79] Chris S Wallace and PR Freeman. Estimation and inference by compact coding. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 240–265, 1987.
- [80] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.
- [81]
- [82] Jorma Rissanen. Stochastic complexity. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 223–239, 1987.
- [83] Siddhartha Chib. Calculating posterior distributions and modal estimates in markov mixture models. *Journal of Econometrics*, 75(1):79–97, 1996.
- [84] Jonathan J Oliver, Rohan A Baxter, and Chris S Wallace. Unsupervised learning using-MML. In *ICML*, pages 364–372, 1996.
- [85] José M Bernardo and Adrian FM Smith. *Bayesian theory*, volume 405. John Wiley & Sons, 2009.
- [86] Sarunas J Raudys and Anil K. Jain. Small sample size effects in statistical pattern recognition: Recommendations for practitioners. *IEEE Transactions on pattern analysis and machine intelligence*, 13(3):252–264, 1991.
- [87] Samuel S Blackman. Multiple hypothesis tracking for multiple target tracking. *Aerospace and Electronic Systems Magazine, IEEE*, 19(1):5–18, 2004.
- [88] David J Salmond. Mixture reduction algorithms for target tracking in clutter. In *OE/LASE'90, 14-19 Jan., Los Angeles, CA*, pages 434–445. International Society for Optics and Photonics, 1990.
- [89] Yaakov Bar-Shalom, Peter K Willett, and Xin Tian. Tracking and data fusion. *A Handbook of Algorithms*. Yaakov Bar-Shalom, 2011.
- [90] David W Scott and William F Szewczyk. From kernels to mixtures. *Technometrics*, 43(3):323–335, 2001.
- [91] David F Crouse, Peter Willett, Krishna Pattipati, and Lennart Svensson. A look at gaussian mixture reduction algorithms. *Proceedings of the 14th International Conference on Information Fusion*, 2011.
- [92] G Valverde, J Quirós Tortós, and V Terzija. Comparison of gaussian mixture reductions for probabilistic studies in power systems. In *Power and Energy Society General Meeting, 2012 IEEE*, 2012.
- [93] Jason L Williams. Gaussian mixture reduction for tracking multiple maneuvering targets in clutter. Technical report, DTIC Document, 2003.
- [94] Jason L Williams and Peter S Maybeck. Cost-function-based gaussian mixture reduction for target tracking. In *Proceedings of the Sixth International Conference of Information Fusion*, volume 2, pages 1047–1054, 2003.
- [95] Andrew R Runnalls. Kullback-leibler approach to gaussian mixture reduction. *Aerospace and Electronic Systems, IEEE Transactions on*, 43(3):989–999, 2007.

- [96] Iris data set. <http://archive.ics.uci.edu/ml/datasets/Iris>.
- [97] A Aguiar, J Almeida, M Bayat, B Cardeira, R Cunha, A Haulsery, P Maurya, A Oliveira, A Pascoal, A Pereira, et al. Cooperative autonomous marine vehicle motion control in the scope of the eu grex project: theory and practice. In *OCEANS 2009-EUROPE*, pages 1–10. IEEE, 2009.
- [98] José Pinto, Paulo Sousa Dias, João Borges Sousa, and Fernando L Pereira. Large scale data collection using networks of heterogeneous vehicles and sensors. In *OCEANS 2009-EUROPE*, pages 1–6. IEEE, 2009.
- [99] Naomi Ehrich Leonard and Edward Fiorelli. Virtual leaders, artificial potentials and coordinated control of groups. In *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, volume 3, pages 2968–2973. IEEE, 2001.
- [100] Edward Fiorelli, Naomi Ehrich Leonard, Pradeep Bhatta, Derek A Paley, Ralf Bachmayer, and David M Fratantoni. Multi-auv control and adaptive sampling in monterey bay. *Oceanic Engineering, IEEE Journal of*, 31(4):935–948, 2006.
- [101] Petter Ogren, Edward Fiorelli, and Naomi Ehrich Leonard. Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment. *Automatic Control, IEEE Transactions on*, 49(8):1292–1302, 2004.
- [102] A. Alvarez, A. Caffaz, (...) Caiti, A., A. Turetta, and R. Viviani. Fòlaga: A low-cost autonomous underwater vehicle combining glider and auv capabilities. *Ocean Engineering*, 36:24–38, 2009.
- [103] Namik Kemal Yilmaz, Constantinos Evangelinos, Pierre FJ Lermusiaux, and Nicholas M Patrikalakis. Path planning of autonomous underwater vehicles for adaptive sampling using mixed integer linear programming. *Oceanic Engineering, IEEE Journal of*, 33(4):522–537, 2008.
- [104] Kevin P Carroll, Stephen R McClaran, Eric L Nelson, David M Barnett, Donald K Friesen, and GN William. Auv path planning: an a\* approach to path planning with consideration of variable vehicle speeds and multiple, overlapping, time-dependent exclusion zones. In *Autonomous Underwater Vehicle Technology, 1992. AUV'92., Proceedings of the 1992 Symposium on*, pages 79–84. IEEE, 1992.
- [105] R Smith, Arvind Pereira, Yi Chao, Peggy P Li, D Caron, B Jones, and G Sukhatme. Autonomous underwater vehicle trajectory design coupled with predictive ocean models: A case study. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 4770–4777. IEEE, 2010.
- [106] Donald P Eickstedt, Michael R Benjamin, and J Curcio. Behavior based adaptive control for autonomous oceanographic sampling. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 4245–4250. IEEE, 2007.
- [107] Petr Svec, Brual C Shah, Ivan R Bertaska, Jose Alvarez, Armando J Sinisterra, Karl Von Ellenrieder, Manhar Dhanak, and Satyandra K Gupta. Dynamics-aware target following for an autonomous surface vehicle operating under colregs in civilian traffic. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 3871–3878. IEEE, 2013.

