2023

# A path planning framework for multi-agent robotic systems based on multivariate skew-normal distributions

Peter Estephan
peter.estephan@live.com

# A PATH PLANNING FRAMEWORK FOR MULTI-AGENT ROBOTIC SYSTEMS BASED ON MULTIVARIATE SKEW-NORMAL DISTRIBUTIONS

A thesis submitted to
Marshall University
in partial fulfillment of
the requirements for the degree of
Master of Science
in
Electrical and Computer Engineering
by
Peter Estephan
Approved by
Pingping Zhu, PhD, Committee Chairperson
Sanghoon Lee, PhD
Mohammed Ferdjallah, PhD

Marshall University
May 2023

## APPROVAL OF THESIS

We, the faculty supervising the work of Peter Estephan, affirm that the thesis, *A Path Planning Framework for Multi-Agent Robotic Systems Based on Multivariate Skew-Normal Distributions*, meets the high academic standards for original scholarship and creative work established by the Department of Computer Sciences and Electrical Engineering and the College of Engineering and Computer Sciences. This work also conforms to the formatting guidelines of Marshall University. With our signatures, we approve the manuscript for publication.

| | | |
|---|---|---|
| | Pingping Zhu | 4/16/2023 |
| Pingping Zhu, PhD, Department of Computer Sciences and Electrical Engineering | Committee Chairperson | Date |
| | | 4/15/2023 |
| Sanghoon Lee, PhD, Department of Computer Sciences and Electrical Engineering | Committee Member | Date |
| | Mohammed Ferdjallah | 4/16/2023 |
| Mohammed Ferdjallah, PhD, Department of Computer Sciences and Electrical Engineering | Committee Member | Date |

## ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor Prof. Pingping Zhu for the continuous support of my study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing this thesis. Besides my advisor, I would like to thank my thesis committee Professors, Prof. Sanghoon Lee, Prof. Mohammed Ferdjallah and the department Chair Prof. Paulus Wahjudi.

## TABLE OF CONTENTS

# LIST OF TABLES

## LIST OF FIGURES

## LIST OF ACRONYMS

| Acronym | Description |
|---|---|
| APF | Artificial Potential Field |
| BHF | Bernoulli Hilbert Field |
| BRF | Bernoulli Random Field |
| BRF-SN | Bernoulli Random Field Skew-Normal |
| BRF-SNMM | Bernoulli Random Field Skew-Noemal Mixture Model |
| CDF | Continuous Density Function |
| CSD | Cauchy-Schwartz Divergence |
| DI | Displacement Interpolation |
| DOC | Distributed Optimal Control |
| EM | Expectation Maximization |
| FUSN | Fundamental Skew-Normal |
| FUSNMM | Fundamental Skew-Normal Mixture Model |
| GMM | Gaussian Mixture Model |
| MARS | Multi-Agent Robotic System |
| MLE | Maximum Likelihood Estimate |
| MVSN | Multivariate Skew-Normal |
| NLL | Negative Log Likelihood |
| PDF | Probability Density Function |
| ROI | Region of Interest |
| SN | Skew-Normal |
| SNMM | Skew-Normal Mixture Model |
| VLSR | Very Large Scale Robotics |

# ABSTRACT

Abstract-This thesis presents a path planning framework for a very-large-scale robotic (VLSR) system in an known obstacle environment, where the time-varying distributions of agents are applied to represent the multi-agent robotic system (MARS). A novel family of the multivariate skew-normal (MVSN) distributions is proposed based on the Bernoulli random field (BRF) referred to as the Bernoulli-random-field based skew-normal (BRF-SN) distribution. The proposed distributions are applied to model the agents' distributions in an obstacle-deployed environment, where the obstacle effect is represented by a skew function and separated from the no-obstacle agents' distributions. First, the obstacle layout is represented by a Hilbert occupancy classification, which can be modeled by a Bernoulli random field and approximated from observation data. Then, we construct the BRF-SN distributions from the approximated BRF and the agents' no-obstacle distributions, which can be modeled by a parametric distribution, e.g., the multivariate normal distributions or Gaussian mixture distributions. To learn unknown parameters from given samples, an Expectation-Maximization (EM) approach was used to minimize the upper bound of the negative log likelihood (NLL) function. Finally, to implement a time-varying path planning framework, two path-planning method are applied, an artificial potential field (APF) method for the macroscopic distribution trajectory was proposed based on $\ell_2$ norm function and Cauchy-Schwarz Divergence, also a displacement interpolation (DI) method is applied for the BRF-SN distribution.

This thesis involves no human subjects, surveys, or interviews, proposed framework performance is demonstrated in a simulated virtual environment.

Index Terms – Skew normal mixture model, Bernoulli-random-field based skew-normal, path-planning, multi-agent robotic system, expectation-maximization, artificial potential field.

# CHAPTER 1

# INTRODUCTION

The advancement of intelligent controls has revolutionized the way we interact with robotics systems, particularly for very-large-scale robotic (VLSR) applications. With the help of artificial intelligence and machine learning algorithms, these systems are capable of performing complex tasks with unprecedented speed and accuracy. This has led to the development of sophisticated control systems that enable these robots to operate autonomously. These advancements have opened up a range of possibilities for industries that rely on VLSR, including search and rescue applications [1][2]. To construct and control a VLSR system or robotic swarms, e.g., guiding or tracking robotic swarms in heavy obstacle deployed environments, a family of intelligent control approaches is proposed based on the time-varying robots or agents' spatial distributions, for example, the distributed optimal control (DOC) based approaches [3], [4], [5], [6]. In these approaches, the time-varying probability density functions (PDFs) of the robots/agents in the environments are generated as the macroscopic control of the agents trajectory, which are controlled to guide the robots/agents' microscopic control. in this thesis we propose novel statistical models to represent agents' distributions based on the multivariate skew-normal (MVSN) distribution. The formal definition of the univariate skew-normal family was provided by Azzalini in 1985 [11]. After that, many different definitions of multivariate skew-normal distributions were proposed [9], [10], [12], [13] and [14]. The key common of these skew-normal distributions is that these PDFs are all proportional to the product between the Gaussian PDF and a skewing function, where the skewing function is a probability function or a cumulative density function (CDF). In applying a skewing function to the Gaussian distribution, the interactions between the agents' distributions and obstacles is considered, a novel skew-normal (SN) distribution intoduced to model the agents' macroscopic distributions. In addition, to allow felixibility in representing the probabilistic models, skew-normal mixture models (SNMM) are introduced. Then, a parameter learning algorithm is developed to estimate unknown parameters of the SNMM from the agents' positions. The simulation results shows that the proposed algorithm can estimate the parameters of the SNMM, and the SNMM can provide

better macroscopic representation of agents in cluttered obstacle-deployed environments.Two path-planning algorithms are developed to generate the trajectories of the agents probability density functions (PDFs). To demonstrate the applications of the SNMM in the VLSR systems problem, two SNMM-based path planning algorithms are developed to guide a group of agents to traverse an artificial forest environment. The simulations also prove the effectiveness of these two path planning algorithms.

# CHAPTER 2

# PROBLEM FORMULATION

The Gaussian mixture model (GMM) is widely utilized to model the macroscopic states because of its fewer parameters and rich representation ability. However, in this thesis, we expose the GMM's limitations and disadvantages when the interactions between agents and obstacles matters significantly. Specifically, for high-density obstacles cluttered environments where the individual obstacle's scale is relatively small but the number of obstacles is relatively large, e.g., the forest [5], it is improper to specify the agents' distribution by GMMs. There are no clear and simple boundaries between agents and obstacles. Where applying GMM principles, will create a burden of computational complexity, as the system requires a very large number of components to successfully navigate the workspace area. It can be seen that the GMM's limitation results from its components, Gaussian or normal distributions. As one of the most popular and simple continuous distributions, the normal distributions have been applied widely to describe data statistical properties in scientific and engineering areas. But normal distribution's mathematical simplicity limits its flexibility and adaptivity. To describe the agents' distributions more properly to interact with complex environments, consider the problem of path planning trajectory of a VLSR system compromised of $N$-cooperative agents, deployed according to a known normal distribution. All agents share the task of navigating through the region of interest (ROI), compromised of a high-density obstacle workspace $\mathscr{W}$ such that $\mathscr{W} \in \mathbb{R}^2$. The agents' positions are denoted by $\mathbf{x} \in \mathbf{W}$, and are modeled as multivariate random variables $\mathbf{X}^*$. The random variables are defined as skew-normal distributions, such that $\mathbf{X}^* \sim FUSN(\mu, \Sigma, Q)$ generated from a normal distribution $\mathbf{X} \sim \mathscr{N}(\mu, \Sigma)$, here $Q$ is interpreted as a skewing function, $\mu$ and $\Sigma$ are the mean and covariance, respectively. On the other hand, let $\mathscr{B} \in \mathscr{W}$, denote the set of locations occupied by obstacles in the workspace. In a setup, where the region of interest (ROI) is a high-density obstacle workspace-deployed area, adopting a GMM technique will be computationally expensive to accurately model the data, as the increased number of components introduces a challenging factor in capturing complex distributions.

# CHAPTER 3

# BACKGROUND

## 3.1 GAUSSIAN DISTRIBUTION

The normal distribution is a type of probability distribution that describes the likelihood of obtaining a particular set of two-dimensional data. It is also known as a bivariate normal distribution.

### 3.1.1 NORMAL DISTRIBUTION

In a 2D environment, data is represented as a set of pairs of two-dimensional coordinates, often denoted as (x, y). A normal distribution in this context is characterized by two parameters: the mean vector $\mu$, which represents the central tendency of the data, and the covariance matrix $\Sigma$, which describes the spread and correlation between the x and y values. The shape of a normal distribution in a 2D environment is elliptical, with the contours of the distribution oriented along the direction of the covariance between x and y. The higher the covariance between the x and y values, the more elongated the elliptical shape of the distribution. When the covariance between x and y is zero, the normal distribution in the 2D environment reduces to a circular shape. The normal distribution in a 2D environment is widely used in statistical analysis and modeling of bivariate data. It allows researchers to describe the relationship between two variables and make predictions about future observations.

A multivariate normal distribution is a generalization of the univariate normal distribution to higher dimension $d$, which can mathematically be denoted as $\mathcal{N}(\mu, \Sigma) \subseteq \mathbb{R}^d$ where $\mu \in \mathbb{R}^d$ is a column vector of means and $\Sigma \in \mathbb{R}^{d \times d}$ is a positive definite covariance matrix. The probability density function (PDF) of the Gaussian distribution can be expressed as:

$$f(\mathbf{x}) = 2\pi^{-d/2} \det{(\Sigma)}^{-1/2} e^{-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1} (\mathbf{x}-\mu)} \tag{3.1}$$

the Gaussian distribution PDF can be representated in terms of $\mu$ and $\Sigma$ as

$$f_{\mathbf{X}}(\mathbf{x}) \triangleq \phi_{\mathbf{X}}(\mathbf{x}|\mu, \Sigma) \tag{3.2}$$

Here the random variable $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2]$, represents the dimension of components within the distribution such that $\mathbf{X} \sim \mathcal{N}(\mu, \Sigma)$.

### 3.1.2 GAUSSIAN MIXTURE MODEL

Gaussian Mixture Models (GMM) are widely used and flexible probabilistic models for representing complex data distributions. The basic idea behind GMMs is to model the data as a mixture of multiple Gaussian distributions [10], each with its own mean, covariance matrix, and mixing coefficient. These models are particularly useful in situations where the data distribution is multimodal, meaning that it has multiple distinct clusters or modes. By modeling each mode as a separate Gaussian Distribution, GMMs allows for the representation of complex, multi-peaked distributions that would be difficult to model using a single Gaussian distribution. GMM is a type of generative model, which means that it can be used to generate new data points from the learned distribution. To do this, one first samples a Gaussian component from the mixture according to the mixing coefficients, and then generates a new data point from the chosen Gaussian. This allows for the generation of synthetic data that resembles the original data distribution, which can be useful for data augmentation and other applications. The probability density function (PDF) of the Gaussian Mixture Model (GMM) can be expressed as the weighted sum of $N_c$ Gaussian distributions:

$$f_{\mathbf{X}}(\mathbf{x}) = \sum_{i=1}^{N_c} \boldsymbol{\omega}_i \phi(\mathbf{x}|\mu_i, \Sigma_i) \tag{3.3}$$

where $\mu_i$ and $\Sigma_i$ are the mean vector and the covariance matrix for the $i$th Gaussian component, respectively. The GMM PDF represents the probability density of observing a data point $\mathbf{x}$ given the GMM model parameters. The GMM model parameters can be estimated using the Expectation-Maximization (EM) algorithm, which iteratively updates the mean vectors, covariance matrices, and mixing coefficients to maximize the likelihood of the observed data given the model.

## 3.2 SKEW NORMAL DISTRIBUTION

Skew Normal Distribution [12][14] is a probability distribution that is a generalization of the Normal Distribution but with an additional skewness parameter. It is often used in statistical modeling to capture data with non-symmetric and skewed distributions. The probability density function (PDF) of the Skew Normal Distribution is given by:

$$f(\mathbf{x}) = 2\phi(\mathbf{x})\Phi(\lambda\mathbf{x}) \tag{3.4}$$

where $\phi(\mathbf{x})$ is the PDF of the Gaussian distribution defined in (3.2) and $\Phi(\mathbf{x})$ is the cumulative distribution function (CDF), while $\lambda$ is the skewness parameter, the CDF of the distribution is defined as follow

$$\Phi(\lambda\mathbf{x}) \triangleq F(x) = \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{\lambda(x-\mu)}{\sigma\sqrt{2}}\right)\right] \tag{3.5}$$

the function $\operatorname{erf}(z) = \frac{2}{\sqrt{\pi}}\int_0^z e^{-t^2}dt$ is the error function for the continuous probability distribution. The skew normal distribution is a generalization of the normal distribution that allows for skewness (as indicated by the parameter $\lambda$). If $\lambda = 0$, the skew normal distribution reduces to the normal distribution with mean $\mu$ and standard deviation $\sigma$. When $\lambda > 0$, the distribution is skewed to the right, and when $\lambda < 0$, the distribution is skewed to the left, where the distribution is often skewed due to the presence of outliers and extreme events. According to [12], many different families of skew-normal distributions can be contained.

### 3.2.1 FUNDAMENTAL SKEW NORMAL DISTRIBUTION

The Fundamental Skew Normal Distribution (FUSN) is a generalization of the Skew Normal Distribution (SN) that allows for more flexible skewness patterns and can model both positive and negative skewness [12]. A FUSN is defined and a modified version is provided as follows: Consider two random vectors, $\mathbf{X} \in \mathbb{R}^n$ and $\mathbf{Z} \in \mathbb{R}^m$, where the random vector $\mathbf{X}$ is multivariate normal distribution, such that $\mathbf{X} \in \mathcal{N}(\mu, \Sigma)$. The random variable, $\mathbf{X}^* = [\mathbf{X}|\mathbf{X} < \xi]$, has a fundamental skew-normal (FUSN) distribution denoted by $\mathbf{X}^* \sim FUSN_{n,m}(\mu, \Sigma, Q_{\mathbf{Z}})$, with a probability density function (PDF) is given by:

$$f_{\mathbf{X}^*}(\mathbf{x}) = K_{\mathbf{Z}}^{-1}\phi_{\mathbf{X}}(\mathbf{x}|\mu, \Sigma)Q_{\mathbf{Z}}(\mathbf{x}) \tag{3.6}$$

where $\phi_{\mathbf{x}}(\mathbf{x}|\mu, \Sigma)$ is the PDF of the multivariate normal random vector $\mathbf{X}$,

$$Q_{\mathbf{Z}}(\mathbf{x}) \triangleq P(\mathbf{Z} < \xi|\mathbf{X} = \mathbf{x}) \tag{3.7}$$

and

$$K_{\mathbf{Z}} \triangleq P(\mathbf{Z} < \xi) \tag{3.8}$$

The term $K_{\mathbf{z}}$ is a normalizing constant defined in (3.8) as the CDF of $\mathbf{Z}$, $\xi$ is a probability setpoint such that $0 \leq \xi \leq 1$, here $Q(\mathbf{x})$ can be interpreted as a skewing function. According to (3.7), the skewing function can be expressed by

$$\begin{aligned} Q_{\mathbf{Z}}(\mathbf{x}) &= P(\mathbf{Z} < \xi|\mathbf{X} = \mathbf{x}) \\ &= \int_{-\infty}^{\xi} \phi_{\mathbf{Z}}(\mathbf{z}|\mathbf{D}\mathbf{x} + \nu, \Omega)d\mathbf{z} \\ &= \int_{-\infty}^{\xi-(\mathbf{D}\mathbf{x}+\nu)} \phi_{\mathbf{Z}}(\mathbf{z}|0, \Omega)d\mathbf{z} \\ &= \Phi_{\mathbf{Z}}\left(\xi - (\mathbf{D}\mathbf{x} + \nu)|0, \Omega\right) \end{aligned} \tag{3.9}$$

where $\phi_{\mathbf{Z}}$ and $\Phi_{\mathbf{Z}}$ indicate the PDF and CDF of the multivariate normal random vector $\mathbf{Z}$, respectively. Besides, the normalizing constant $K_{\mathbf{Z}}$ can be expressed by

$$K_{\mathbf{Z}} = P(\mathbf{Z} < \xi) = \Phi_{\mathbf{Z}}(\xi|\mathbf{D}\mu + \nu, \mathbf{D}\Sigma\mathbf{D}^T + \Omega) \tag{3.10}$$

Therefore, by substituting (3.9) and (3.10) in (3.8) we obtain a FUSN associated with the following PDF

$$f_{\mathbf{X}^*}(\mathbf{x}) = \frac{\phi_{\mathbf{X}}(\mathbf{x}|\mu, \Sigma)\Phi_{\mathbf{Z}}(\xi - (\mathbf{D}\mathbf{x}|0, \Omega))}{\Phi_{\mathbf{Z}}(\xi|\mathbf{D}\mu + \nu, \mathbf{D}\Sigma\mathbf{D} + \Omega))} \tag{3.11}$$

If assuming that $\mu = 0, \nu = 0,\ \xi = 0,\ \Sigma = \mathbf{I}_p,\ \Omega = \mathbf{I}_q,$ and $\mathbf{D} = -\Lambda$, where $\mathbf{I}_p$ and $\mathbf{I}_q$ are identity matrices, respectively, we can obtain the following PDF

$$f_{\mathbf{X}^*}(\mathbf{x}) = \frac{\phi_{\mathbf{X}}(\mathbf{x}|0, \mathbf{I}_n)\Phi_{\mathbf{Z}}(\Lambda\mathbf{x}|0, \mathbf{I}_p)}{\Phi_{\mathbf{Z}}(0|0, \Lambda\Lambda^T + \mathbf{I}_q)} \tag{3.12}$$

which is equivalent to the multivariate skew normal distribution considered in (3.6) and (3.7). Furthermore, considering the case of $m = 1$ and $\Lambda = \lambda^T$, where $\lambda \in \mathbb{E}$ is a column vector, we can have

$$f_{\mathbf{X}^*}(\mathbf{x}) = 2\phi_{\mathbf{X}}(\mathbf{x}|0, \mathbf{I}_q)\Phi_{\mathbf{Z}}(\lambda^T\mathbf{x}|0, 1) \tag{3.13}$$

which is equivalent to the SN in (3.3), by considering $p = q = 1$, the FUSN PDF can be expressed by

$$f_{\mathbf{X}^*}(\mathbf{x}) = 2\phi_{\mathbf{X}}(\mathbf{x}|0, 1)\Phi_{\mathbf{Z}(\lambda\mathbf{x}|0,1)} \tag{3.14}$$

Therefore, the SN is a special case of the FUSN, where the skewness is restricted to a linear function of the data. In contrast, the FUSN allows for more complex and non-linear skewness patterns, making it more flexible and applicable to a wider range of data distributions. The FUSN has many applications in statistical modeling and in image processing, where it has been used to model the distribution with complex skewness patterns.