- [108] Craig A Woolsey. Review of marine control systems: Guidance, navigation, and control of ships, rigs and underwater vehicles. *Journal of Guidance, Control, and Dynamics*, 28(3):574–575, 2005.
- [109] Alexander Bahr, John J Leonard, and Maurice F Fallon. Cooperative localization for autonomous underwater vehicles. *The International Journal of Robotics Research*, 28(6):714–728, 2009.
- [110] James C Kinsey, Ryan M Eustice, and Louis L Whitcomb. A survey of underwater vehicle navigation: Recent advances and new challenges. In *IFAC Conference of Manoeuvering and Control of Marine Craft*, 2006.
- [111] Luke Stutters, Honghai Liu, Carl Tiltman, and David J Brown. Navigation technologies for autonomous underwater vehicles. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 38(4):581–589, 2008.
- [112] Zuheir Altamimi, Patrick Sillard, and Claude Boucher. Itrf2000: A new release of the international terrestrial reference frame for earth science applications. *Journal of Geophysical Research: Solid Earth (1978–2012)*, 107(B10):ETG–2, 2002.
- [113] Neil Harvey Kussat, C David Chadwell, and Richard Zimmerman. Absolute positioning of an autonomous underwater vehicle using gps and acoustic measurements. *Oceanic Engineering, IEEE Journal of*, 30(1):153–164, 2005.
- [114] HR Everett. *Sensors for mobile robots: theory and application*. AK Peters, Ltd., 1995.
- [115] Oliver J Woodman. An introduction to inertial navigation. *University of Cambridge, Computer Laboratory, Tech. Rep. UCAMCL-TR-696*, 14:15, 2007.
- [116] Stephen D McPhail and Miles Pebody. Navigation and control of an autonomous underwater vehicle using a distributed, networked, control architecture. *Underwater Technology*, 23(1):19–30, 1998.
- [117] Nuno Ricardo Gracias, Sjoerd Van Der Zwaan, Alexandre Bernardino, and José Santos-Victor. Mosaic-based navigation for autonomous underwater vehicles. *Oceanic Engineering, IEEE Journal of*, 28(4):609–624, 2003.
- [118] Nuno Gracias and José Santos-Victor. Underwater video mosaics as visual navigation maps. *Computer Vision and Image Understanding*, 79(1):66–91, 2000.
- [119] A.D. Waite. *Sonar for practising engineers*. Wiley, 2002. URL: <http://books.google.pt/books?id=GXgZAQAAIAAJ>.
- [120] Lee Freitag, Matthew Grund, Sandipa Singh, James Partan, Peter Koski, and Keenan Ball. The whoi micro-modem: an acoustic communications and navigation system for multiple platforms. In *OCEANS, 2005. Proceedings of MTS/IEEE*, pages 1086–1092. IEEE, 2005.
- [121] Ryan M Eustice, Louis L Whitcomb, Hanumant Singh, and Matthew Grund. Experimental results in synchronous-clock one-way-travel-time acoustic navigation for autonomous underwater vehicles. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 4257–4264. IEEE, 2007.

- [122] A Pedro Aguiar and Joao P Hespanha. Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty. *Automatic Control, IEEE Transactions on*, 52(8):1362–1379, 2007.
- [123] X Yun, ER Bachmann, RB McGhee, RH Whalen, RL Roberts, RG Knapp, AJ Healey, and MJ Zyda. Testing and evaluation of an integrated gps/ins system for small auv navigation. *Oceanic Engineering, IEEE Journal of*, 24(3):396–404, 1999.
- [124] PK Maurya, AP Agular, and António M Pascoal. Marine vehicle path following using inner-outer loop control. *8th IFAC International Conference on Manoeuvring and Control of Marine Craft 2009, September 16-18, 2009, Guaruja-Brazil. 6 pp.*, 2009.