### 3.2.2 SKEW NORMAL MIXTURE MODEL

The Fundamental Skew Normal Mixture Model (FUSNMM) is an extension of the Fundamental Skew Normal Distribution (FUSN), which models a mixture of FUSN distributions [13]. The FUSNMM can capture complex patterns of skewness and multimodality in the data, making it a powerful tool for modeling various types of data distributions. The probability density function (PDF) of the FSNMM is given by:

$$f(\mathbf{x}) = \sum_{i=1}^{N_c} \omega_i f_i(\mathbf{x}), \tag{3.15}$$

where $\omega_i$ is the mixing weight for the $N_c$th component such that $\sum_{i=1}^{K} \omega_i = 1$, and $f_i(x)$ is the PDF of the $i$th component, which is a FUSN distribution with parameters $(\mu_i, \sigma_i, \lambda_i)$. The parameters of the FUSNMM can be estimated using the Expectation-Maximization (EM) algorithm or other optimization methods. The FSNMM has many applications in statistical modeling, such as in finance, where it can be used to model financial returns with complex patterns of skewness and multimodality. It can also be used in image processing and computer vision, where it has been used to model the distribution of image intensities with multiple modes and varying degrees of skewness. The FSNMM is a more flexible model than the FUSN, as it allows for the modeling of more complex data distributions with multiple modes and varying degrees of skewness. The FUSNMM can be viewed as a generalization of the FUSN, where the FUSN is a special case of the FUSNMM with only one component. The FUSNMM can be used to model a wide range of data distributions, making it a powerful tool for statistical modeling and analysis.

## 3.3 BERNOULLI RANDOM FIELD BASED HILBERT OCCUPANCY

Bernoulli random field-based Hilbert occupancy (BHF) is a method of modeling the occupancy of an environment using a Hilbert curve and a Bernoulli random field. In this method, the environment is divided into a grid of cells, and each cell is assigned a binary value to indicate whether it is occupied (1) or free (0). The goal is to estimate the occupancy status of each cell based on measurements of the environment. The BHF method uses a Hilbert curve to traverse the grid of cells in a space-filling order. The Hilbert curve provides a way to visit all the cells in a particular order that preserves spatial locality. This property is useful for modeling environments where adjacent cells are likely to have similar occupancy statuses.

### 3.3.1 HILBERT OCCUPANCY CLASSIFICATION

Hilbert occupancy classification is a way to represent an environment using Hilbert curve [17], the occupancy represents the status (occupied/free) of a continuous function over a workspace $\mathscr{W} \in \mathbb{R}^d$. The Hilbert map represents a probability map learned from local ordering data points on $\mathscr{W}$ [16]. Let $\mathbf{x} \in \mathscr{W}$ be any point in $\mathscr{W}$ and $Y \in \{0, 1\}$ be defined as a categorical random variable such that the Hilbert occupancy classification map at the position $\mathbf{x}$ is defined as

$$\begin{aligned} P(Y=1|\mathbf{x}) &= p(\mathbf{x}) \\ P(Y=0|\mathbf{x}) &= 1-p(\mathbf{x}) \end{aligned} \tag{3.16}$$

considering the data sample $\mathscr{D} = \{\mathbf{x}_i\}_{n=1}^{N}$, here $\mathbf{x} \in \mathscr{W}$ represents auxiliary data points on the workspace with the resolution $\Delta x$ and $\Delta y$, while $n = 1, ..., N$ is the environment auxiliary data points collected. The probability $P(Y|\mathbf{x})$ can be updated in terms of the Hilbert function defined on $\mathscr{W}$ as follows

$$\begin{aligned} P(Y=1|\mathbf{x}) &= p(\mathbf{x}) = \frac{e^{f(\mathbf{x})}}{1+e^{f(\mathbf{x})}} \\ P(Y=0|\mathbf{x}) &= 1-p(\mathbf{x}) = \frac{1}{1+e^{f(\mathbf{x})}} \end{aligned} \tag{3.17}$$

### 3.3.2 BERNOULLI RANDOM FIELD

A Bernoulli random field is a type of spatial stochastic process that models binary-valued data [20] (i.e., data that can take only two values, such as presence or absence). In a Bernoulli random field, each point is associated with a binary variable that takes a value of 0 or 1, depending on whether the point is in a particular state or not. A Bernoulli random field is characterized by a probability distribution that specifies the probability of observing a particular configuration of binary variables in the field. The probability of a configuration is determined by a set of parameters that capture the spatial dependence structure of the field. Let $\mathscr{B} \in \mathscr{W}$ denote the set of the locations occupied in the workspace, and let $Y(\mathbf{x})$ a Bernoulli random variable such that

$$\begin{aligned} Y : \mathscr{W} &\to \{0,1\} \\ \mathbf{x} \mapsto Y(\mathbf{x}) &= \begin{cases} 0 & \text{if } \mathbf{x} \notin \mathscr{B} \\ 1 & \text{if } \mathbf{x} \in \mathscr{B} \end{cases} \end{aligned} \tag{3.18}$$

then, we obtain the random field $\mathscr{Y}$ defined by

$$\mathscr{Y} \triangleq \{Y(\mathbf{x}) : \mathbf{x} \in \mathscr{W}\} \tag{3.19}$$

## 3.4 PARAMETER LEARNING

Parameter learning is a process of estimating the parameters of a statistical or machine learning model from data. The parameters of a model are values that determine the behavior of the model and are typically unknown a priori. Parameter learning involves finding the values of these parameters that best fit the data, so that the model can be used to make accurate predictions or inferences. There are several methods for parameter learning, depending on the type of model and the available data. Some common methods include:

- Maximum Likelihood Estimation (MLE) [25] is a statistical method used to estimate the parameters of a probability distribution that best explains a given set of observations or data. The goal of MLE is to find the values of the parameters that maximize the likelihood of observing the data, assuming a certain probability distribution.

- Gradient descent [16] is an optimization algorithm used to find the minimum of a function by iteratively adjusting the parameters of the function. The basic idea behind gradient descent is to start with an initial guess for the parameter values and then move iteratively in the direction of the negative gradient of the function with respect to those parameters. This direction is known as the steepest descent, and the size of each step is determined by a learning rate.

- Expectation-Maximization (EM) method [28] is explained in ($textbfsection$3.4.1.

### 3.4.1 EXPECTATION-MAXIMIZATION METHOD

The Expectation-Maximization (EM) algorithm is a powerful and widely used method for estimating the parameters of statistical models that involve latent variables, i.e., variables that are not observed in the data [26]. The EM algorithm is an iterative procedure that alternates between two steps: the E-step and the M-step. In the E-step, the algorithm estimates the values of the latent variables given the current estimates of the model parameters. This is done using Bayes' rule, which computes the conditional probability distribution of the latent variables given the observed data and the current estimates of the parameters. The E-step results in a set of expected values of the latent variables, hence the name ”Expectation” step. In the M-step, the algorithm updates the estimates of the model parameters based on the expected values of the latent variables obtained in the E-step. This is done by maximizing a likelihood function that incorporates the expected values of the latent variables. The M-step results in a new set of parameter estimates, hence the name ”Maximization” step. The EM algorithm iterates between the E-step and the M-step until convergence is reached, i.e., until the change in the estimated parameters between successive iterations falls below a predefined threshold. The EM algorithm is particularly useful in situations where some of the variables are unobserved or missing, and need to be estimated from the available data.

## 3.5 PATH-PLANNING BASED ARTIFICIAL POTENTIAL FIELD

Artificial Potential Field (APF) is a widely used approach for path planning in robotics and autonomous systems [27]. The basic idea of APF is to model the environment as a potential field, where each point in the space is assigned a scalar value that represents its potential. The robot is then guided to move along the gradient of the potential field, which corresponds to the direction of decreasing potential. In this way, the robot can avoid obstacles and reach its goal while minimizing its energy consumption. In an APF-based path planning system, the environment is typically represented by two types of potential fields: the attractive potential field and the repulsive potential field. The attractive potential field is centered at the goal location and encourages the robot to move towards it. The repulsive potential field, on the other hand, is centered on obstacles and repels the robot away from them. The total potential field is computed as a sum of the attractive and repulsive potentials, and the robot is guided to move along the negative gradient of the total potential field. The resulting path is a smooth and continuous trajectory that avoids obstacles and reaches the goal, such that

$$U^{total} = \sum_{n=1}^{N} U_n^{attractive} + U_n^{repulsive} \tag{3.20}$$

where $n = 1, ..., N$ denotes the total number of robots or agents, $U^{attractive}$ and $U^{repulsive}$ represent the attractive and repulsive potentials, respectively.

### 3.5.1 L-2 NORM FUNCTION

The L2-norm function, also known as the Euclidean norm or the $\ell 2$-norm, is a mathematical function that measures the length or magnitude of a vector in Euclidean space [21]. The L2-norm is defined as the square root of the sum of the squares of the vector's components. Mathematically, the L2-norm of a vector $\mathbf{x}_n = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N]$ is given by:

$$\begin{aligned} ||\mathbf{x}|| &= \sqrt{(\mathbf{x}_1^2 + \mathbf{x}_2^2 + ... + \mathbf{x}_N^2)} \\ &= \sqrt{\sum_{n=1}^{N} \mathbf{x}_n^2} \end{aligned} \tag{3.21}$$

Geometrically, the L2-norm of a vector represents the distance of the vector from the origin of the coordinate system to its endpoint, which is also known as the vector's magnitude or length. The L2-norm is commonly used in various applications, such as machine learning, signal processing, and optimization, where it is used to measure the similarity or distance between two vectors or to regularize the objective function in optimization problems. One useful property of the L2-norm is that it satisfies the triangle inequality, which means that the length of the sum of two vectors is always less than or equal to the sum of their lengths. Another useful property is that it is differentiable with respect to its components, which makes it convenient for optimization algorithms that require gradient-based optimization. The L2-norm is just one of many different norm functions that can be used to measure the length or distance of a vector.

### 3.5.2 CAUCHY-SCHWARZ DIVERGENCE

The Cauchy-Schwarz Divergence (CSD) is a measure of the dissimilarity or distance between two probability distributions [21]. It is based on the Cauchy-Schwarz inequality , which is a fundamental inequality in mathematics that relates the inner product of two vectors to their norms. The CSD is often used in machine learning and information theory, where it is used to compare different models or to measure the quality of a generative model. Mathematically, the

CSD between two probability distributions $p$ and $q$ is defined as:

$$D(p||q) = -\log \frac{\int p(\mathbf{x})q(\mathbf{x})d\mathbf{x}}{\sqrt{\int p(\mathbf{x})d\mathbf{x} \int q(\mathbf{x})d\mathbf{x}}} \tag{3.22}$$

where $D(p||q)$ denotes the CSD between $p$ and $q$, and the integral is taken over the entire space of possible events or outcomes. The numerator of the equation represents the inner product between $p$ and $q$, and the denominator is the product of their L2-norms. The logarithm is used to convert the ratio into a distance measure. The CSD has several useful properties, such as being a symmetric and non-negative measure of divergence, and satisfying the data processing inequality, which means that the divergence between two distributions can only decrease or remain the same after a data processing step. The CSD can also be used as a building block for other divergence measures, such as the Total Variation (TV) distance [29], Kullback-Leibler divergence [30] and the Jensen-Shannon (JS) divergence [31] .

## 3.6 PATH-PLANNING BASED DISPLACEMENT INTERPOLATION

Path-planning based displacement interpolation is a technique used in computer graphics and animation to generate smooth and natural-looking motion paths for objects or characters. The goal of displacement interpolation is to generate a sequence of intermediate positions and orientations for an object or character, based on a given set of keyframes or control points. Path-planning based displacement interpolation approaches this problem by first generating a smooth path that connects the keyframes or control points. This path is typically represented as a sequence of positions or waypoints that the object or character must follow.

Once the path is generated, the next step is to interpolate the object or character's position and orientation along the path to generate a sequence of intermediate frames. This is done by computing the displacement vector between each pair of consecutive waypoints on the path, and then using this displacement vector to compute the object or character's position and orientation at each intermediate frame. The displacement vector can be computed using various techniques, such as linear interpolation, cubic spline interpolation, or other interpolation methods. It is commonly used in computer animation and game development to create realistic and fluid character animations. By using path-planning algorithms and interpolation techniques, displacement interpolation can generate smooth and natural-looking motion paths. More details are provide in **Chapter 5** sections **5.1.1** and **5.2.1**.

# CHAPTER 4

# MULTIVARIATE SKEW-NORMAL MIXTURE MODEL

A fundamental skew normal distribution (FUSN) was defined in (3.8), building off of that definition, a modified version of SN based Bernoulli Random Field distribution is introduced to better understand the SND behavior in a binary environment.

## 4.1 BERNOULLI RANDOM FIELD SKEW-NORMAL DISTRIBUTION

Consider the Bernoulli Random Field (BRF) defined in (3.18) and (3.19) where $\mathscr{Y} \triangleq \{Y(\mathbf{x}) : \mathbf{x} \in \mathbb{E}^d\}$, which is specified by a paramater function $q(x)$ defined by

$$q : \mathbb{R}^d \to [0, 1]$$

$$\mathbf{x} \mapsto \mathbb{E}[Y(\mathbf{x})] = P(Y(\mathbf{x} = 1)) \tag{4.1}$$

Then given a multivariate normal random variable, $\mathbf{X} \sim \mathscr{N}(\mu, \Sigma)$, and a shape parameter $\zeta \in [0, 1]$. The random variable $\mathbf{X}^* \triangleq [\mathbf{X}|\amalg < \zeta]$, here the random variable $\amalg \triangleq q(\mathbf{X})$ and $\mathbf{X}^*$ has a Bernoulli random field skew normal (BRF-SN) distribution denoted by $\mathbf{X}^* \sim BRFSN(\mu, \Sigma, q)$ with a given PDF

$$\begin{aligned} f_{\mathbf{X}^*}(\mathbf{x}) &= K_{\amalg}^{-1} \phi_{\mathbf{x}}(\mathbf{x}|\mu, \Sigma) Q_{\amalg}(\mathbf{x}) \\ &= \frac{\phi_{\mathbf{x}}(\mathbf{x}|\mu, \Sigma) Q_{\amalg}(\mathbf{x})}{\mathbb{E}_{\mathbf{X} \sim \phi_{\mathbf{X},i}}[Q_{\amalg}(\mathbf{X})]} \end{aligned} \tag{4.2}$$

where $Q_{\amalg}(\mathbf{x}) \triangleq P(\amalg < \zeta|\mathbf{X} = \mathbf{x})$ and $K_{\amalg} \triangleq P(\amalg < \zeta) = \mathbb{E}_{\mathbf{x}}[Q_{\amalg}(\mathbf{x})]$ Since $q$ is a deterministic binary function the range of $Q_{\amalg}$ is $\{0, 1\}$ such that

$$Q_{\amalg}(\mathbf{x}) = P(\amalg < \zeta|\mathbf{X} = \mathbf{x}) = \begin{cases} 0 & \text{if } q(\mathbf{x}) \geq \zeta \\ 1 & \text{if } q(\mathbf{x}) < \zeta \end{cases} \tag{4.3}$$

Thus, the BRFSN distribution can be interpreted as a truncated and normalized normal distribution using the skewing function $Q_{\amalg}$. To obtain a smooth skew function, referring to the

FUSN in section (3.2.1), we introduce an observation random vector $\mathbf{Z}(\mathbf{x}) \in \mathbb{R}^m$ in the statistical model of multi-agent exploration. Assume that the random vector $[\mathbf{Z}(\mathbf{x})|Y(\mathbf{x}) = y]$ is the multivariate normal distribution, and the corresponding PDF is specified by

$$\phi_{\mathbf{Z}|Y}(\mathbf{z}|y) = \begin{cases} \phi_{\mathbf{Z}}(\mathbf{z}|\nu_0, \Omega_0) & \text{if } y = 0 \\ \phi_{\mathbf{Z}}(\mathbf{z}|\nu_1, \Omega_1) & \text{if } y = 1 \end{cases} \tag{4.4}$$

here, $\nu_i \in \mathbb{R}^m$ and $\Sigma_i$ are mean and covariance matrices, respectively, for $i = 1, 2$, which are all determined by the obstacle observation model. The random vector $\mathbf{Z}(\mathbf{x})$ can be interpreted as a two-component Gaussian mixture model (GMM), and its PDF can be expressed by

$$f_{\mathbf{Z}}(\mathbf{z}) = \sum_{i=\{0,1\}} P(Y(\mathbf{x} = i)\phi_{\mathbf{Z}|Y}(\mathbf{z}|Y(\mathbf{x}) = i) \tag{4.5}$$

Thus, we obtain another random field, $\mathscr{Z}$, referred to as the Gaussian-mixture-random-field (GMRF), which is defined by

$$\mathscr{Z} \triangleq \{\mathbf{Z}(\mathbf{x}) : \mathbf{x} \in \mathscr{W}\} \tag{4.6}$$

as $\mathscr{Z}$ is specified by the parameter function $q$, the mean vectors $\nu_i$ and the covariance matrices $\Omega_i$, for $i = 1, 2$. Furthermore, consider the two-component random field defined in (4.6) with the PDF of the two-component Gaussian mixture random variable $\mathbf{Z}(\mathbf{x}) \in \mathbb{R}^m$ is provided by

$$f_{\mathbf{Z}}(\mathbf{z}) = (1 - \omega)\phi_{\mathbf{Z}}(\mathbf{z}|\nu_0, \Omega_0) + \pi\phi_{\mathbf{Z}}(\mathbf{z}|\nu_1, \Omega_1) \tag{4.7}$$

where, $\omega$ is the evaluation of the parameter function, such that $\pi = q(\mathbf{x}) \in [0, 1]$, and $\pi = [\pi \;\; (1 - \pi))^T$ indicated the weights of the Gaussian components. Then given the multivariate normal random vector, $\mathbf{X} \sim \mathscr{N}(\mu, \Sigma)$, and a parameter vector, $\zeta \in \mathbb{R}^m$, a random vector $\mathbf{X}^* \triangleq [\mathbf{X}|\mathbf{Z} < \zeta]$, can be constructed. The random variable $\mathbf{X}^*$ is a skew-normal (SN) distribution, with a PDF dentoted by

$$f_{\mathbf{X}^*}(\mathbf{x}) = K_{\mathbf{Z}}^{-1}\phi_{\mathbf{X}}(\mathbf{x}|\mu, \Sigma)Q_{\mathbf{Z}}(\mathbf{x}) \tag{4.8}$$

## 4.2 BERNOULLI RANDOM FIELD SKEW-NORMAL MIXTURE MODEL

To describe a more complex agents' distribution, rather than the normal distribution, we extend the BRF-SN (4.2) distribution to the Bernoulli random field skew-normal mixture model (BRF-SNMM) associated with the following PDF,

$$\begin{aligned} \varphi_{\mathbf{X}^*}(\mathbf{x}) &\triangleq \sum_{i=1}^{N_c} \omega_i f_{\mathbf{X}^*}(\mathbf{x}|\mu_i, \Sigma_i) \\ &= \sum_{i=1}^{N_c} \omega_i \frac{\phi_{\mathbf{x},i}(\mathbf{x}|\mu_i, \Sigma_i) Q_Y \mathbf{x})}{\mathbb{E}_{\mathbf{X} \sim \phi_{\mathbf{x},i}}[Q_Y(\mathbf{X})]} \end{aligned} \tag{4.9}$$

Which consists of $N_c$ BRF-SN components, $\{\phi_{\mathbf{x},i}\}_{i=1}^{N_c}$ defined in (4.1) associated with the corresponding weights, $\{\omega_i\}_{i=1}^{N_c}$, given the skewing function $Q_Y(\mathbf{x})$. The BRF-SNMM is specified by 2 components, the component parameters, $\Theta_i = (\mu_i, \Sigma_i)$, for $i = 1, ..., N_c$, and the weight parameters, $\omega = [\omega_1 ... \omega_{N_c}]$, such that $\sum_{i=1}^{N_c} \omega_i = 1$ and $\omega_i > 0$ for all $i$, which are also denoted for short as

$$\Theta_{1:N_c} = \{(\omega_i, \Theta_i)\}_{i=1}^{N_c} \tag{4.10}$$

## 4.3 PARAMETERS LEARNING FOR SKEW-NORMAL MIXTURE MODEL

Consider that there are $N$ samples, $\mathscr{D} = \{\mathbf{x}_n\}_{n=1}^{N}$, which are all independently sampled according to the BRF-SN distribution associated with the PDF $f_{\mathbf{X}^*}(\mathbf{x}|\mu, \Sigma)$. To specify the unknown underlying parameters, $\Theta = (\mu, \Sigma)$ from the samples, we can construct the following negative log-likelihood (NLL) cost function,

$$\begin{aligned} J(\Theta) &= -ln \prod_{n=1}^{N} f_{\mathbf{X}^*}(\mathbf{x}_n|\mu, \Sigma) \\ &= \sum_{n=1}^{N} \mathscr{L}(\mathbf{x}_n, \mu, \Sigma) \end{aligned} \tag{4.11}$$

where the term, $(\mathbf{x}_n|\mu_i, \Sigma_i)$ is defined by

$$\begin{aligned}
\mathscr{L}(\mathbf{x}_n|\mu, \Sigma) &\triangleq -ln f_{\mathbf{X}^*}(\mathbf{x}_i|\mu, \Sigma) \\
&= -ln\left[\frac{\phi_{\mathbf{X}_n}(\mathbf{x}|\mu_i, \Sigma_i)Q_{(}\mathbf{x}_n)}{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{X}}}[Q(\mathbf{X})]}\right] \\
&= -ln[\phi_{\mathbf{X}_n}(\mathbf{x}|\mu_i, \Sigma_i)] - ln[Q(\mathbf{x}_n)] + ln[\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{X}}} Q(\mathbf{X})] \\
&= -ln[\phi_{\mathbf{X}_n}(\mathbf{x}|\mu_i, \Sigma_i)] - ln[Q(\mathbf{x}_n)] + ln[\int \phi_{\mathbf{X}_n}(\mathbf{x}|\mu_i, \Sigma_i)Q(\mathbf{X})]
\end{aligned} \tag{4.12}$$

### 4.3.1 PARAMETER LEARNING FOR BRF-SN

In order to minimize the cost function, we compile the derivative to the parameters, $\mu$ and $\Sigma$, which can be expressed by

$$\frac{\partial J(\Theta)}{\partial \mu} = \sum_{n=1}^{N} \frac{\partial}{\partial \mu}\mathscr{L}(\mathbf{x}_n, \mu, \Sigma) \tag{4.13}$$

$$\frac{\partial J(\Theta)}{\partial \Sigma} = \sum_{n=1}^{N} \frac{\partial}{\partial \Sigma}\mathscr{L}(\mathbf{x}_n, \mu, \Sigma) \tag{4.14}$$

where

$$\frac{\partial}{\partial \mu}\mathscr{L}(\mathbf{x}_n, \mu, \Sigma) = \Sigma^{-1}\left[\frac{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{x}}}[\mathbf{X}Q(\mathbf{X})]}{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{x}}}[Q(\mathbf{X})]} - \mathbf{x}_i\right] \tag{4.15}$$

and

$$\frac{\partial}{\partial \Sigma}\mathscr{L}(\mathbf{x}_n, \mu, \Sigma) = \frac{1}{2}\Sigma^{-1}\left[\frac{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{x}}}[Q(\mathbf{X})(\mathbf{X}-\mu)(\mathbf{X}-\mu)^T]}{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{x}}}[Q(\mathbf{X})]} - (\mathbf{x}_n - \mu)(\mathbf{x}_n - \mu)^T\right]\Sigma^{-1} \tag{4.16}$$

The derivation of (4.15) and (4.16) are provided in **Appendix B**.

By substituting (4.15) into (4.13) we get

$$\begin{aligned}
\frac{\partial J(\Theta)}{\partial \mu} &= N\Sigma^{-1}\left[\frac{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{x}}}[\mathbf{X}Q(\mathbf{X})]}{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{x}}}[Q(\mathbf{X})]} - \frac{\sum_{n=1}^{N}\mathbf{x}_n}{N}\right] \\
&= N\Sigma^{-1}\left[\frac{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{x}}}[\mathbf{X}Q(\mathbf{X})]}{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{x}}}[Q(\mathbf{X})]} - \hat{\mu}\right]
\end{aligned} \tag{4.17}$$

Now, substituting (4.16) into (4.14) we get

$$\begin{aligned}\frac{\partial J(\Theta)}{\partial \Sigma} &= \frac{N}{2}\Sigma^{-1}\left[\frac{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{x}}}[Q(\mathbf{X})(\mathbf{X}-\mu)(\mathbf{X}-\mu)^T]}{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{x}}}[Q(\mathbf{X})]} - \frac{\sum_{n=1}^{N}(\mathbf{x}_n-\mu)(\mathbf{x}_n-\mu)^T}{N}\right]\Sigma^{-1} \\ &= \frac{N}{2}\Sigma^{-1}\left[\frac{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{x}}}[Q(\mathbf{X})(\mathbf{X}-\mu)(\mathbf{X}-\mu)^T]}{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{x}}}[Q(\mathbf{X})]} - \hat{\Sigma}(\mu)\right]\Sigma^{-1}\end{aligned} \tag{4.18}$$

where $\hat{\mu}$ and $\hat{\Sigma}(\mu)$ are estimated parameters from the sample data set $\mathscr{D}$, such that

$$\hat{\mu} = \frac{\sum_{n=1}^{N}\mathbf{x}_n}{N} \tag{4.19}$$

and

$$\hat{\Sigma}(\mu) = \frac{\sum_{n=1}^{N}(\mathbf{x}_n-\mu)(\mathbf{x}_n-\mu)^T}{N} \tag{4.20}$$

It is noteworthy that $\hat{\Sigma}(\mu)$ is not the estimated covariance from the samples, $\mathscr{D}$. The term, $\hat{\Sigma}(\mu)$, is also a function with the argument of $\mu$.

## PARAMETER LEARNING BASED ON EM METHOD

Given the partial derivatives in (4.17) and (4.18), the underlying multivariate normal distribution's parameters, $\Theta = (\mu, \Sigma)$, can be updated iteratively. However, to evaluate the partial derivatives one has to calculate these expectations, which are expressed in terms of the skewing function $Q$. Considering that the skewing function is determined by the occupied obstacles location in the workspace such that $\mathscr{B} \in \mathscr{W}$. Since the skewing function is an arbitrary function, one cannot implement the expectations in closed-forms. Therefore, we propose a sampling-based expectation-maximization method to implement the parameter learning and approximate the expected value. The first step is generating an auxiliary data set of $M$ samples, $\mathscr{D}_A^l = \{\zeta_j\}_{j=1}^M$, on the grids of the workspace with the resolution of $\Delta x$ and $\Delta y$, where the subscript "A" indicated the auxiliary date set and the superscript "$l$" indicates the iteration index. The second step is to approximate the partial derivatives of the cost function with respect to the current parameters, $\Theta^l$, in (4.17) and (4.18), respectively, by summing over the auxiliary sample data set, such that

$$\frac{\partial J(\Theta^l)}{\partial \mu^l} \approx N(\Sigma^l)^{-1}\left[\frac{\sum_{j=1}^{M}[\zeta_j Q(\zeta_j)\phi_{\mathbf{X}}(\zeta_j|\Theta^l)]}{\sum_{j=1}^{M}[Q(\zeta_j)\phi_{\mathbf{X}}(\zeta_j|\Theta^l)]} - \hat{\mu}\right] \tag{4.21}$$

and

$$\frac{\partial J(\Theta)}{\partial \Sigma} \approx \frac{N}{2}(\Sigma^l)^{-1}\left[\frac{\sum_{j=1}^{M}\left[\phi_{\mathbf{X}}(\zeta_j|\Theta)[Q(\zeta_j)(\zeta_j-\mu^l)(\zeta_j-\mu^l)^T]\right]}{\sum_{j=1}^{M}[\phi_{\mathbf{x}}(\zeta_j|\Theta^l)[Q(\zeta_j)]]}-\hat{\Sigma}(\mu^l)\right](\Sigma^l)^{-1} \tag{4.22}$$

The workspace resolution terms $\Delta x$ and $\Delta y$ exist in both the numerator and denominator canceling each other out.

After obtaining the partial derivatives, we can update the parameter to the next iteration $l+1$ to obtain $\Theta^{l+1} = (\mu^{l+1}, \Sigma^{l+1})$ by

$$\mu^{l+1} = \mu^l - \lambda_\mu \frac{\partial J(\Theta^l)}{\partial \mu^l} \tag{4.23}$$

$$\Sigma^{l+1} = \mu^l - \lambda_\Sigma \frac{\partial J(\Theta^l)}{\partial \Sigma^l} \tag{4.24}$$

here $\lambda_\mu$, $\lambda_\Sigma \in \mathbb{R}^+$ are both positive scalars indicating the learning rates. Final step, is updating the cost function $J(\Theta^{l+1})$ which is evaluated at the $(l+1)$th iteration. The repeated procedure is determined when the index is larger than a user-defined maximum iteration number, $L$¡ or the evaluation of the cost function is less than a user-defined minimum threshold of the cost function, $J_{th}$. Furthermore, using the auxiliary data, $\mathscr{D}_A$, the NLL cost function based on (4.11) can be approximated as

$$J(\Theta^l) \approx N \ln\left[\sum_{j=1}^{M}\phi_{\mathbf{X}}(\zeta_j|\Theta^l)Q(\zeta_j)\Delta x\Delta y\right] - \sum_{n=1}^{N}\ln[\phi_{\mathbf{X}}(\mathbf{x}_n|\mu^l,\Sigma^l)] - \sum_{n=1}^{N}\ln[Q(\mathbf{x}_n)] \tag{4.25}$$

The proposed pseudo summarizes the parameter learning method adopted in **Algorithm 1**. However, to overcome the constraint of maintaining a positive definite for the covariance matrix $\Sigma$. The Cholesky decomposition of the inverse of the covariance matrix is introduced to the NLL cost function, such that

$$\Sigma^{-1} = \mathbf{L}\mathbf{L}^T \tag{4.26}$$

**Algorithm 1** Parameter Learning for BRF-SN based $\mu$ and $\Sigma$

```
Require:
  Sample data set: $\mathscr{D} = \{\mathbf{x}_n\}_n^N$
  Auxiliary sample data set: $\mathscr{D}_A = \{\zeta_j\}_j^M$
  Skewing function $Q$ defined on $\mathscr{W} \in \mathbb{R}^{n_x}$
  Initial parameters $\mu^0 \in \mathbb{R}^{n_x}$ and $\Sigma^0 \in \mathbb{R}^{n_x \times n_x}$
  Initial iteration index: $l \leftarrow 0$
  Approximate the calculation of the cost function: $J(\Theta^0)$ according to (4.11)
  Set the termination parameters: $L$ and $J_{th}$
  Set the learning rates: $\lambda_\mu$ and $\lambda_\Sigma$
  while $(l \leq L)$ and $\big(J(\Theta^l) > J_{th}\big)$ do
    Approximate the partial derivatives, $\frac{\partial J(\Theta)}{\partial \mu}$ and $\frac{\partial J(\Theta)}{\partial \Sigma}$, based on $\mathscr{D}_A$ according to (4.21) and (4.22).
    Update the parameters $\Theta^{l+1}$, according to (4.23) and (4.24)
    Evaluate the approximation of the cost function, $J(\Theta^{l+1})$ according to (4.25)
    Update the iteration index: $l \leftarrow l + 1$
  end while
```

Where $\mathbf{L}$ is a lower triangular matrix. Thus, the $\mathscr{L}(\mathbf{x}_n, \mu, \Sigma)$, can be rewritten as

$$\mathscr{L}(\mathbf{x}_n, \mu, \mathbf{L}) = -ln[\phi_{\mathbf{X}}(\mathbf{x_n}|\mu, \mathbf{L}\mathbf{L}^T)] - ln[Q(\mathbf{x}_n)] + ln\left[\int \phi_{\mathbf{X}}(\mathbf{x}|\mu, \mathbf{L}\mathbf{L}^T)Q(\mathbf{X})\right] \tag{4.27}$$

in order to update the cost function with respect to $\mathbf{L}$, we start by expressing (4.27) derivative with respect to $\mathbf{L}$ such that

$$\frac{\partial}{\partial \mathbf{L}}\mathscr{L}(\mathbf{x}_n, \mu, \mathbf{L}) = \left[(\mathbf{x}_n - \mu)(\mathbf{x}_n - \mu)^T - \frac{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{X}}}[(\mathbf{X} - \mu)(\mathbf{X} - \mu)^T Q(\mathbf{X})]}{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{X}}}[Q(\mathbf{X}])}\right]\mathbf{L} \tag{4.28}$$

The development for (4.28) is provided in **Appendix C**. With that, the partial derivative of the cost function with respect to $\mathbf{L}$ can be expressed as

$$\frac{\partial J(\Theta^l)}{\partial \mathbf{L}} = \sum_{n=1}^{N} \frac{\partial}{\partial \mathbf{L}^l}\mathscr{L}(\mathbf{x}_n, \mu^l, \mathbf{L}^l) = \sum_{n=1}^{N}\left[(\mathbf{x}_i - \mu^l)(\mathbf{x}_i - \mu^l)^T - \frac{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{X}}}[(\mathbf{X} - \mu^l)(\mathbf{X} - \mu^l)^T Q(\mathbf{X})]}{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{X}}}[Q(\mathbf{X})]}\right]\mathbf{L}^l \tag{4.29}$$

Therefore, based on (4.17) and (4.29), given the parameters at the $l$th iteration, $\Theta^l = (\mu^l, \Sigma^l) = \Big(\mu^l, (\mathbf{L}^l\mathbf{L}^{\mathbf{l}^T})^{-1}\Big)$. The parameter $\mathbf{L}^{l+1} = \mathbf{L} - \lambda_{\mathbf{L}}\frac{\partial J(\Theta^l)}{\partial \mathbf{L}^l}$, where $\lambda_{\mathbf{L}} \in \mathbb{R}$ is a small positive scalar-valued learning rate. By applying all the necessary changes mentioned above to the proposed **Algorithm 1**, an updated algorithm is proposed in **Algorithm 2**.

**Algorithm 2** Parameter Learning for BRF-SN based $\mu$ and $\mathbf{L}$

```
Require:
  Sample data set: $\mathscr{D} = \{\mathbf{x}_n\}_i^N$
  Auxiliary sample data set: $\mathscr{D}_A = \{\zeta_j\}_j^M$
  Skewing function $Q$ defined on $\mathscr{W} \in \mathbb{R}^{n_x}$
  Initial parameters $\mu^0 \in \mathbb{R}^{n_x}$ and $\Sigma^0 \in \mathbb{R}^{n_x \times n_x}$
  Initial iteration index: $l \leftarrow 0$
  Approximate the calculation of the cost function: $J(\Theta^0)$ according to (4.25)
  Set the termination parameters: $L$ and $J_{th}$
  Set the learning rates: $\lambda_\mu$ and $\lambda_{\mathbf{L}}$
  while $(l \leq L)$ and $\big(J(\Theta^l) > J_t h\big)$ do
    Obtain $\mathbf{L}$ according to (4.26)
    Approximate the partial derivatives, $\frac{\partial J(\Theta)}{\partial \mu}$ and $\frac{\partial J(\Theta)}{\partial \mathbf{L}}$, based on $\mathscr{D}_A$ according to (4.21) and (4.29).
    Update the parameters $\Theta^{l+1}$, according to (4.23), (4.24) and (4.26)
    Evaluate the approximation of the cost function, $J(\Theta^{l+1})$ according to (4.25)
    Update the iteration index: $l \leftarrow l + 1$
  end while
```

### 4.3.2 PARAMETER LEARNING FOR BRF-SNMM

We extend the BRF-SN distribution to the Bernoulli random field-skewed normal mixture model (BRF-SNMM) associated with the PDF according to (4.9), with component parameters denoted in (4.10). Because the BRF-SNMM is similarly structured to a GMM, we use the EM method to learn parameters from given samples. First, we consider an $N_c$-dimensional categorical random variable, $C \in \{1, ..., N_c\}$, indicating the label of the BRF-SNMM components. Given the data samples $\mathscr{D} = \{\mathbf{x}_n\}_{n=1}^N$ and the parameters, $\Theta_{1:N_c}$, the conditional distribution can be specified by the component weight $\omega$, such that $\omega_i = P(C = i|\Theta_{1:N_c})$, for $i = 1, ..., N_c$. Thus, the BRF-SN distribution, $f_{\mathbf{X}^*}(\mathbf{x}|\mu_i, \Sigma_i)$, can be interpreted as the conditional distribution of the random variable $\mathbf{X}^*$, given $\mathbf{X}^*$ is associate with the $i$th BRF-SN component, such that

$$\varphi_{\mathbf{X}^*|C}(\mathbf{x}|C = 1, \Theta_{1:N_c}) = f_{\mathbf{X}^*}(\mathbf{x}|\mu_i, \Sigma_i) \tag{4.30}$$

Similarly, to (4.11), the cost function can be expressed by

$$\begin{aligned} J(\Theta) &= -ln \prod_{n=1}^{N} \varphi_{\mathbf{X}^*}(\mathbf{x}_n|\mu, \Sigma) \\ &= \sum_{i=1}^{N} \mathscr{L}(\mathbf{x}_n, \Theta_{1:N_c}) \end{aligned} \tag{4.31}$$

where the term, $\mathscr{L}(\mathbf{X}_i, \Theta_{1:N_c})$ is defined by

$$\begin{aligned} \mathscr{L}(\mathbf{X}_i, \Theta_{1:N_c}) &\triangleq -\ln[\varphi_{\mathbf{X}^*}(\mathbf{x}|\Theta_{1:N_c})] \\ &= -\ln\left[\sum_{i=1}^{N_c} \omega_i f_{\mathbf{X}^*}(\mathbf{x}|\mu_i, \Sigma_i)\right] \end{aligned} \tag{4.32}$$

thus, the PDF of the BRF-SNMM defined in (4.30) can be interpreted as the marginal of the joint PDF such that

$$\begin{aligned} \varphi_{\mathbf{X}^*}(\mathbf{x}|\Theta_{1:N_c}) &= \sum_{i=1}^{N_c} P(C = i|\Theta_{1:N_c}) \varphi_{\mathbf{X}^*|C}(\mathbf{x}|C = i, \Theta_{1:N_c}) \\ &= \varphi_{\mathbf{X}^*,C}(\mathbf{x}, C = i, \Theta_{1:N_c}) \end{aligned} \tag{4.33}$$

then, the complete likelihood can be interpreted in terms of the expectation with respect to $N_c$, such that

$$\begin{aligned} \mathscr{L}(\mathbf{x}_n, \Theta_{1:N_c}) &= -\ln\left[\varphi_{\mathbf{X}^*}(\mathbf{x}|\Theta_{1:N_c})\right] \\ &= -\ln\left[\varphi_{\mathbf{X}^*,C}(\mathbf{x}_i, C = i|\Theta_{1:N_c})\right] \\ &= -\ln\left\{\mathbb{E}_{C|\mathbf{x}_i}\left[\frac{\varphi_{\mathbf{X}^*,C}(\mathbf{x}_n, C = i|\Theta_{1:N_c})}{P(C = i|\mathbf{x}_i)}\right]\right\} \end{aligned} \tag{4.34}$$

Instead of minimizing the NLL of the data sample $\mathscr{D}$, like the GMM approach, we solve the parameter learning problem by minimizing the upper bound of the NLL which is denoted by

$$\hat{J}(\Gamma, \Theta_{1:N_c}) \triangleq \sum_{n=1}^{N} \hat{\mathscr{L}}(\mathbf{x}_n, \gamma_i, \Theta_{1:N_c}) \tag{4.35}$$

Here $\gamma_i \triangleq P(C=i|\mathbf{x}_i)$ and $\Gamma = [\gamma_{n,i}]_{n=1,i=1}^{N,N_c}$ is a $N \times N_c$ matrix. We use the EM method to solve the optimization problem. In the E-step, we calculate the distribution $P(C|\mathbf{x}_i)$ for every sample $\mathbf{x}_n$, given the parameters, $\Theta^l_{1:N_c}$, at the $l$-th iteration, such that

$$\begin{aligned}
\gamma_{n,i} &= \varphi_{C|\mathbf{X}^*}(C=i|\mathbf{x}_n, \Theta^l_{1:N_c}) \\
&= \frac{\varphi_{\mathbf{X}^*}(\mathbf{x}_n, C=i|\Theta^l_{1:N_c})}{\sum_{i=1}^{N_c} \varphi_{\mathbf{X}^*}(\mathbf{x}_n, C=i|\Theta^l_{1:N_c})} \\
&= \frac{\omega^l_i f_{\mathbf{X}^*}(\mathbf{x}|\mu^l_i, \Sigma^l_i)}{\sum_{i=1}^{N_c} \omega^l_i f_{\mathbf{X}^*}(\mathbf{x}|\mu^l_i, \Sigma^l_i)}
\end{aligned} \tag{4.36}$$

where the superscript $l$ indicated that the distributions, $P(C|\mathbf{x}_n)$ for $n = 1, ..., N$, specified by $\Gamma^l = [\gamma_{n,i}]_{n=1,i=1}^{N,N_c}$ are all obtained based on $\Theta^l_{1:N_c}$. In the M-step, fixing $\Gamma^l$ obtained in the E-step, we need to minimize $\hat{J}(\Gamma^l, \Theta_{1:N_c})$ with respect to the parameters for the $(l+1)$th iteration, such that

$$\begin{aligned}
\Theta^{l+1}_{1:N_c} &= \arg\ \min \hat{J}(\Gamma^l, \Theta_{1:N_c}) \\
&= \arg\ \min \sum_{n=1}^{N} \hat{\mathscr{L}}(\mathbf{x}_n, \gamma^l_i, \Theta_{1:N_c}) \\
&= \arg\ \min \sum_{n=1}^{N} \sum_{i=1}^{N_c} \gamma^l_{n,i} \ln \frac{\gamma^l_{n,N_c}}{\omega_i f_{\mathbf{X}^*}(\mathbf{x}|\mu_i, \Sigma_i)} \\
&= \arg\ \min \sum_{n=1}^{N} \sum_{i=1}^{N_c} \left\{ -\gamma^l_{n,i} \left[ \ln\{\omega_i\} + \ln f_{\mathbf{X}^*}(\mathbf{x}|\mu_i, \Sigma_i) \right] \right\}
\end{aligned} \tag{4.37}$$

Referring back to section (3.2.2), $\sum_{i=1}^{N_c} \omega_i = 1$ and $\omega_i > 0$ for all $N_c$, with the weight component $\omega^{l+1}_i = [\omega^{l+1}_1, ..., \omega^{l+1}_{N_c}]$ can be obtained by

$$\omega^{l+1}_i = \frac{\sum_{i=1}^{N} \gamma^l_{n,i}}{N}, \qquad \text{for } i = 1, ..., N_c \tag{4.38}$$

In order to obtain the parameters for $\Theta^{l+1}_i = (\mu^{l+1}_i, \Sigma^{l+1}_i$, first we calculate the derivatives of $\hat{J}(\Gamma^l, \Theta_{1:N_c})$ with respect to $\mu_i$ and $\mathbf{L}_i$, since we established the need to use Cholesky decomposition to maintain a positive definite covariance matrix $\Sigma$, the partial derivatives are denoted as follows. According to (4.17), the BRF-SNMM will incorporate the weight components such that the partial derivative with respect to $\mu$

$$\frac{\partial}{\partial \mu_i}\hat{J}(\Gamma^l, \Theta_{1:N_c}) = \omega_i^{l+1} N \Sigma_i^{-1} \left[ \frac{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{x}_n}}[\mathbf{X}Q(\mathbf{X})]}{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{x}_n}}[Q(\mathbf{X})]} - \hat{\mu}_i \right] \tag{4.39}$$

Now, According to (4.29), the BRF-SNMM will incorporate the weight components such that the partial derivative with respect to $\mathbf{L}_i$

$$\frac{\partial}{\partial \mathbf{L}_i}\hat{J}(\Gamma^l, \Theta_i^l) = \omega_i^{l+1} \sum_{n=1}^{N} \left[ (\mathbf{x}_n - \mu)(\mathbf{x}_n - \mu)^T - \frac{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{X}}}[(\mathbf{X}-\mu)(\mathbf{X}-\mu)^T Q(\mathbf{X})]}{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{X}}} Q(\mathbf{X})} \right] \mathbf{L} \tag{4.40}$$

Furthermore, summing over the auxiliary sample, to (4.39) and (4.40) respectively, we obtain

$$\frac{\partial}{\partial \mu_i}\hat{J}(\Gamma^l, \Theta_{1:N_c}) \approx N(\Sigma^l)^{-1} \left[ \frac{\sum_{j=1}^{M}[\zeta_j Q(\zeta_j)\phi_{\mathbf{X}}(\zeta_j|\Theta^l)]}{\sum_{j=1}^{M}[Q(\zeta_j)\phi_{\mathbf{X}}(\zeta_j|\Theta^l)]} - \hat{\mu}_i \right] \tag{4.41}$$

$$\frac{\partial}{\partial \mathbf{L}_i}\hat{J}(\Gamma^l, \Theta_i) \approx N \left[ \hat{\Sigma}_i(\mu_i^l) - \frac{\sum_{j=1}^{M}\left[Q(\zeta_j)(\zeta_j - \mu^l)(\zeta_j - \mu^l)^T \phi_{\mathbf{X}}(\zeta_j|\Theta^l)\right]}{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{X}}} Q(\mathbf{X})} \mathbf{L}^l \right] \tag{4.42}$$

here, $\hat{\mu}_i$ and $\hat{\Sigma}_i(\mu_i)$ are defined in (4.18) and (4.19), respectively, for all $i$. Furthermore, the cost function $J(\Theta_{1:N_c})$ and its upper nound $\hat{J}(\Gamma, \Theta_{1:N_c})$ can be expressed and approximated, respectively, by

$$\begin{aligned} J(\Theta_{1:N_c}) &= -\ln \prod \varphi_{\mathbf{X}^*}(\mathbf{x}_n|\Theta_{1:N_c}) \\ &= -\sum_{n=1}^{N} \ln[\sum_{i=1}^{N_c} \omega_i f_{\mathbf{X}^*}(\mathbf{x}|\mu_i, \Sigma_i)] \\ &= -\sum_{n=1}^{N} \ln \left[ \sum_{i=1}^{N_c} \omega_i \frac{\phi_{\mathbf{X},i}(\mathbf{x}_n|\mu_i, \Sigma_i) Q_(\mathbf{x}_i)}{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{X},i}}[Q_(\mathbf{x}_n)]} \right] \\ &\approx -\sum_{n=1}^{N} \ln \left[ \sum_{i=1}^{N_c} \omega_i \frac{\phi_{\mathbf{X},i}(\mathbf{x}_n|\mu_i, \Sigma_i) Q(\mathbf{x}_n)}{\sum_{j=1}^{M} \phi_{\mathbf{X}}(\zeta_j|\Theta_i^l) Q(\zeta_j) \Delta x \Delta y} \right] \end{aligned} \tag{4.43}$$

and

$$
\begin{aligned}
\hat{J}(\Gamma, \Theta_{1:N_c}) &= \sum_{n=1}^{N} \sum_{i=1}^{N_c} \left\{ -\gamma_{n,i}^{l} \left[ \ln \omega_i + \ln f_{\mathbf{X}^*}(\mathbf{x}_n | \mu_i, \Sigma_i) \right] \right\} \\
&= N \sum_{i=1}^{N_c} \omega_i^{l+1} - \sum_{n=1}^{N} \sum_{i=1}^{N_c} -\gamma_{n,i}^{l} \ln f_{\mathbf{X}^*}(\mathbf{x}_n | \mu_i, \Sigma_i) \qquad (4.44) \\
&= \sum_{i=1}^{N_c} \omega_i^{l+1} \hat{J}_i(\Gamma_i^l, \Theta_i^l)
\end{aligned}
$$

where

$$
\hat{J}_i(\Gamma_i^l, \Theta_i^l) = N \ln \left[ \sum_{j=1}^{M} \phi_{\mathbf{X}}(\zeta_j | \Theta_i^l) Q(\zeta_j) \Delta x \Delta y \right] - N \ln \omega_i - \frac{N}{\sum_{n=1}^{N} \gamma_{n,i}^{l}} \ln \left[ \phi_{\mathbf{X}}(\mathbf{x} | \Theta_i) Q(\mathbf{x}_n) \right] \qquad (4.45)
$$

Finally, we can update the parameters to obtain $\Theta_i^{l_1+1} = (\mu_i^{l_1+1}, \Sigma_i^{l_1+1})$ for $i = 1, ..., N_c$, such that

$$
\mu_i^{l+1} = \mu_i^l - \lambda_\mu \frac{\partial \hat{J}(\Theta^l)}{\partial \mu_i} \qquad (4.46)
$$

$$
\mathbf{L}_i^{l+1} = \mathbf{L}_i^l - \lambda_{\mathbf{L}} \frac{\partial \hat{J}(\Theta^l)}{\partial \mathbf{L}_i} \qquad (4.47)
$$

$$
\Sigma_i^{l+1} = \left( \mathbf{L}_i^{l+1} \mathbf{L}_i^{l+1^T} \right)^{-1} \qquad (4.48)
$$

Applying the BRF-SNMM, in **Algorithm 3**, there are 2 while loops associated with 2 iteration indices, $l_1$ and $l_2$, respectively. The iterations associated with the index $l_1$ is applied to learn the parameter $\omega$, while with the fixed $\omega$, the iteration associated with the index $l_2$ are applied to learn the parameters, $\Theta_i$ separately, for $i = 1, ..., N_c$.

**Algorithm 3** Parameter Learning for BRF-SNMM based $\mu$ and $\mathbf{L}$

```
Require:
  Sample data set: $\mathscr{D} = \{\mathbf{x}_n\}_n^N$
  Auxiliary sample data set: $\mathscr{D}_A = \{\zeta_j\}_j^M$
  Skewing function $Q$ defined on $\mathscr{W} \in \mathbb{R}^{n_x}$
  Set the number of mixture components: $N_c$
  Initial iteration index: $l_1 \leftarrow 0$ and $l_2 \leftarrow 0$
  Initial parameters $\Theta_{1:N_c}^{l_1} = \{(\omega_i^{l_1}, \Theta_i^{l_2})\}_{i=1}^{N_c}$
  Approximate the upper bound of the cost function: $\hat{J}_i(\Gamma_i^l, \Theta_i^l)$ according to (4.44)
  Set the termination parameters: $L_1$, $L_2$ and $\Delta J_{th}$
  Set the learning rates: $\lambda_\mu$ and $\lambda_{\mathbf{L}}$
  while $(l_1 \leq L_1)$ and $J(\Theta_{1:N_c}^{l_1} > J_{th})$ do
    Generate the matrix $\Gamma^{l_1} = [\gamma_{n,i}^{l_1}]_{n=1,i=1}^{N,N_c}$ according to (4.36) given $\Theta_{1:N_c}^{l_1}$ and $\mathscr{D}_A$
    Calculate component weights, $\omega^{l_1+1}$, according to (4.38)
    Update parameters: $\Theta_{1:N_c}^{l_1+1} = \{(\omega_i^{l_1+1}, \Theta_i^{l_2})\}_{i=1}^{N_c}$
    for $i = 1, ..., N_c$ do
      Reset the iteration index $l_2 \leftarrow 0$
      while $(l_2 < L_2)$ and $\hat{J}_i(\Gamma_i^{l_1}, \Theta_i^{l_2}) > \Delta J_{th}$ do
        Obtain $\mathbf{L}_i^{l_2}$ based on $\Sigma_i^{l_2}$, according to (4.26)
        Approximate $\frac{\partial}{\partial \mu_i^{l_2}} \hat{J}(\Gamma^{l_1}, \Theta_{1:N_c}^{l_2})$ and $\frac{\partial}{\partial \mathbf{L}_i^{l_2}} \hat{J}(\Gamma^{l_1}, \Theta_i^{l_2})$, according to (4.39) and (4.40), respectively.
        Update $\Theta_i^{l_2+1} = (\mu_i^{l_2+1}, \Sigma_i^{l_2+1})$ according to (4.46) and (4.47)
        Update parameters: $\Theta_i^{l_2} \leftarrow \Theta_i^{l_2+1}$
      end while
    end for
    Update the iteration index: $l \leftarrow l + 1$
  end while
```

# CHAPTER 5

# PATH-PLANNING BASED ON SKEW-NORMAL MIXTURE MODEL

In this section, we apply the proposed BRF-SNMM to solve the path-planning problems for VLSR systems in a known obstacle deployed environment. In the VLSR systems, the macroscopic state is represented bt the agents' PDF. Then, in the macroscopic scale, the path-planning task of the VLSR systems can be formulated as the distributions path-planning problem, where a trajectory of agents' PDF in the workspace over a time interval $[t_0, t_f]$ is obtained, to guide the agents' microscopic controls. Specifically, the agents' PDF at time $t \in [t_0, t_f]$ is described by a time-varying BRF-SNMM, $\varphi(\mathbf{x}, t) = \varphi_{\mathbf{x}^*}(\mathbf{x}|\Theta_{1:N_c}(t))$ defined in (4.9), which is specified by a time-varying parameter set $\Theta_{1:N_c}(t)$. In this path-planning problem, for simplicity, we make these 3 assumptions:

- The number of the BRF-SN component $N_c$ and the component weights $\omega$ are all known constants.
- The desited agents' distribution is a BRF-SN, $\varphi = f_{\mathbf{X}^*}(\mathbf{x}|\Theta_f)$, specified by the parameter set, $\Theta_f = (\mu_f, \Sigma_f)$
- The obstacles deployed in the workspace can all be presented geometrically by convex polygons, circles or ovals.

## 5.1 PATH-PLANNING FOR SKEWED NORMAL DISTRIBUTION

In this section, we only focus on the macroscopic scale path-planning task and propose twp approaches to generate the BRF-SNMM trajectory of agents from the initial distribution, $\varphi(\mathbf{x}, t_0)$, to the desired distribution, $\varphi_f$.

### 5.1.1 PATH-PLANNING OF DISTRIBUTIONS BASED ON DISPLACEMENT INTERPOLATION FOR SND

Given the skewing function $Q_Y$, the time-varying BRFSND $f_{\mathbf{X}^*}(\mathbf{x}, t)$ is specified by the parameter set $\Theta_i(t) = (\mu_i(t), \Sigma_i(t))$, which can also specify a time-varying Gaussian distribution, such that

$$g(\mathbf{x}, t) \triangleq \sum_{i=1}^{N_c} \phi_{\mathbf{X}}(\mathbf{x}|\mu_i(t), \Sigma_i(t)) \tag{5.1}$$

Similarly, the parameter set of the desired BRF-SN, $\Theta_f$, can specify a desired normal PDF, $g_f = \phi_{\mathbf{X}}(\mathbf{x}|\mu_f, \Sigma_f)$. Thus, it is reasonable to obtain a trajectory of the time-varying Gaussian distribution from the start, $g(\mathbf{x}, t_0)$, to the desired normal distribution, $g_f$, in the obstacle-free workspace first, then form the time-varying BRF-SND trajectory with the obtained time-varying parameter set, $\Theta_i(t)$. Now, we apply the displacement interpolation (DI) between $g(\mathbf{x}, t_0)$ and $g_f$ to specify the time-varying parameter set, $\Theta_i(t) = (\mu_i(t), \Sigma_i(t))$ for $i = 1, \ldots, N_c$ [16], such that

$$\mu_i(t) = \frac{t_f - t - t_0}{t_f - t_0}\mu_i(t_0) + \frac{t - t_0}{t_f - t_0}\mu_f \tag{5.2}$$

$$\Sigma_i(t) = \Sigma_i(t_0)^{-1/2}\left(\frac{t_f - t - t_0}{t_f - t_0}\Sigma_i(t_0) + \frac{t}{t - t_0}\cdot\left[\Sigma_i(t_0)^{1/2}\Sigma_f\Sigma_i(t_0)^{1/2}\right]\right)\Sigma_i(t_0)^{-1/2} \tag{5.3}$$

Although the obtained Gaussian distribution trajectory, $g(\mathbf{x}, t|\Theta_i(t))$, is a geodesic path, which provides the shortest $\ell_2$ Wasserstein distance [17], it is not guaranteed that the obtained BRF-SNMM trajectory can give $\ell_2$ Wasserstein distance since the time-varying parameter set is obtained without considering the obstacles.

### 5.1.2 PATH-PLANNING OF DISTRIBUTIONS BASED ON ARTIFICIAL POTENTIAL FIELD FOR SND

Considering that the environment information is embedded in the BRF-SN the APF approach can be applied to solve the path-planning. Two APF methods are developed and theit derivatives with respect to the parameter set $\Theta_i(t) = (\mu(_i(t), \Sigma_i(t))$ for $i = 1, ..., N_c$ are approximated to generate the BRF-SN trajectory.

#### POTENTIAL FIELD BASED L-2 NORM

First, to reduce the difference between the current agents' distribution and the desired distribution, the following potential field based on the swaure of the $\ell - 2$ norm is applied, such that

$$U_{SN}(\mu, \Sigma) \triangleq \frac{1}{2}\int_{\mathscr{W}} \left[f_{\mathbf{X}^*}(\mathbf{x}|\mu, \Sigma) - f_{\mathbf{X}_T^*}(\mathbf{x}|\mu_T, \Sigma_T)\right]^2 d\mathbf{x} \tag{5.4}$$

Now, we derive with respect to $\mu$

$$
\begin{aligned}
\frac{\partial}{\partial \mu} U_{SN}(\mu, \Sigma) &= \frac{1}{2} \int_{\mathscr{W}} \frac{\partial}{\partial \mu} \left[ f_{\mathbf{X}^*}(\mathbf{x}|\mu, \Sigma) - f_{\mathbf{X}_T^*}(\mathbf{x}|\mu_T, \Sigma_T) \right]^2 d\mathbf{x} \\
&= \int_{\mathscr{W}} \left[ f_{\mathbf{X}^*}(\mathbf{x}|\mu, \Sigma) - f_{\mathbf{X}_T^*}(\mathbf{x}|\mu_T, \Sigma_T) \right] \frac{\partial f_{\mathbf{X}^*}(\mathbf{x}|\mu, \Sigma)}{\partial \mu} d\mathbf{x}
\end{aligned}
\tag{5.5}
$$

Here, because $\ = \exp^{\ln(\mathbf{x})}$ and $\mathscr{L}(\mathbf{x}, \mu, \Sigma) = -\ln\left[f_{\mathbf{X}^*}(\mathbf{x}|\mu, \Sigma)\right]$ defined previously in (4.12) the derivative term can be expressed by

$$
\begin{aligned}
\frac{\partial f_{\mathbf{X}^*}(\mathbf{x}|\mu, \Sigma)}{\partial \mu} &= \frac{\partial}{\partial \mu} e^{\ln[f_{\mathbf{X}^*}(\mathbf{x}|\mu, \Sigma)]} \\
&= e^{\ln[f_{\mathbf{X}^*}(\mathbf{x}|\mu, \Sigma)]} \frac{\partial}{\partial \mu} \ln[f_{\mathbf{X}^*}(\mathbf{x}|\mu, \Sigma)] \\
&= -f_{\mathbf{X}^*}(\mathbf{x}|\mu, \Sigma)] \frac{\partial}{\partial \mu} \mathscr{L}(\mathbf{x}, \mu, \Sigma) \\
&= -\Sigma^{-1} f_{\mathbf{X}^*}(\mathbf{x}|\mu, \Sigma)] \left[ \frac{\mathbb{E}_{\mathbf{X} \sim \phi_{\mathbf{x}}}[\mathbf{X} Q(\mathbf{X})]}{\mathbb{E}_{\mathbf{X} \sim \phi_{\mathbf{x}}}[Q(\mathbf{X})]} - \mathbf{x} \right]
\end{aligned}
\tag{5.6}
$$

By substituting (5.6) in (5.5), we have

$$
\frac{\partial}{\partial \mu} U_{SN}(\mu, \Sigma) = -\Sigma^{-1} \int_{\mathscr{W}} \left[ f_{\mathbf{X}^*}(\mathbf{x}|\mu, \Sigma) - f_{\mathbf{X}_T^*}(\mathbf{x}|\mu_T, \Sigma_T) \right] f_{\mathbf{X}^*}(\mathbf{x}|\mu, \Sigma)] \left[ \frac{\mathbb{E}_{\mathbf{X} \sim \phi_{\mathbf{x}}}[\mathbf{X} Q(\mathbf{X})]}{\mathbb{E}_{\mathbf{X} \sim \phi_{\mathbf{x}}}[Q(\mathbf{X})]} - \mathbf{x} \right] d\mathbf{x}
\tag{5.7}
$$

Furthermore, considering the Cholesky decomposition defined in (4.26) such that, $\Sigma^{-1} = \mathbf{L}\mathbf{L}^T$, the derivative with respect to $\mathbf{L}$ can be expressed by

$$
\begin{aligned}
\frac{\partial}{\partial \mathbf{L}} U_{sn}(\mu, \Sigma) &= \int_{\mathscr{W}} \frac{1}{2} \int_{\mathscr{W}} \frac{\partial}{\partial \mathbf{L}} \left[ f_{\mathbf{X}^*}(\mathbf{x}|\mu, \Sigma) - f_{\mathbf{X}_T^*}(\mathbf{x}|\mu_T, \Sigma_T) \right]^2 d\mathbf{x} \\
&= \int_{\mathscr{W}} \left[ f_{\mathbf{X}^*}(\mathbf{x}|\Theta) - f_{\mathbf{X}_T^*}(\mathbf{x}|\Theta) \right] \frac{\partial f_{\mathbf{X}^*}(\mathbf{x}|\Theta)}{\partial \mathbf{L}} d\mathbf{x} \\
&= \int_{\mathscr{W}} \left[ f_{\mathbf{X}^*}(\mathbf{x}|\Theta) - f_{\mathbf{X}_T^*}(\mathbf{x}|\Theta) \right] .f_{\mathbf{X}^*}(\mathbf{x}|\Theta) \frac{\partial}{\partial \mathbf{L}} \mathscr{L}(\mathbf{x}, \mu, \Sigma) d\mathbf{x}
\end{aligned}
\tag{5.8}
$$

Considering (4.28) where we solved for $\frac{\partial}{\partial \mathbf{L}} \mathscr{L}(\mathbf{x}, \mu, \Sigma)$, replacing (4.28) in (5.1) we obtain

$$
\begin{aligned}
\frac{\partial}{\partial \mathbf{L}} U_{sn}(\mu, \Sigma) = \int_{\mathscr{W}} & \left[ f_{\mathbf{X}^*}(\mathbf{x}|\Theta) - f_{\mathbf{X}_T^*}(\mathbf{x}|\Theta) \right] .f_{\mathbf{X}^*}(\mathbf{x}|\Theta) \\
& . \left[ (\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^T - \frac{\mathbb{E}_{\mathbf{X} \sim \phi_{\mathbf{x}}}[(\mathbf{X} - \mu)(\mathbf{X} - \mu)^T Q(\mathbf{X})]}{\mathbb{E}_{\mathbf{X} \sim \phi_{\mathbf{x}}}[Q(\mathbf{X})]} \right] \mathbf{L}
\end{aligned}
\tag{5.9}
$$

Finally, to approximate the partial derivatives based on the auxiliary samples, $\mathscr{D}_A$, such that

$$\frac{\partial}{\partial \mu} U_{SN}(\mu, \Sigma) \approx -\Sigma^{-1} \sum_{j=1}^{M} \left[ f_{\mathbf{X}^*}(\zeta_j|\Theta) - f_{\mathbf{X}_T^*}(\zeta|\Theta_T) \right] f_{\mathbf{X}^*}(\zeta_j|\Theta)(\zeta_j - \tilde{\mathbf{x}})\Delta x \Delta y \tag{5.10}$$

and

$$\frac{\partial}{\partial \mathbf{L}} U_{SN}(\mu, \Sigma) = \sum_{j=1}^{M} \left[ f_{\mathbf{X}^*}(\zeta_j|\Theta) - f_{\mathbf{X}_T^*}(\zeta_j|\Theta_T) \right] . f_{\mathbf{X}^*}(\zeta|\Theta) \left[ (\zeta_j - \mu)(\zeta_j - \mu)^T - \tilde{\Sigma} \right] \mathbf{L} \Delta x \Delta y \tag{5.11}$$

Where,

$$\frac{\mathbb{E}_{\mathbf{X} \sim \phi_{\mathbf{x}}}[\mathbf{X} Q(\mathbf{X})]}{\mathbb{E}_{\mathbf{X} \sim \phi_{\mathbf{x}}}[Q(\mathbf{X})]} \approx \tilde{\mathbf{x}} \triangleq \frac{\sum_{j=1}^{N} \phi_{\mathbf{X}}(\zeta_j|\Theta) \zeta_j Q(\zeta_j)}{\sum_{j=1}^{M} \phi_{\mathbf{X}}(\zeta_j|\Theta) Q(\zeta_j)} \tag{5.12}$$

and

$$\frac{\mathbb{E}_{\mathbf{X} \sim \phi_{\mathbf{x}}}[(\mathbf{X} - \mu)(\mathbf{X} - \mu)^T Q(\mathbf{X})]}{\mathbb{E}_{\mathbf{X} \sim \phi_{\mathbf{x}}}[Q(\mathbf{X})]} \approx \tilde{\Sigma} \triangleq \frac{\sum_{j=1}^{N} \phi_{\mathbf{X}}(\zeta_j|\Theta)(\zeta_j - \mu)(\zeta - \mu)^T Q(\zeta_j)}{\sum_{j=1}^{M} \phi_{\mathbf{X}}(\zeta_j|\Theta) Q(\zeta_j)} \tag{5.13}$$

## POTENTIAL FIELD BASED CS-DIVERGENCE

Using CS-divergence between the current distribution and the target distribution as the cost function, compared to the traditional optimization of the cost function method, have 2 major advantages, 1) the integration resolution will affect the results, 2) the stop-criterion is not easy to set since the cost function is not normalized, the CS-divergence cost function is obtained such that,

$$\begin{aligned} U_{CS}(\mu, \Sigma) &\triangleq (f_{\mathbf{X}^*} || f_{\mathbf{X}_T^*}) \\ &= -\ln \frac{\int_{\mathscr{W}} f_{\mathbf{X}^*}(\mathbf{x}|\mu, \Sigma) f_{\mathbf{X}_T^*}(\mathbf{x}|\mu_T, \Sigma_T) d\mathbf{x}}{\sqrt{\int_{\mathscr{W}} f_{\mathbf{X}^*}(\mathbf{x}|\mu, \Sigma) f_{\mathbf{X}_T^*}(\mathbf{x}|\mu_T, \Sigma_T) d\mathbf{x}}} \end{aligned} \tag{5.14}$$

The derivative of $U_{CS}$ with respect to $\mu$ and $\Sigma$ can be expressed as

$$\begin{aligned}
\frac{\partial}{\partial \mu} U_{CS}(\mu, \Sigma) &= \frac{1}{2} \frac{\partial}{\partial \mu} \ln \int_{\mathscr{W}} f_{\mathbf{X}^*}^2(\mathbf{x}|\mu, \Sigma) d\mathbf{x} - \frac{\partial}{\partial \mu} \ln \int_{\mathscr{W}} f_{\mathbf{X}^*}(\mathbf{x}|\mu, \Sigma) f_{\mathbf{X}_T^*}(\mathbf{x}|\mu_T, \Sigma_T) d\mathbf{x} \\
&= \frac{1}{2 \int_{\mathscr{W}} f_{\mathbf{X}^*}^2(\mathbf{x}|\mu, \Sigma) d\mathbf{x}} \frac{\partial}{\partial \mu} \left[ \int_{\mathscr{W}} f_{\mathbf{X}^*}^2(\mathbf{x}|\mu, \Sigma) d\mathbf{x} \right] \\
&- \frac{1}{\int_{\mathscr{W}} f_{\mathbf{X}^*}(\mathbf{x}|\mu, \Sigma) f_{\mathbf{X}_T^*}(\mathbf{x}|\mu_T, \Sigma_T) d\mathbf{x}} \frac{\partial}{\partial \mu} \left[ \int_{\mathscr{W}} f_{\mathbf{X}^*}(\mathbf{x}|\mu, \Sigma) f_{\mathbf{X}_T^*}(\mathbf{x}|\mu_T, \Sigma_T) d\mathbf{x} \right] \\
&= \int_{\mathscr{W}} \left[ \frac{f_{\mathbf{X}^*}(\mathbf{x}|\mu, \Sigma)}{c_1} - \frac{f_{\mathbf{X}_T^*}(\mathbf{x}|\mu_T, \Sigma_T)}{c_2} \right] \frac{\partial f_{\mathbf{X}^*}(\mathbf{x}|\mu, \Sigma)}{\partial \mu} d\mathbf{x}
\end{aligned} \tag{5.15}$$

and

$$\frac{\partial}{\partial \mathbf{L}} U_{CS}(\mu, \Sigma) = \int_{\mathscr{W}} \left[ \frac{f_{\mathbf{X}^*}(\mathbf{x}|\mu, \Sigma)}{c_1} - \frac{f_{\mathbf{X}_T^*}(\mathbf{x}|\mu_T, \Sigma_T)}{c_2} \right] \frac{\partial f_{\mathbf{X}^*}(\mathbf{x}|\mu, \Sigma)}{\partial \mathbf{L}} d\mathbf{x} \tag{5.16}$$

where

$$c_1 = \int_{\mathscr{W}} f_{\mathbf{X}^*}^2(\mathbf{x}|\mu, \Sigma) d\mathbf{x} \tag{5.17}$$

$$c_2 = \int_{\mathscr{W}} f_{\mathbf{X}^*}(\mathbf{x}|\mu, \Sigma) f_{\mathbf{X}_T^*}(\mathbf{x}|\mu_T, \Sigma_T) d\mathbf{x} \tag{5.18}$$

The two partial derivatives, $\frac{\partial f_{\mathbf{X}^*}(\mathbf{x}|\mu, \Sigma)}{\partial \mu} d\mathbf{x}$ and $\frac{\partial f_{\mathbf{X}^*}(\mathbf{x}|\mu, \Sigma)}{\partial \mathbf{L}} d\mathbf{x}$, were solved in (4.17) and in (4.28), respectively. Substituting the derivative in (5.15) and (5.16) we obtain the following

$$\frac{\partial}{\partial \mu} U_{CS}(\mu, \Sigma) = \Sigma^{-1} \int_{\mathscr{W}} \left[ \frac{f_{\mathbf{X}^*}(\mathbf{x}|\mu, \Sigma)}{c_1} - \frac{f_{\mathbf{X}_T^*}(\mathbf{x}|\mu_T, \Sigma_T)}{c_2} \right] \left[ \frac{\mathbb{E}_{\mathbf{X} \sim \phi_{\mathbf{x}}}[\mathbf{X} Q(\mathbf{X})]}{\mathbb{E}_{\mathbf{X} \sim \phi_{\mathbf{x}}}[Q(\mathbf{X})]} - \mathbf{x} \right] d\mathbf{x} \tag{5.19}$$

and

$$\begin{aligned}
\frac{\partial}{\partial \mathbf{L}} U_{CS}(\mu, \Sigma) = \int_{\mathscr{W}} & \left[ \frac{f_{\mathbf{X}^*}(\mathbf{x}|\mu, \Sigma)}{c_1} - \frac{f_{\mathbf{X}_T^*}(\mathbf{x}|\mu_T, \Sigma_T)}{c_2} \right] \cdot \\
& \left[ (\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^T - \frac{\mathbb{E}_{\mathbf{X} \sim \phi_{\mathbf{X}}}[(\mathbf{X} - \mu)(\mathbf{X} - \mu)^T Q(\mathbf{X})]}{\mathbb{E}_{\mathbf{X} \sim \phi_{\mathbf{X}}}[Q(\mathbf{X})]} \right] \mathbf{L} d\mathbf{x}
\end{aligned} \tag{5.20}$$

Furthermore, consider that

$$\begin{aligned}
&\int_{\mathscr{W}} \left[ \frac{f_{\mathbf{X}^*}(\mathbf{x}|\mu,\Sigma)}{c_1} - \frac{f_{\mathbf{X}_T^*}(\mathbf{x}|\mu_T,\Sigma_T)}{c_2} \right] f_{\mathbf{X}^*}(\mathbf{x}|\mu,\Sigma) d\mathbf{x} \\
&= \frac{\int_{\mathscr{W}} f^2_{\mathbf{X}^*}(\mathbf{x}|\mu,\Sigma) d\mathbf{x}}{c_1} - \frac{\int_{\mathscr{W}} f_{\mathbf{X}_T^*}(\mathbf{x}|\mu_T,\Sigma_T) f_{\mathbf{X}^*}(\mathbf{x}|\mu,\Sigma) d\mathbf{x}}{c_2} \\
&= \frac{c_1}{c_1} - \frac{c_2}{c_2} \\
&= 0
\end{aligned} \tag{5.21}$$

and the terms, $\frac{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{x}}}[\mathbf{X}Q(\mathbf{X})]}{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{x}}}[Q(\mathbf{X})]}$ and $\frac{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{X}}}[(\mathbf{X}-\mu)(\mathbf{X}-\mu)^T Q(\mathbf{X})]}{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{X}}}[Q(\mathbf{X})]}$ in (5.19) and (5.20), are both constants. One can have

$$\frac{\partial}{\partial \mu} U_{CS}(\mu,\Sigma) = \Sigma^{-1} \int_{\mathscr{W}} \left[ \frac{f_{\mathbf{X}^*}(\mathbf{x}|\mu,\Sigma)}{c_1} - \frac{f_{\mathbf{X}_T^*}(\mathbf{x}|\mu_T,\Sigma_T)}{c_2} \right] f_{\mathbf{X}^*}(\mathbf{x}|\mu,\Sigma) \mathbf{x} d\mathbf{x} \tag{5.22}$$

and

$$\frac{\partial}{\partial \mathbf{L}} U_{CS}(\mu,\Sigma) = \int_{\mathscr{W}} \left[ \frac{f_{\mathbf{X}^*}(\mathbf{x}|\mu,\Sigma)}{c_1} - \frac{f_{\mathbf{X}_T^*}(\mathbf{x}|\mu_T,\Sigma_T)}{c_2} \right] f_{\mathbf{X}^*}(\mathbf{x}|\mu,\Sigma) \left[ (\mathbf{x}-\mu)(\mathbf{x}-\mu)^T \right] d\mathbf{x}\mathbf{L} \tag{5.23}$$

According to the definition of the SND in **section** (3.2.1), we have

$$\begin{aligned}
\frac{f_{\mathbf{X}^*}(\mathbf{x}|\mu,\Sigma) f_{\mathbf{X}^*}(\mathbf{x}|\mu,\Sigma)}{c_1} &= \frac{f^2_{\mathbf{X}^*}(\mathbf{x}|\mu,\Sigma)}{c_2} \\
&= \frac{f^2_{\mathbf{X}^*}(\mathbf{x}|\mu,\Sigma)}{\int_{\mathscr{W}} f^2_{\mathbf{X}^*}(\mathbf{x}|\mu,\Sigma) d\mathbf{x}} \\
&= \frac{[\phi_{\mathbf{X}}(\mathbf{x}|\Theta) Q(\mathbf{x})]^2}{\int_{\mathscr{W}} [\phi_{\mathbf{X}}(\mathbf{x}|\Theta) Q(\mathbf{x})]^2 d\mathbf{X}} \\
&\triangleq p_1(\mathbf{x})
\end{aligned} \tag{5.24}$$

and

$$\begin{aligned}
\frac{f_{\mathbf{X}^*}(\mathbf{x}|\mu,\Sigma) f_{\mathbf{X}_T^*}(\mathbf{x}|\mu_T,\Sigma_T)}{c_2} &= \frac{f_{\mathbf{X}^*}(\mathbf{x}|\mu,\Sigma) f_{\mathbf{X}_T^*}(\mathbf{x}|\mu_T,\Sigma_T)}{\int_{\mathscr{W}} f_{\mathbf{X}^*}(\mathbf{x}|\mu,\Sigma) f_{\mathbf{X}_T^*}(\mathbf{x}|\mu_T,\Sigma_T) d\mathbf{x}} \\
&= \frac{\phi_{\mathbf{X}}(\mathbf{x}|\Theta) \phi_{\mathbf{X}}(\mathbf{x}|\Theta_T) Q(\mathbf{x})}{\int_{\mathscr{W}} \phi_{\mathbf{X}}(\mathbf{x}|\Theta) \phi_{\mathbf{X}}(\mathbf{x}|\Theta_T) Q(\mathbf{x}) d\mathbf{x}} \\
&\triangleq p_2(\mathbf{x})
\end{aligned} \tag{5.25}$$

Both these functions $p_1(\mathbf{x})$ and $p_2(\mathbf{x})$, are both PDFs defined on the workspace $\mathscr{W}$. Finally, the derivatives can be expressed by

$$\frac{\partial}{\partial \mu} U_{CS}(\mu, \Sigma) = \Sigma^{-1} \int_{\mathscr{W}} \left[ p_1(\mathbf{x}) - p_2(\mathbf{x}) \right] \mathbf{x} d\mathbf{x} \tag{5.26}$$

and

$$\frac{\partial}{\partial \mathbf{L}} U_{CS}(\mu, \Sigma) = - \int_{\mathscr{W}} \left[ p_1(\mathbf{x}) - p_2(\mathbf{x}) \right] \left[ (\mathbf{x} - \mu)(\mathbf{x} - \mu)^T \right] d\mathbf{x}\mathbf{L} \tag{5.27}$$

Similarly, these two derivatives with respect to $\mu$ and $\mathbf{L}$ can both be approximated based on the auxiliary samples, $\mathscr{D}_A$, such that

$$\frac{\partial}{\partial \mu} U_{CS}(\mu, \Sigma) \approx \Sigma^{-1} \sum_{j=1}^{M} \left[ \hat{p}_1(\zeta_j) - \hat{p}_2(\zeta_j) \right] \mathbf{x}_n \tag{5.28}$$

and

$$\frac{\partial}{\partial \mathbf{L}} U_{CS}(\mu, \Sigma) \approx \Sigma^{-1} \sum_{j=1}^{M} \left[ \hat{p}_1(\zeta_j) - \hat{p}_2(\zeta_j) \right] \left[ (\zeta_j - \mu)(\zeta_j - \mu)^T \right] \mathbf{L} \tag{5.29}$$

Here, the two functions, $\hat{p}_1(\zeta_j)$ and $\hat{p}_2(\zeta_j)$ are applied to approximate $p_1(\zeta_j)$ and $p_2(\zeta_j)$ respectively, such that

$$p_1(\zeta_j) \approx \hat{p}_1(\zeta_j) \triangleq \frac{[\phi_{\mathbf{X}}(\zeta_j|\Theta) Q(\zeta_j)]^2}{\sum_{j=1}^{M} [\phi_{\mathbf{X}}(\zeta_j|\Theta) Q(\zeta_j)]^2} \tag{5.30}$$

and

$$p_2(\zeta_i) \approx \hat{p}_2(\zeta_j) \triangleq \frac{\phi_{\mathbf{X}}(\zeta_j|\Theta) \phi_{\mathbf{X}}(\zeta_j|\Theta_T) Q(\mathbf{x})}{\sum_{j=1}^{M} \phi_{\mathbf{X}}(\zeta_j|\Theta) \phi_{\mathbf{X}}(\zeta_j|\Theta_T) Q(\zeta_j)} \tag{5.31}$$

The trajectory of the SND is generated from the initial distribution, $f_{\mathbf{X}^*}(\mathbf{x}|\Theta_0)$, until the target distribution $f_{\mathbf{X}^*}(\mathbf{x}|\Theta_T)$. Where the sequence of the parameters $\tau_\Theta = \{\Theta_0, ..., \Theta_T\}$, is generated by

$$\mu_{i+1} = \mu_i - \lambda_\mu \frac{\partial U_{CS}}{\partial \mu} \tag{5.32}$$

$$\mathbf{L}_{i+1} = \mathbf{L}_i - \lambda_{\mathbf{L}} \frac{\partial U_{CS}}{\partial \mathbf{L}} \tag{5.33}$$

$$\Sigma_{i+1} = \mathbf{L}_{i+1} \mathbf{L}_{i+1}^T \tag{5.34}$$

where the subscript $\tau$ indicates a time indice such that $\tau = 0, 1, ...T_f$.

## 5.2 PATH-PLANNING BASED ON SNMM

In this section, consider the PDF path-planning problem where the start distribution is described by a skewed normal mixture model (SNMM), $\varphi_{\mathbf{X}^*}$, defined in (3.15), such that

$$\varphi_{\mathbf{X}^*}(\mathbf{x}|\Theta_{1:N_c}) \triangleq \sum_{i=1}^{N_c} \omega_i f_{\mathbf{X}_i^*}(\mathbf{x}|\mu_i, \Sigma_i) \tag{5.35}$$

and the target distribution is still a SND, $f_{\mathbf{X}^*}(\mathbf{x}|\mu_T, \Sigma_T)$. Since there is only one SD component in the target distribution, we only need to find the PDF trajectories for each SN component $f_{\mathbf{X}^*}(\mathbf{x}|\mu_i, \Sigma_i)$ to the target SND $f_{\mathbf{X}^*}(\mathbf{x}|\mu_T, \Sigma_T)$.

### 5.2.1 PATH-PLANNING BASED ON DISPLACEMENT INTERPOLATION FOR SNMM

Given the skewing function $Q_Y$, the time-varying BRF-SNMM $\varphi(\mathbf{x}, t)$ is specified by the parameter set $\Theta_{1:N_c}(t) = (\omega_i, \mu_i(t), \Sigma_i(t))$, which can also specify a time-varying GMM, such that

$$g(\mathbf{x}, t) \triangleq \sum_{i=1}^{K} \omega_i \phi_{\mathbf{X},i}(\mathbf{x}|\mu_i(t), \Sigma_i(t)) \tag{5.36}$$

Similarly, the parameter set of the desired BRF-SN, $\Theta_f$, can specify a desired normal PDF, $g_f = \phi_{\mathbf{X}}(\mathbf{x}|\mu_f, \Sigma_f)$. Thus, it is reasonable to obtain a trajectory of the time-varying GMM from the start GMM, $g(\mathbf{x}, t_0)$, to the desired normal distribution, $g_f$, in the obstacle-free workspace first, then form the time-varying BRF-SNMM trajectory with the obtained time-varying parameter set, $\Theta_{1:N_c}(t)$. Now, we apply the displacement interpolation (DI) between $g(\mathbf{x}, t_0)$ and $g_f$ to specify the time-varying parameter set, $\Theta_i(t) = (\mu_i(t), \Sigma_i(t))$ for $i = 1, \ldots, N_c$ [16], such that

$$\mu_i(t) = \frac{t_f - t - t_0}{t_f - t_0}\mu_i(t_0) + \frac{t - t_0}{t_f - t_0}\mu_f \tag{5.37}$$

$$\Sigma_i(t) = \Sigma_i(t_0)^{-1/2}\left(\frac{t_f - t - t_0}{t_f - t_0}\Sigma_i(t_0) + \frac{t}{t - t_0}\cdot\left[\Sigma_i(t_0)^{1/2}\Sigma_f\Sigma_i(t_0)^{1/2}\right]\right)\Sigma_i(t_0)^{-1/2} \tag{5.38}$$

Although the obtained GMM trajectory, $g(\mathbf{x}, t|\Theta_{1:N_c}(t))$, is a geodesic path, which provides the shortest $\ell_2$ Wasserstein distance [17], it is not guaranteed that the obtained BRF-SNMM trajectory can give $\ell_2$ Wasserstein distance since the time-varying parameter set is obtained without considering the obstacles.

### 5.2.2 PATH-PLANNING BASED ON ARTIFICIAL POTENTIAL FOR SNMM

This section will be a continuity of **section** 5.1.2.

### POTENTIAL FIELD BASED L-2 NORM FOR BRF-SNMM

First, consider the potential is defined by

$$U_{SN}(\Theta_{1:N_c}) \triangleq \frac{1}{2} \int_{\mathscr{W}} \left[\varphi_{\mathbf{X}^*}(\mathbf{x}) - f_{\mathbf{X}^*}(\mathbf{x}|\mu_T, \Sigma_T)\right]^2 d\mathbf{x} \tag{5.39}$$

then, the derivative with respect to $\mu_i$ can be expressed by

$$\frac{\partial U_{SN}(\Theta_{1:N_c})}{\partial \mu_i} = -\Sigma^{-1} \int_{\mathscr{W}} \left[\varphi_{\mathbf{X}^*}(\mathbf{x}) - f_{\mathbf{X}_T^*}(\mathbf{x}|\mu_T, \Sigma_T)\right] f_{\mathbf{X}^*}(\mathbf{x}|\mu, \Sigma)] \left[\frac{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{x}}}[\mathbf{X}Q(\mathbf{X})]}{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{x}}}[Q(\mathbf{X})]} - \mathbf{x}\right] d\mathbf{x} \tag{5.40}$$

which, can be approximated based on the auxiliary samples, $\mathscr{D}_A$, such that

$$\begin{aligned} \frac{\partial(\Theta_{1:N_c})}{\partial \mu_i} U_{SN} &\approx \omega_i \Sigma^{-1} \sum_{j=1}^{M} \left[\varphi_{\mathbf{X}^*}(\zeta_j) - f_{\mathbf{X}_T^*}(\zeta|\Theta_T)\right] f_{\mathbf{X}^*}(\zeta_j|\Theta_i)(\zeta_j - \tilde{\mathbf{x}}_i)\Delta x \Delta y \\ &= \omega_i \Sigma^{-1} \sum_{j=1}^{M} \alpha_{i,j}(\zeta_j - \tilde{\mathbf{x}}_i) \end{aligned} \tag{5.41}$$

where, $\tilde{\mathbf{x}}_i$ is defined in (5.12)

$$\alpha_{i,j} = \left[\varphi_{\mathbf{X}^*}(\zeta_j) - f_{\mathbf{X}_T^*}(\zeta_j|\mu_T, \Sigma_T)\right] f_{\mathbf{X}^*}(\zeta_j|\mu_i, \Sigma_i)\Delta x \Delta y \tag{5.42}$$

similarly, the derivative with respect to $\mathbf{L}_i$ can be expressed by

$$\begin{aligned} \frac{\partial}{\partial \mathbf{L}} U_{SN}(\Theta_{1:N_c}) = \omega_i \int_{\mathscr{W}} &\left[\varphi_{\mathbf{X}^*}(\mathbf{x}) - f_{\mathbf{X}_T^*}(\mathbf{x}|\Theta)\right] .f_{\mathbf{X}_i^*}(\mathbf{x}|\Theta_i) \\ &. \left[(\mathbf{x} - \mu_i)(\mathbf{x} - \mu_i)^T - \frac{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{X}}}[(\mathbf{X} - \mu_i)(\mathbf{X} - \mu_i)^T Q(\mathbf{X})]}{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{X}}}[Q(\mathbf{X})]}\right] \mathbf{L}_i \end{aligned} \tag{5.43}$$

which, can be approximated based on the auxiliary samples, $\mathscr{D}_A$, such that

$$\begin{aligned}\frac{\partial}{\partial \mathbf{L}} U_{SN}(\Theta_{1:N_c}) &\approx \omega_i \sum_{j=1}^{M} \left[\varphi_{\mathbf{X}^*}(\zeta_j) - f_{\mathbf{X}_T^*}(\zeta_j|\Theta_T)\right] . f_{\mathbf{X}_i^*}(\zeta|\Theta_i) \left[(\zeta_j - \mu_i)(\zeta_j - \mu_i)^T - \tilde{\Sigma}_i\right] \mathbf{L}_i \Delta x \Delta y \\ &= \omega_i \left[\sum_{j=1}^{M} \tilde{\alpha}_{i,j}(\zeta_j - \mu_i)(\zeta_j - \mu_i)^T\right] \mathbf{L}_i\end{aligned} \tag{5.44}$$

where,

$$\tilde{\alpha}_{i,j} = \left(\sum_{j=1}^{M} \alpha_{i,j}\right) \frac{\phi_{\mathbf{X}}(\zeta_j|\Theta_i) Q(\zeta_j)}{\sum_{j=1}^{M} \phi_{\mathbf{X}}(\zeta_j|\Theta_i) Q(\zeta_j)} - \alpha_{i,j} \tag{5.45}$$

Furthermore, we can have the novel potential by considering the log operation, such that

$$U_{log_{SN}}(\Theta_{1:N_c}) \triangleq \frac{1}{2} ln \int_{\mathscr{W}} \left[\varphi_{\mathbf{X}^*}(\mathbf{x}) - f_{\mathbf{X}^*}(\mathbf{x}|\mu_T, \Sigma_T)\right]^2 d\mathbf{x} \tag{5.46}$$

Then, we can generate the derivatives based on the data samples, such that

$$\frac{\partial(\Theta_{1:N_c})}{\partial \mu} U_{log_{SN}} \approx \omega_i \Sigma^{-1} \sum_{j=1}^{M} \kappa_{i,j}(\zeta_j - \tilde{\mathbf{x}}_n) \tag{5.47}$$

where, $\alpha_{i,j}$ defined in (5.42) and

$$\kappa_{i,j} = \frac{\alpha_{i,j}}{\sum_{j=1}^{M} \left[\varphi_{\mathbf{X}^*}(\zeta_j) - f_{\mathbf{X}_T^*}(\zeta_j|\Theta_T)\right]^2 \Delta x \Delta y} \tag{5.48}$$

also,

$$\frac{\partial(\Theta_{1:N_c})}{\partial \mathbf{L}} U_{log_{SN}} \approx \omega_i \left[\sum_{j=1}^{M} \tilde{\kappa}_{i,j}(\zeta_j - \mu_i)(\zeta_j - \mu_i)^T\right] \mathbf{L}_i \tag{5.49}$$

where, $\tilde{\alpha}_{i,j}$ defined in (5.45) and

$$\tilde{\kappa}_{i,j} = \frac{\tilde{\alpha}_{i,j}}{\sum_{j=1}^{M} \left[\varphi_{\mathbf{X}^*}(\zeta_j) - f_{\mathbf{X}_T^*}(\zeta_j|\Theta_T)\right]^2 \Delta x \Delta y} \tag{5.50}$$

also, the artificial potential for the path-planning from the current SNMM, $\varphi_{\mathbf{X}^*}$ to the target SND, $f_{\mathbf{X}^*}(\mathbf{x}|\Theta_T)$, can be expressed by

$$U_{CS}(\varphi_{\mathbf{X}^*}, f_{\mathbf{X}_T^*}) \triangleq \omega_i \sum_{i=1}^{N_c} U_{CS}\left( f_{\mathbf{X}_i^*}(\mathbf{x}|\Theta_{1:N_c}) || f_{\mathbf{X}_T^*}(\mathbf{x}|\Theta_{1:N_c}) \right) \tag{5.51}$$

where the component weights $\omega = [\omega_1, ..., \omega_i, ..., \omega_{N_c}]$, are fixed when the SNMM, $\varphi_{\mathbf{X}^*}$, is given for $i = 1, ..., N_c$. Furthermore, we notice that the potential, $U_{SN}(t)$, provides larger force when the agents' distribution is close to the desired distribution due to the logarithm function, while the potentla, $U_{CS}(t)$ can contribute larger force at the initial stage of the distribution path-planning problem. Therefore, we combine the two potentials to create a novel potential to generate the trajectory of agents' distributions, such that

$$U(t) \triangleq \lambda_{SN} U_{SN}(t) + \lambda_{CS} U_{CS}(t) \tag{5.52}$$

where, $\lambda_{SN}, \lambda_{CS} \in \mathbb{R}^+$ are user defined parameters.

## 5.3 PATH-PLANNING FOR GAUSSIAN MIXTURE MODEL

Considering the GMM can be treated as a special SNMM where the skewing function is always set to 1. Then the potentials, $U_{SN}$ and $U_{CS}$ can be applied directly by substituting $Q(\mathbf{X}) = 1$. The GMM defined in (3.3) can also be expressed as

$$\varphi_{\mathbf{X}}(\mathbf{x}|\Theta_{1:N_c}) = \sum_{i=1}^{N_c} \omega_i \phi_{\mathbf{X}}(\mathbf{x}|\mu_i, \Sigma_i) \tag{5.53}$$

Furthermore, because the skewing function is independent with $\mu$ and $\mathbf{L}$, the corresponding derivatives can be easily obtained based on the auxiliary samples, $\mathscr{D}_A$, such that

$$\frac{\partial}{\partial \mu} U_{SN}(\Theta_{i:N_c}) \approx \omega_i \Sigma^{-1} \sum_{j=1}^{M} \left[ \varphi_{\mathbf{X}}(\zeta_j) - \phi_{\mathbf{X}_T^*}(\zeta|\Theta_T) \right] \phi_{\mathbf{X}}(\zeta_j|\Theta_i)(\zeta_j - \mu_i) \Delta x \Delta y \tag{5.54}$$

and

$$\frac{\partial}{\partial \mathbf{L}_i} U_{SN}(\Theta_{1:N_c}) \approx \omega_i \sum_{j=1}^{M} \Big[\varphi_{\mathbf{X}}(\zeta_j) - \phi_{\mathbf{X}_T^*}(\zeta_j|\Theta_T)\Big] .\phi_{\mathbf{X}_i^*}(\zeta|\Theta_i) \left[(\zeta_j - \mu_i)(\zeta_j - \mu_i)^T - \Sigma_i\right] \mathbf{L}_i \Delta x \Delta y \tag{5.55}$$

Similarly, we generate the for CS-divergence for the GMM, based on the auxiliary samples $\mathscr{D}_A$ such that

$$U_{CS}(\varphi_{\mathbf{X}}, \phi_{\mathbf{X}_T^*}) \triangleq \omega_i \sum_{i=1}^{N_c} U_{CS}\left(\phi_{\mathbf{X}_i}(\mathbf{x}|\Theta_{1:N_c})||\phi_{\mathbf{X}_T}(\mathbf{x}|\Theta_{1:N_c})\right) \tag{5.56}$$

Since the obstacle information is not embedded in the GMM distributions, a repulsive potential against the skewing function $Q(\mathbf{X})$, is applied

$$\begin{aligned} U_{Rep}(\Theta_{1:N_c}) &= \int_{\mathscr{W}} \varphi_{\mathbf{X}}(\mathbf{x}|\Theta_{1:N_c}[1 - Q(\mathbf{x})]d\mathbf{x} \\ &= \sum_{i=1}^{N_c} \omega_i \int_{\mathscr{W}} \phi_{\mathbf{X}}(\mathbf{x}|\Theta_{1:N_c})[1 - Q(\mathbf{x})]d\mathbf{x} \end{aligned} \tag{5.57}$$

Similarly, the derivative of the repulsive potential with respect to $\mu_i$ and $\Sigma_i$, based on the auxiliary data samples, $\mathscr{D}_A$, can be expressed by

$$\frac{\partial U_{Rep}(\Theta_{1:N_c})}{\partial \mu_i} = \omega_i \sum_{j=1}^{M} \phi_{\mathbf{X}}(\zeta_j|\mu_i, \Sigma_i)(\zeta_j - \mu_i)[1 - Q(\zeta_j)]\Delta x \Delta y \tag{5.58}$$

and

$$\frac{\partial U_{Rep}(\Theta_{1:N_c})}{\partial \mathbf{L}_i} = -\omega_i \sum_{j=1}^{M} \phi_{\mathbf{X}}(\zeta_j|\mu_i, \Sigma_i) \left[(\zeta_j - \mu_i)(\zeta_j - \mu_i)^T - (\mathbf{L}_i \mathbf{L}_i^T)^{-1}\right] .\mathbf{L}_i[1 - Q(\zeta_j)]\Delta x \Delta y \tag{5.59}$$

Furthermore, let's consider the logarithm version of (5.57), such that

$$U_{log_{Rep}}(\Theta_{1:N_c}) = ln \sum_{i=1}^{N_c} \omega_i \int_{\mathscr{W}} \phi_{\mathbf{X}}(\mathbf{x}|\Theta_{1:N_c})[1 - Q(\mathbf{x})]d\mathbf{x} \tag{5.60}$$

then, the derivatives of the repulsive potential with respect to $\mu_i$ and $\Sigma_i$, can be expressed by

$$\frac{\partial U_{log_{Rep}}(\Theta_{1:N_c})}{\partial \mu_i} \approx \omega_i \Sigma_i^{-1} \sum_{j=1}^{M} \beta_{i,j}(\zeta_j - \mu_i) \tag{5.61}$$

and

$$\frac{\partial U_{log_{Rep}}(\Theta_{1:N_c})}{\partial \mathbf{L}_i} \approx \omega_i \Sigma_i^{-1} \sum_{j=1}^{M} \beta_{i,j} \left[ (\zeta_j - \mu_i)(\zeta_j - \mu_i)^T - (\mathbf{L}_i \mathbf{L}_i^T)^{-1} \right] \mathbf{L}_i \tag{5.62}$$

where

$$\beta_{i,j} = \frac{\phi_{\mathbf{X}}(\zeta_j | \mu_i, \Sigma_i)[1 - Q(\zeta_j)]}{\sum_{j=1}^{M} \varphi_{\mathbf{X}}(\zeta_j | \Theta_{1:N_c})[1 - Q(\zeta_j)]} \tag{5.63}$$

To finally conclude the total artificial potential field for the GMM can be expressed as

$$U_{GMM}(t) \triangleq \lambda_{SN} U_{SN}(t) + \lambda_{CS} U_{CS}(t) + \lambda_{Rep} U_{log_{Rep}}(t) \tag{5.64}$$

where, $\lambda_{SN}, \lambda_{CS}, \lambda_{Rep} \in \mathbb{R}^+$ are all user defined parameters.

# CHAPTER 6

# SIMULATIONS AND RESULTS

Two experiments are provided in this section to demonstrate the effectiveness of the proposed parameter learning algorithms proposed in **Chapter 4** and the BRF-SNMM based path-planning approach in **Chapter 5**.

## 6.1 LEARNING PARAMETER COST FUNCTION COMPARISON

To evaluate the learning ability of **Algorithm**1, $N = 300$ samples, $\mathscr{D} = \{\mathbf{x}_n\}_{n=1}^{N}$, are generated according to the BRF-SN distribution, $f_{\mathbf{X}^*}(\mathbf{x}|\mu, \Sigma)$, and deployed with the obstacle, as shown in **Figure 1**. For simplicity, a binary skewing function, $Q_Y(\mathbf{x}) \in \{0, 1\}$, is applied to indicate if the obstacle in **Figure 1** occupies the location. The parameters of the BRF-SN are set to $\mu = [10 \quad 12]$ and $\Sigma = \begin{bmatrix} 1 & 0.3 \\ 0.3 & 0.7 \end{bmatrix}$. although the sample data set, $\mathscr{D}$, is generated according to the BRF-SN, only the sample data set are available to estimate the parameters. Thus, the samples are assumed to be generated by a BRF-SNMM, and the parameters, $N_C$ and $\Theta_{1:N_C} = \{(\omega_i, \Theta_i)\}_{i=1}^{N_C}$, all need to be estimated. In this experiment, three values of $N_C$ are

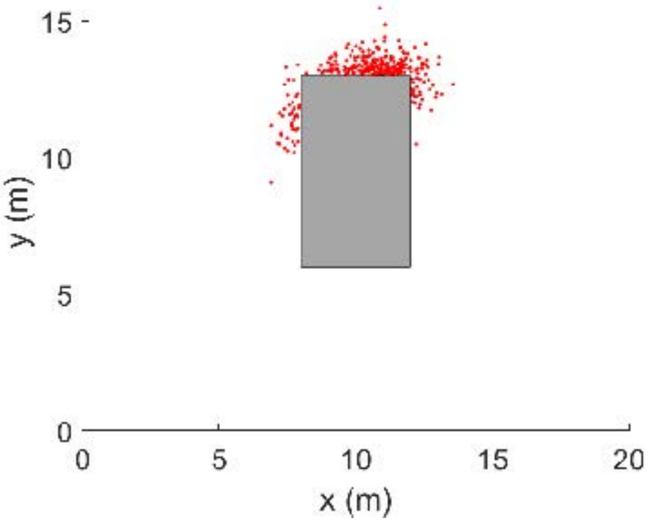


Figure 1: Samples ($N = 300$) generated according to the BRF-SN distribution are deployed with the obstacle in the workspace. The red points indicate the samples, and the gray rectangle indicates the obstacle.

considered, $N_C = 1, 2, 3, 4, 5, 6$ to run **Algorithm** 1, respectively. The learning performance of **Algorithm**1 is compared with the other two approaches.

- GMM approach: assume that the samples are generated by the GMM and estimate the GMM's parameters from $\mathscr{D}$.

- SNMM-GMM approach: assume that the samples are generated by BRF-SNMM but the parameters $\Theta_{1:N_C}$ are estimated using the GMM's parameter learning algorithm. For a fair comparison, in **Algorithm** 1, the SNMM is also initialized by GMM's parameters.

The NLLs of the samples generated by different approaches are plotted in **Figure 2** (Top). It can be seen that only our algorithm can provide the smallest NLL for $N_C = 1$, which shows that up to 6 components are requested to describe the samples if we use GMM or SNMM-GMM approaches to estimate the parameters, respectively. Furthermore, the CS-divergences between the ground-truth distribution and the estimated distributions are plotted in **Figure 2** (Bottom), where the dash curves represent the CS-divergence between the ground-truth PDF and the approximated PDFs from samples via Kernel Density Estimate method. It also shows that our algorithm can provide the best performance.

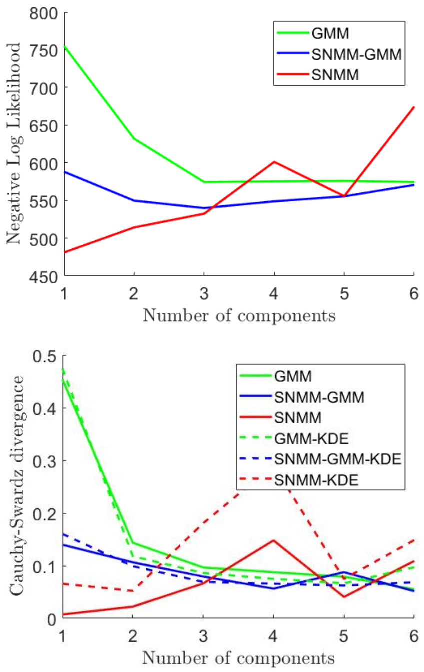


Figure 2: Comparison of parameter learning performances. (top) The NLL for different approaches and different values of $N_C$, and (bottom) the CS-divergences between the ground-truth distribution and the estimated distributions (solid curves) and the CS-divergences between the groundtruth PDF and the approximated PDFs from samples via the KDE method (dash curves).

### 6.2 SIMULATION OF MULTI-AGENT PATH-PLANNING BASED BRF-SNMM

To evaluate the proposed path planning algorithms, first we create a simple artificial environment (Forest-I) shown in **Figure 3** (a) left, where 14 black circles indicate the trees in the artificial forest. In this experiment, a VLSR system compromised of $N = 300$ agents is tested. Given the agents' initial positions in the artificial forest, the red points in **Figure 3** (a) Left, the goal of the path-planning task is to guide all of the agents to traverse the artificial forest and avoid collisions with the obstacles "trees", while achieving the target distribution, $\varphi_f$, showing in **Figure 3** (b) Right. The white circles shown in **Figure 3** (b) Right indicated the area s where $Q_Y = 0$, which refers to a location in the area that agents are not allowed to occupy because of obstacles. The SNMM-DI and the SNMM-APF path-planing algorithms are first executed to generate the trajectories of the agents' PDFs, where $N_C$ is set to 2. The snapshots of trajectories of agents and corresponding PDFs generated by SNMM-DI and SNMM-APF path-planning algorithms are presented in **Figure 4** and **Figure 5**, respectively. These snapshots show that both path-planning algorithms can provide acceptable performance, altough different trajectories of agents' PDFs are generated. It can also be observed that the time-varying agents' distributions are split by the obstacles due to the skewing function, $Q_Y$, which is the expected advantage of the SNMM over GMM. To demonstrate the advantage of SNMM over GMM in path-planning problems, a GMM-based approach is also applied to the same problem for comparison. Because the GMM is a special case of the SNMM where $Q_Y(\mathbf{X}) \equiv 1$ according to (5.3) for all $\mathbf{x} \in \mathscr{W}$. Since the obstacles information is not embedded in the GMM distributions, $\varphi(\mathbf{x}, t)$, a repulsive potential against skewing function is added to the potential defined in (5.57), such that

$$U_{Rep}(t|\Theta_{1:N_C}) = \int_{\mathscr{W}} \varphi_{\mathbf{X}}(\mathbf{x}|\Theta_{1:N_C})[1 - Q(\mathbf{x})]d\mathbf{x} \tag{6.1}$$

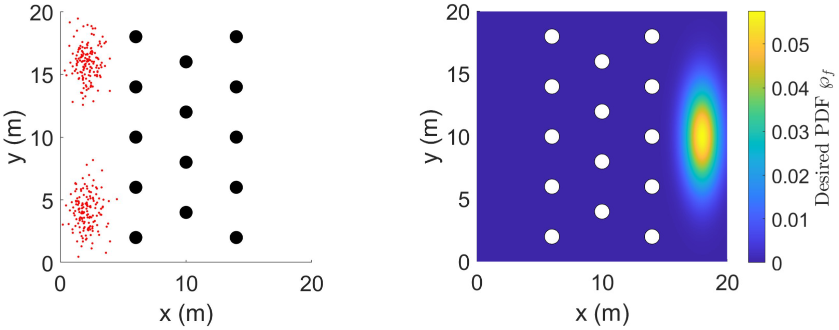
Where the repulsive potential is zero-ed if $Q_Y(\mathbf{x}) = 1$ for all $\mathbf{x} \in \mathscr{W}$, which represents the no-obstacle scenario. The range of the repulsive potential is $U_{rep} \in [0, 1]$. Thus, the potential for the GMM-based approach can be expressed by

$$U_{GMM}(t) \triangleq \lambda_{SN} U_{SN}[t|Q(\mathbf{x} \equiv 1)] + \lambda_{CS} U_{CS}[t|Q(\mathbf{x} \equiv 1)] + \lambda_{Rep} U_{log_{Rep}}[t|Q(\mathbf{x} \equiv 1)] \tag{6.2}$$

Finally, similarly to the SNMM-APF, the trajectory of agents' PDFs is obtained by minimizing the potential, $U_{GMM}(t)$ with repsect to the parameters denoted in (5.61) and (5.62). For distinction, this GMM-based approach is referred to as the GMM-APF approach. The same user-designed parameter as the SNMM-APF are applied to the GMM-APF, for a fair comparison,and the parameter $\lambda_{Rep}$ is set to 5%. The snapshots of trajectories of agents and corresponding PDFs generated by GMM-APF path-planning algorithms are presented in **Figure 6**.

Table 1: Performance Comparison

| Environment | Method | Time (min) | Length (m) |
|---|---|---|---|
| ForestI | SNMM-DI | 1.55 | $19.75 \pm 0.92$ |
|  | SNMM-APF | 54.69 | $27.17 \pm 1.74$ |
|  | GMM-APF | 64.25 | $33.46 \pm 2.58$ |
| ForestII | SNMM-DI | 1.55 | $22.68 \pm 1.25$ |
|  | SNMM-APF | 57.18 | $30.15 \pm 1.56$ |

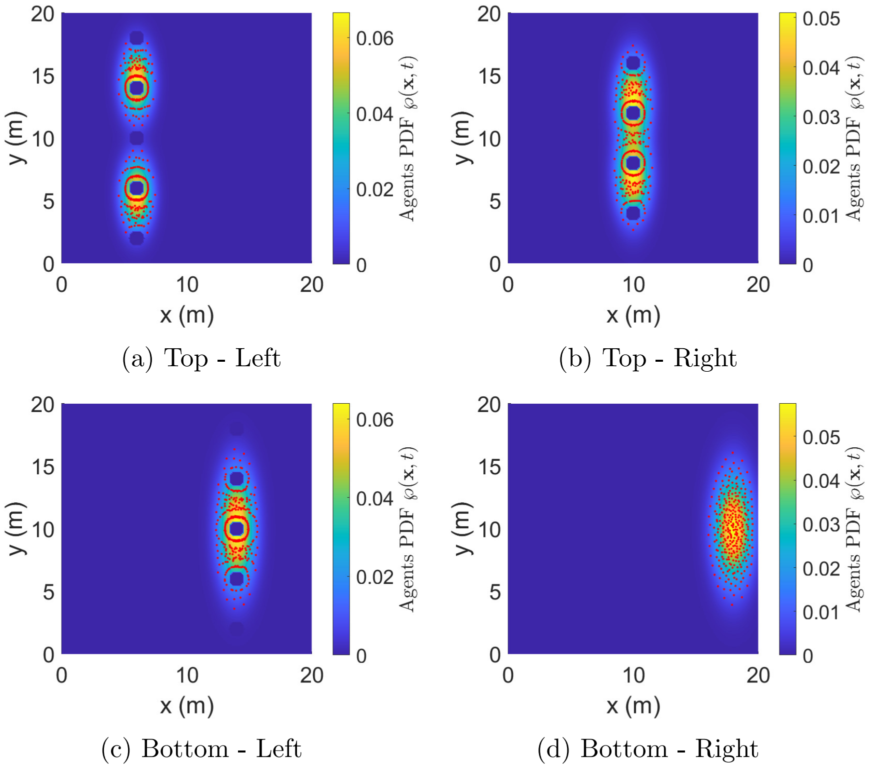


(a) Left

(b) Right

Figure 3: The path-planning problem for VLSR systems in the artificial forest environment. (a) Left - The red points indicate the initially deployed agents, and the black circles represent the "trees" in the forest. (b) Right - The desired agents' PDF is shown, where the white areas are occupied by "trees".

The numerical comparison results are tabulated in **Table 1** denoted by the experiment of ForestI. All of the simulations are conducted by MATLAB on the same computer. The column "Time" indicates the computational time for obtaining the optimal agents' PDFs. The column "Length" indicated the static results of the length $N = 300$ agents' trajectories in the form of "mean ± standard deviation". It can be seen that the proposed two SNMM-based approaches can outperform the GMM-based approach in terms of computational time and trajectory length. In addition, because the the SNMM-DI approach does not need to solve the optimization problem, it can obtain the agents' PDFs in a short period. However, it is also noteworthy that the absence of optimization in the SNMM-DI approach will result in a non-continuous OSD trajectory when the VLSR system travels a huge obstacle. Furethermore, to demonstrate the effectiveness of the proposed SNMM-based approaches for the cluttered environments, a complex artifical forest environments (ForestII) is also considered, where 50 smaller circles are deployed randomly in the workspace to represent the trees in a forest. Then, assuming the same agents' initial and the desired ditributions, we re-conduct the simulation. Results from the simulation show success using the proposed SNMM-based approaches, while the GMM-based approach fails, when only two components $N_C = 2$ are applied. The agents' trajectories generated by the two SNMM-based approaches are plotted in both **Figure 8** and **Figure 9**.



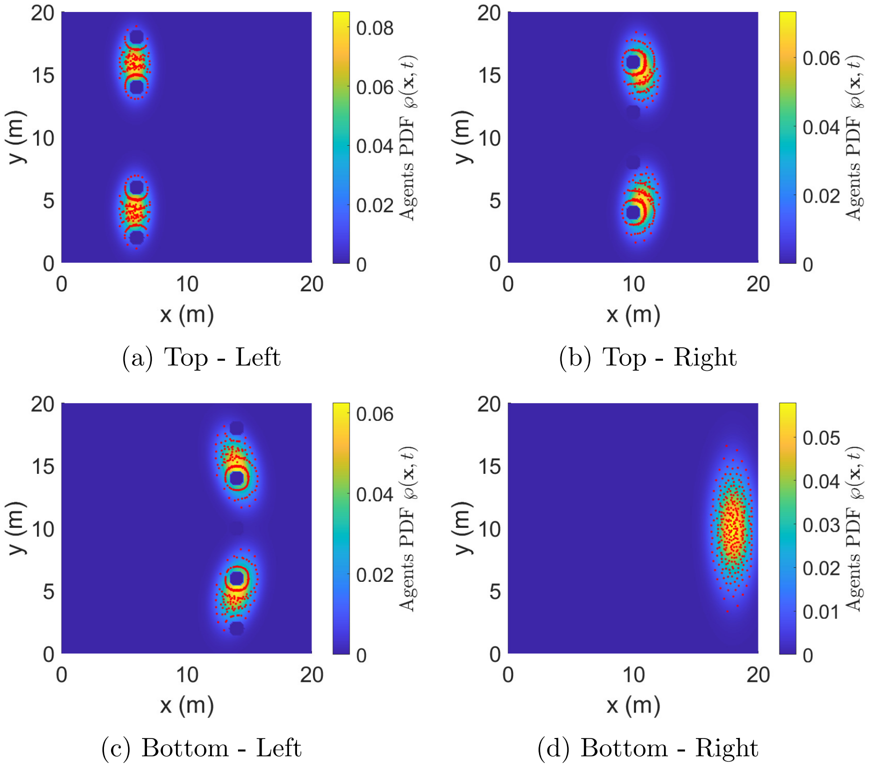
(a) Top - Left

(b) Top - Right

(c) Bottom - Left

(d) Bottom - Right

Figure 4: Snapshots of the trajectory of agents and corresponding PDFs generated by the SNMM-DI approach.



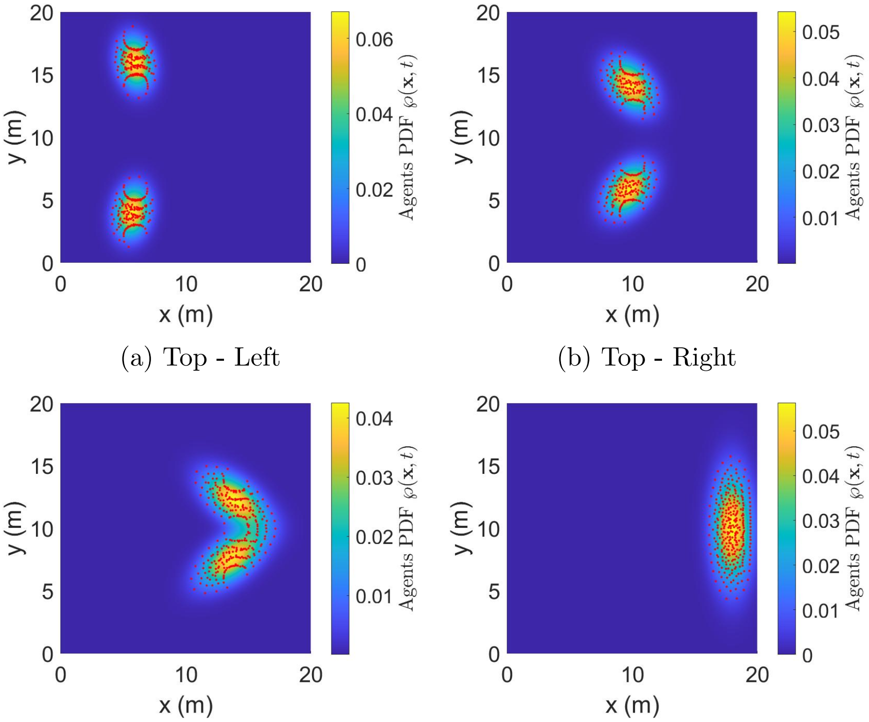
(a) Top - Left

(b) Top - Right

(c) Bottom - Left

(d) Bottom - Right

Figure 5: Snapshots of the trajectory of agents and corresponding PDFs generated by the SNMM-APF approach.



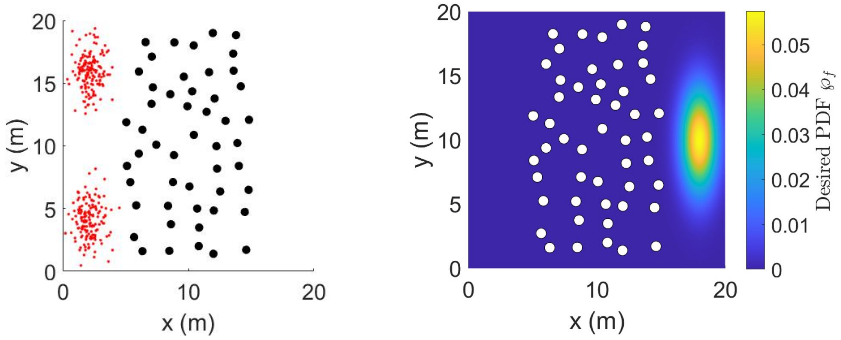
(a) Top - Left

(b) Top - Right

(c) Bottom - Left

(d) Bottom - Right

Figure 6: Snapshots of the trajectory of agents and corresponding PDFs generated by the GMM-APF approach.



(a) Left

(b) Right

Figure 7: The path-planning problem for VLSR systems in the complex artificial forest environment. (a) Left - The red points indicate the initially deployed agents, and the black circles represent the "trees" in the forest. (b) Right - The desired agents' PDF is shown, where the white areas are occupied by "trees".

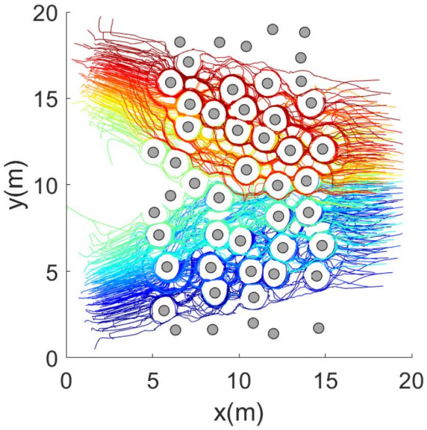


Figure 8: Trajectories of agents in different colors generated by the SNMM-DI approach in the Forest-II simulation.

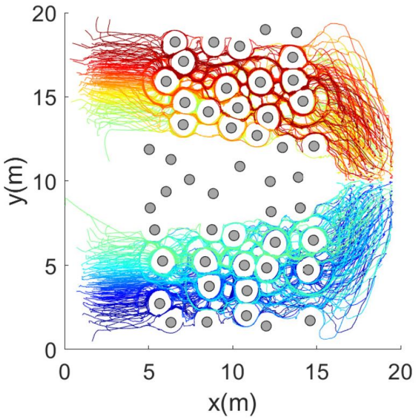


Figure 9: Trajectories of agents in different colors generated by the SNMM-APF approach in the Forest-II simulation.

# CHAPTER 7

# CONCLUSION AND FUTURE WORK

## 7.1 CONCLUSION AND DISCUSSION

The two proposed novel Bernoulli-random-field based multivariate skew-normal distribution and the skew-normal mixture model, represent the macroscopic states of the VLSR systems. In addition, a parameter learning algorithm is provided to estimate the model parameters for the BRF-SNMM. Furthermore, two path-planning algorithms are also developed based on the SNMM to guide the VLSR systems to traverse cluttered environments. The simulations in the artificial forest environments demonstrate the effectiveness of these two path-planning algorithms and the superiority of the SNMM-based approaches over the GMM-based approach, especially in cluttered environments.

## 7.2 FUTURE WORK

The proposed BRF-SNMM provides a novel research direction for VLSR system problems in cluttered environments, including robotic swarm path planning and tracking. More research on BRF-SNMM is requested, including generating microscopic states of the agents within the macroscopic distributions and adopting an infromation-sharing communication strategy to optimize energy consumption and model agents optimal trajectory in an unknown environment.

# REFERENCES

[1] Li, Bingxi, Sharvil Patankar, Barzin Moridian, and Nina Mahmoudian. *Planning large-scale search and rescue using team of uavs and charging stations.* In 2018 IEEE international symposium on safety, security, and rescue robotics (SSRR), pp. 1-8. IEEE, 2018.

[2] Tadokoro, Satoshi, ed. *Rescue robotics: DDT project on robots and systems for urban search and rescue.* Springer Science & Business Media, 2009.

[3] Steinberg, Marc, Jason Stack, and Terri Paluszkiewicz, *Long duration autonomy for maritime systems: Challenges and opportunities*, Autonomous Robots, MR vol. 40, no. 7, pp. 1119–1122, 2016.

[4] Hüttenrauch, Maximilian, Sosic Adrian, and Gerhard Neumann, *Deep reinforcement learning for swarm systems,* Journal of Machine Learning Research, vol. 20, no. 54, pp. 1–31, 2019.

[5] Hernandez-Leal, Pablo, Bilal Kartal, and Matthew E. Taylor, *A survey and critique of multiagent deep reinforcement learning,* Autonomous Agents and Multi-Agent Systems, vol. 33, no. 6, pp. 750–797, 2019.

[6] Foderaro, Greg, Silvia Ferrari, and Thomas A. Wettergren, *Distributed optimal control for multi-agent trajectory optimization,* Automatica, vol. 50, no. 1, pp. 149–154, 2014.

[7] Ferrari, Silvia, Greg Foderaro, Pingping Zhu, and Thomas A. Wettergren, *Distributed optimal control of multiscale dynamical systems: a tutorial,* IEEE Control Systems Magazine, vol. 36, no. 2, pp. 102–116, 2016.

[8] Rudd, Keith, Greg Foderaro, Pingping Zhu, and Silvia Ferrari, *A generalized reduced gradient method for the optimal control of very-large-scale robotic systems,* IEEE transactions on robotics, vol. 33, no. 5, pp. 1226– 1232, 2017.

[9] Zhu, Pingping, Chang Liu, and Silvia Ferrari, *Adaptive online distributed optimal control of very-large-scale robotic systems,* IEEE Transactions on Control of Network Systems, vol. 8, no. 2, pp. 678–689, 2021.

[10] Do, Chuong B. *The multivariate Gaussian distribution.* Section Notes, Lecture on Machine Learning, CS 229, 2008.

[11] Soria, Enrica, Fabrizio Schiano, and Dario Floreano, *Distributed predictive drone swarms in cluttered environments,* IEEE Robotics and Automation Letters, vol. 7, no. 1, pp. 73–80, 2021.

[12] Azzalini, Adelchi, *A class of distributions which includes the normal ones,* Scandinavian journal of statistics, pp. 171–178, 1985.

[13] Azzalini, Adelchi and A. Dalla Valle, *The multivariate skew-normal distribution,* Biometrika, vol. 83, no. 4, pp. 715–726, 1996.

[14] Gupta, Arjun K., Graciela González-Farıas, and J. Armando Domınguez-Molina, *A multivariate skew normal distribution,* Journal of multivariate analysis, vol. 89, no. 1, pp. 181–190, 2004.

[15] Arellano-Valle, Reinaldo B., and Marc G. Genton, *On fundamental skew distributions,* Journal of Multivariate Analysis, vol. 96, no. 1, pp. 93–116, 2005.

[16] Ramos, Fabio, and Lionel Ott, *Hilbert maps: Scalable continuous occupancy mapping with stochastic gradient descent,* The International Journal of Robotics Research, vol. 35, no. 14, pp. 1717–1730, 2016.

[17] Morelli, Julian, Pingping Zhu, Bryce Doerr, Richard Linares, and Silvia Ferrari, *Integrated mapping and path planning for very large-scale robotic (vlsr) systems,* in 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019, pp. 3356–3362.

[18] Zhu, Pingping, Silvia Ferrari, Julian Morelli, Richard Linares, and Bryce Doerr, *Scalable gas sensing, mapping, and path planning via decentralized hilbert maps,* Sensors, vol. 19, no. 7, p. 1524, 2019.

[19] Durrett, Rick, *Probability: theory and examples.* Cambridge university press, 2019, vol. 49.

[20] Chen, Yongxin, Tryphon T. Georgiou, and Allen Tannenbaum, *Optimal transport for gaussian mixture models,* IEEE Access, vol. 7, pp. 6269–6278, 2018.

[21] Kampa, Kittipat, Erion Hasanbelliu, and Jose C. Principe, *Closed-form cauchyschwarz pdf divergence for mixture of gaussians,* in The 2011 International Joint Conference on Neural Networks. IEEE, 2011, pp. 2578–2585.

[22] Petersen, Kaare Brandt, and Michael Syskind Pedersen, *The Matrix Cookbook*, 11-2012.

[23] Rue, Havard, and Leonhard Held. *Gaussian Markov Random Fields: Theory and Applications*, Chapman and Hall/CRC. 2005.

[24] Michael Nielsen, *Neural Networks and Deep Learning*, Determination Press, 2015.

[25] Dutilleul, Pierre. *The MLE algorithm for the matrix normal distribution.* Journal of statistical computation and simulation 64, no. 2 (1999): 105-123.

[26] Do, Chuong B., and Serafim Batzoglou. *What is the expectation maximization algorithm?.* Nature biotechnology 26, no. 8 (2008): 897-899.

[27] Chen, Yong-bo, Guan-chen Luo, Yue-song Mei, Jian-qiao Yu, and Xiao-long Su. *UAV path planning using artificial potential field method updated by optimal control theory.* International Journal of Systems Science 47, no. 6 (2016): 1407-1420.

[28] Do, Chuong B., and Serafim Batzoglou. *What is the expectation maximization algorithm?.* Nature biotechnology 26, no. 8 (2008): 897-899.

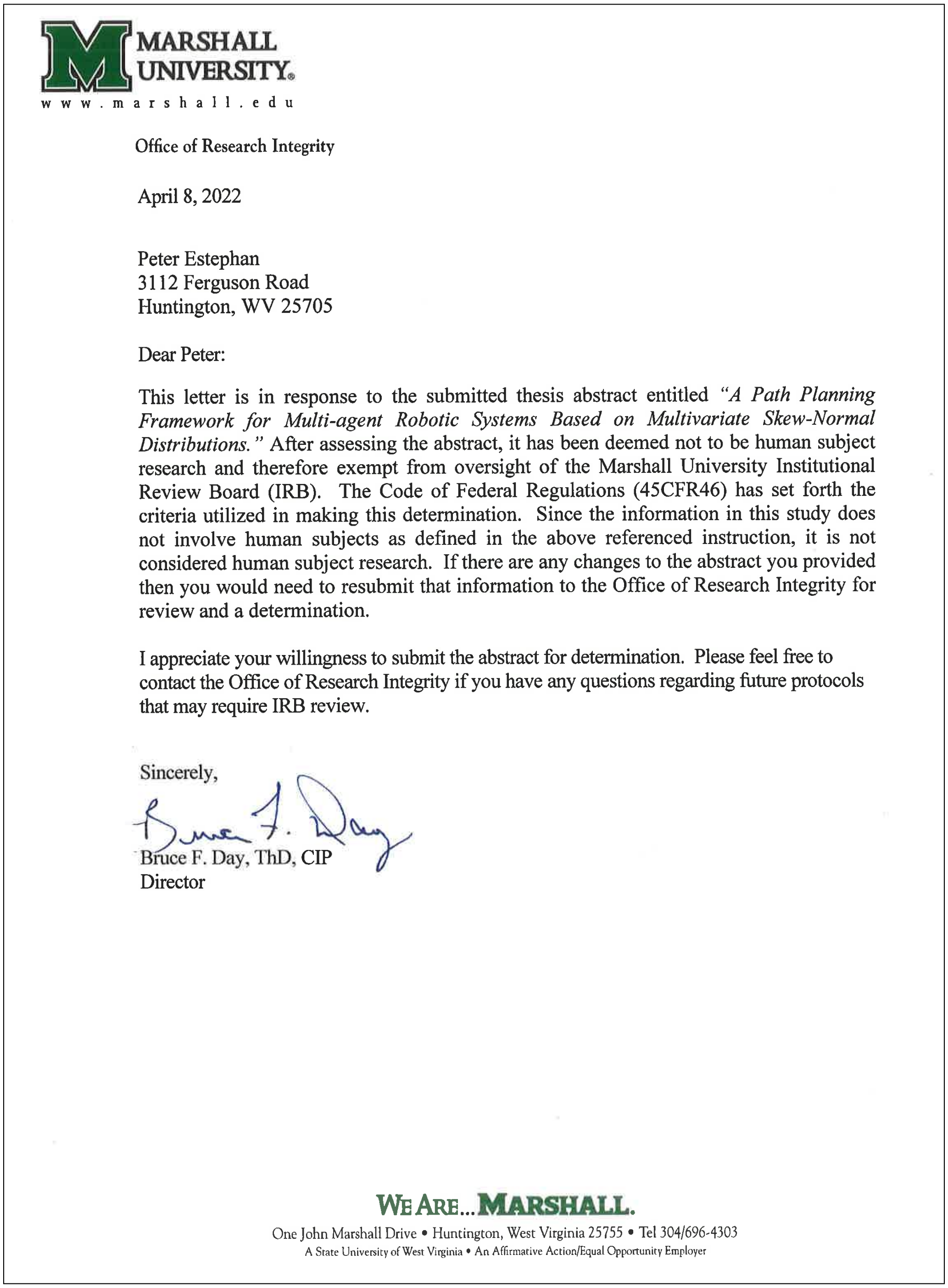
[29] Verdú, Sergio. *Total variation distance and the distribution of relative information.* In 2014 Information Theory and Applications Workshop (ITA), pp. 1-3. IEEE, 2014.

[30] Buchner, Johannes. *An intuition for physicists: information gain from experiments.* (2022-04-29)

[31] Nielsen, Frank. *On a generalization of the Jensen–Shannon divergence and the Jensen–Shannon centroid.* Entropy 22, no. 2 (2020): 221.

# APPENDIX A

# IRB APPROVAL LETTER



MARSHALL UNIVERSITY®
www.marshall.edu

Office of Research Integrity

April 8, 2022

Peter Estephan
3112 Ferguson Road
Huntington, WV 25705

Dear Peter:

This letter is in response to the submitted thesis abstract entitled *"A Path Planning Framework for Multi-agent Robotic Systems Based on Multivariate Skew-Normal Distributions."* After assessing the abstract, it has been deemed not to be human subject research and therefore exempt from oversight of the Marshall University Institutional Review Board (IRB). The Code of Federal Regulations (45CFR46) has set forth the criteria utilized in making this determination. Since the information in this study does not involve human subjects as defined in the above referenced instruction, it is not considered human subject research. If there are any changes to the abstract you provided then you would need to resubmit that information to the Office of Research Integrity for review and a determination.

I appreciate your willingness to submit the abstract for determination. Please feel free to contact the Office of Research Integrity if you have any questions regarding future protocols that may require IRB review.

Sincerely,

Bruce F. Day

Bruce F. Day, ThD, CIP
Director

WE ARE... MARSHALL.

# APPENDIX B

## DERIVATIVE OF PARTIAL DERIVATIVES IN (4.13) and (4.14)

In this appendix, some matrix derivative properties are applied. First consider the partial derivative with respect to $u$, which can be expressed by

$$\begin{aligned}
\frac{\partial}{\partial\mu}\mathscr{L}(\mathbf{x}_i,\mu,\Sigma) &= -\frac{\partial}{\partial\mu}\ln\left[\phi_{\mathbf{X}}(\mathbf{x}_i|\mu,\Sigma)\right] + \frac{\partial}{\partial\mu}\ln\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{X}}}[Q(\mathbf{X})] \\
&= -\frac{\partial}{\partial\mu}\ln\left(\frac{1}{2\pi|\Sigma|^{1/2}}\exp\left[-\frac{1}{2}(\mathbf{x}_i-\mu)^T\Sigma^{-1}(\mathbf{x}_i-\mu)\right]\right) + \frac{\partial}{\partial\mu}\ln\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{X}}}[Q(\mathbf{X})] \\
&= \frac{1}{2}\frac{\partial}{\partial\mu}\left((\mathbf{x}_i-\mu)^T\Sigma^{-1}(\mathbf{x}_i-\mu)\right) + \frac{1}{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{X}}}[Q(\mathbf{X})]}\int\frac{\partial\phi_{\mathbf{X}}(\mathbf{x})}{\partial\mu}Q(\mathbf{X})d\mathbf{x} \\
&= \Sigma^{-1}(\mu-\mathbf{x}_i) - \Sigma^{-1}\frac{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{X}}}[(\mu-\mathbf{x})Q(\mathbf{X})]}{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{X}}}[Q(\mathbf{X})]} \\
&= \Sigma^{-1}\left(\frac{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{X}}}[\mathbf{X}Q(\mathbf{X})]}{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{X}}}[Q(\mathbf{X})]} - \mathbf{x}_i\right)
\end{aligned} \tag{B.1}$$

Then consider the other partial derivative with respect to $\Sigma$, which can be expressed by

$$\begin{aligned}
\frac{\partial}{\partial\Sigma}\mathscr{L}(\mathbf{x}_i,\mu,\Sigma) &= -\frac{\partial}{\partial\Sigma}\ln\phi_{\mathbf{X}}(\mathbf{x}_i|\mu,\Sigma) + \frac{\partial}{\partial\Sigma}\ln\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{X}}}[Q(\mathbf{X})] \\
&= -\frac{\partial}{\partial\Sigma}\ln\left(\frac{1}{(2\pi|\Sigma|^{1/2}}\exp[-\frac{1}{2}(\mathbf{x}_i-\mu)^T\Sigma^{-1}(\mathbf{x}_i-\mu)]\right) + \frac{\partial}{\partial\Sigma}\ln\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{X}}}[Q(\mathbf{X})] \\
&= \frac{1}{2}\frac{\partial}{\partial\Sigma}\ln|\Sigma| + \frac{1}{2}\frac{\partial}{\partial\Sigma}\left((\mathbf{x}_i-\mu)^T\Sigma^{-1}(\mathbf{x}_i-\mu)\right) + \frac{1}{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{X}}}[Q(\mathbf{X})]}\int\frac{\partial\phi_{\mathbf{X}}(\mathbf{x})}{\partial\Sigma}Q(\mathbf{x})d\mathbf{x} \\
&= \frac{1}{2}\Sigma^{-1} - \frac{1}{2}\Sigma^{-1}(\mathbf{x}_i-\mu)(\mathbf{x}_i-\mu)^T\Sigma^{-1} + \frac{1}{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{X}}}[Q(\mathbf{X})]}\int\frac{\partial\phi_{\mathbf{X}}(\mathbf{x})}{\partial\Sigma}Q(\mathbf{x})d\mathbf{x}
\end{aligned} \tag{B.2}$$

Here the partial derivative $\frac{\partial\phi_{\mathbf{X}}(\mathbf{x})}{\partial\Sigma}$ can be expressed by

$$\begin{aligned}
\frac{\partial}{\partial\Sigma}\phi_{\mathbf{X}}(\mathbf{x}) &= \frac{\partial|\Sigma|^{-1/2}}{\partial\Sigma}\frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right)}{2\pi} + \frac{1}{2\pi|\Sigma|^{1/2}}\frac{\partial}{\partial\Sigma}\exp\left[-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right] \\
&= -\frac{1}{2}\Sigma^{-1}\frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right)}{2\pi|\boldsymbol{\Sigma}|^{1/2}} + \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right)}{2\pi|\boldsymbol{\Sigma}|^{1/2}} \\
&\cdot\frac{\partial\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right)}{2\pi|\boldsymbol{\Sigma}|^{1/2}} \\
&= -\frac{1}{2}\phi_{\mathbf{X}}\Sigma^{-1} + \frac{1}{2}\phi_{\mathbf{X}}\left[\Sigma^{-1}(\mathbf{x}-\mu)(\mathbf{x}-\mu)^T\Sigma^{-1}\right]
\end{aligned} \tag{B.3}$$

Thus, the integration can be expressed by

$$\int\frac{\partial\phi_{\mathbf{X}}(\mathbf{x})}{\partial\Sigma}Q(\mathbf{x})d\mathbf{x} = -\frac{1}{2}\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{x}}}[Q(\mathbf{X})]\Sigma^{-1} + \frac{1}{2}\Sigma^{-1}\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{x}}}[Q(\mathbf{X})(\mathbf{X}-\mu)(\mathbf{X}-\mu)^T]\Sigma^{-1} \tag{B.4}$$

Finally by substituting $(B.4)$ into $(B.2)$, one can have

$$\begin{aligned}
\frac{\partial}{\partial\Sigma}\mathscr{L}(\mathbf{x}_i,\mu,\Sigma) &= \frac{1}{2}\Sigma^{-1} - \frac{1}{2}\Sigma^{-1}(\mathbf{x}_i-\mu)(\mathbf{x}_i-\mu)^T\Sigma^{-1} - \frac{1}{2}\Sigma^{-1} + \frac{1}{2}\Sigma^{-1}\frac{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{X}}}\left[Q(\mathbf{X})(\mathbf{X}-\mu)(\mathbf{X}-\mu)^T\right]}{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{x}}}\left[Q(\mathbf{X})\right]}\Sigma^{-1} \\
&= \frac{1}{2}\Sigma^{-1}\left(\frac{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{x}}}\left[Q(\mathbf{X})(\mathbf{X}-\mu)(\mathbf{X}-\mu)^T\right]}{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{x}}}\left[Q(\mathbf{X})\right]} - (\mathbf{x}_i-\mu)(\mathbf{x}_i-\mu)^T\right)\Sigma^{-1}
\end{aligned} \tag{B.5}$$

# APPENDIX C

## DERIVATIVE OF PARTIAL DERIVATIVES IN (4.28)

The partial derivative of the cost function in (4.28) with respect to $\mathbf{L}$ can be expressed by

$$\frac{\partial}{\partial \mathbf{L}}\mathscr{L}(\mathbf{x}_i, \mu, \mathbf{L}) = -\frac{\partial \ln[\phi_{\mathbf{X}}(\mathbf{x}_i|\mu, \Sigma)]}{\partial \mathbf{L}} + \frac{\partial \ln \mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{X}}}[Q(\mathbf{X})]}{\partial \mathbf{L}} \tag{C.1}$$

The first section of $(C.1)$ is expressed as

$$\begin{aligned}
\frac{\partial \ln[\phi_{\mathbf{X}}(\mathbf{x}_i|\mu, \Sigma)]}{\partial \mathbf{L}} &= \frac{\partial}{\partial \mathbf{L}} \ln \left\{ \frac{1}{2\pi|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}_i - \mu)^T \Sigma^{-1}(\mathbf{x}_i - \mu)\right]\right\} \\
&= \frac{1}{2}\frac{\partial}{\partial \mathbf{L}} \ln|\Sigma^{-1}| - \frac{1}{2}\frac{\partial}{\partial \mathbf{L}}\left[(\mathbf{x}_i - \mu)^T \Sigma^{-1}(\mathbf{x}_i - \mu)\right] \\
&= \frac{1}{2}\frac{\partial}{\partial \mathbf{L}} \ln|(\mathbf{L}\mathbf{L}^T)^{-1}| - \frac{1}{2}\frac{\partial}{\partial \mathbf{L}}\left[(\mathbf{x}_i - \mu)^T (\mathbf{L}\mathbf{L}^T)^{-1}(\mathbf{x}_i - \mu)\right] \\
&= \frac{1}{2|\mathbf{L}\mathbf{L}^T|}\frac{\partial}{\partial \mathbf{L}}|\mathbf{L}\mathbf{L}^T - \frac{1}{2}\frac{\partial}{\partial \mathbf{L}} tr[\mathbf{L}^T(\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^T\mathbf{L}] \\
&= (\mathbf{L}\mathbf{L}^T)^{-1}\mathbf{L} - (\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^T\mathbf{L}
\end{aligned} \tag{C.2}$$

Now, the second section of $(C.1)$ is expressed as

$$\frac{\partial \ln \mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{X}}}[Q(\mathbf{X})]}{\partial \mathbf{L}} = \frac{1}{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{x}}}[Q(\mathbf{X})]} \int \frac{\partial \phi_{\mathbf{X}(\mathbf{x})}}{\partial \mathbf{L}} Q(\mathbf{x}) d\mathbf{x} \tag{C.3}$$

Where the partial derivative $\frac{\partial \phi_{\mathbf{X}(\mathbf{x})}}{\partial \mathbf{L}}$ can be expressed by

$$\begin{aligned}
\frac{\partial \phi_{\mathbf{X}(\mathbf{x})}}{\partial \mathbf{L}} &= \frac{\partial}{\partial \mathbf{L}} \left\{ \frac{1}{2\pi|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}_i - \mu)^T \Sigma^{-1}(\mathbf{x}_i - \mu)\right]\right\} \\
&= \frac{1}{2\pi}\frac{\partial|\mathbf{L}\mathbf{L}^T|^{1/2}}{\partial \mathbf{L}}. \exp\left[-\frac{1}{2}(\mathbf{x}_i - \mu)^T \Sigma^{-1}(\mathbf{x}_i - \mu)\right] \\
&+ \frac{1}{2\pi|\Sigma|^{1/2}}.\frac{\partial}{\partial \mathbf{L}}. \exp\left[-\frac{1}{2}(\mathbf{x}_i - \mu)^T \Sigma^{-1}(\mathbf{x}_i - \mu)\right] \\
&= \frac{1}{2\pi}|\mathbf{L}\mathbf{L}^T|^{1/2}(\mathbf{L}\mathbf{L}^T)^{-1}\mathbf{L}. \exp\left[-\frac{1}{2}(\mathbf{x}_i - \mu)^T \Sigma^{-1}(\mathbf{x}_i - \mu)\right] \\
&- \frac{1}{2}\phi_{\mathbf{X}}(\mathbf{x})\frac{\partial}{\partial \mathbf{L}} tr\left[\mathbf{L}^T(\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^T\mathbf{L}\right] \\
&= \phi_{\mathbf{x}}(\mathbf{x})\left[(\mathbf{L}\mathbf{L}^T)^{-1} - (\mathbf{x} - \mu)(\mathbf{x} - \mu)^T\right]\mathbf{L}
\end{aligned} \tag{C.4}$$

Thus, replacing $(C.4)$ in $(C.3)$, the expression is denoted as

$$\begin{aligned}
\frac{\partial \ln \mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{X}}}[Q(\mathbf{X})]}{\partial \mathbf{L}} &= \frac{1}{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{x}}}[Q(\mathbf{X})]} \int \phi_{\mathbf{x}}(\mathbf{x}) Q(\mathbf{X}) d\mathbf{x} (\mathbf{L}^{-1})^T \\
&- \frac{1}{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{x}}}[Q(\mathbf{X})]} \int \phi_{\mathbf{x}}(\mathbf{x})(\mathbf{x}-\mu)(\mathbf{x}-\mu)^T Q(\mathbf{X}) d\mathbf{x}(\mathbf{L}) \\
&= (\mathbf{L}^{-1})^T - \frac{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{x}}}\left[Q(\mathbf{X})(\mathbf{X}-\mu)(\mathbf{X}-\mu)^T Q(\mathbf{X})\right]}{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{x}}}[Q(\mathbf{X})]} \mathbf{L}
\end{aligned} \tag{C.5}$$

Finally, going back to $(C.1)$ the partial derivative of the cost function with respect to $\mathbf{L}$ can be expressed by

$$\begin{aligned}
\frac{\partial}{\partial \mathbf{L}} \mathscr{L}(\mathbf{x}_i, \mu, \mathbf{L}) &= -\frac{\partial \ln[\phi_{\mathbf{X}}(\mathbf{x}_i|\mu, \Sigma)]}{\partial \mathbf{L}} + \frac{\partial \ln \mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{X}}}[Q(\mathbf{X})]}{\partial \mathbf{L}} \\
&= -(\mathbf{L}\mathbf{L}^T)^{-1}\mathbf{L} - (\mathbf{x}_i-\mu)(\mathbf{x}_i-\mu)^T\mathbf{L} \\
&+ (\mathbf{L}\mathbf{L}^T)^{-1} - \frac{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{x}}}\left[Q(\mathbf{X})(\mathbf{X}-\mu)(\mathbf{X}-\mu)^T Q(\mathbf{X})\right]}{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{x}}}[Q(\mathbf{X})]} \mathbf{L} \\
&= \left\{ (\mathbf{x}_i-\mu)(\mathbf{x}_i-\mu)^T - \frac{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{x}}}\left[Q(\mathbf{X})(\mathbf{X}-\mu)(\mathbf{X}-\mu)^T Q(\mathbf{X})\right]}{\mathbb{E}_{\mathbf{X}\sim\phi_{\mathbf{x}}}[Q(\mathbf{X})]} \right\} \mathbf{L}
\end{aligned} \tag{C.6}$$