Sensor Planning for Bayesian Nonparametric Target Modeling

by

Hongchuan Wei

Department of Mechanical Engineering and Materials Science
Duke University

Date: _____

Approved:

_____
Silvia Ferrari, Co-Supervisor

_____
Michael Zavlanos, Co-Supervisor

_____
Earl H. Dowell

_____
Xiaobai Sun

Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Mechanical Engineering and Materials Science
in the Graduate School of Duke University
2016

Abstract

Sensor Planning for Bayesian Nonparametric Target Modeling

by

Hongchuan Wei

Department of Mechanical Engineering and Materials Science
Duke University

Date: _____

Approved:

_____
Silvia Ferrari, Co-Supervisor

_____
Michael Zavlanos, Co-Supervisor

_____
Earl H. Dowell

_____
Xiaobai Sun

An abstract of a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Mechanical Engineering and Materials Science
in the Graduate School of Duke University
2016

Abstract

Bayesian nonparametric (BNP) models, such as Gaussian processes (GPs) and Dirichlet processes (DPs), have been successively applied for modeling target behaviors or kinematics in various applications including environmental monitoring, traffic planning, endangered species tracking, dynamic scene analysis, autonomous robot navigation, and human motion modeling. The advantages of BNP models over the other approaches are that they are able to adjust the model complexity adaptively based on data, and increase their dimensionality as necessary, while avoiding both overfitting and underfitting. However, most existing works assume that the sensor measurements used to learn the BNP models are obtained *a priori* or that the target kinematics can be measured by the sensor at any given time throughout the task. Little work has been done for planning or controlling the sensors that obtain target measurements of mobile targets such that the most informative data may be obtained for reducing the uncertainty of the BNP models. This dissertation presents a systematic sensor planning approach for leaning BNP models from data, by deciding the sensor motion and control inputs that bring about the greatest model improvement over time. The approach is demonstrated by first considering the problem of learning a Gaussian process of a model from a time-invariant spatial phenomenon, such as an ocean current, a temperature distribution or a wind velocity field. Then, the approach is demonstrated by learning a Dirichlet process-Gaussian process (DPGP) model of multiple mobile targets, such as robots in a bounded workspaces, and

pedestrians motion patterns insider buildings.

The sensor planning and control approach involves the development of novel information theoretic functions applicable to BNP models and capable of representing the expected utility of future sensor measurements as a function of sensor control inputs and random environmental variables. A computationally efficient form of the expected Kullback Leibler divergence for Gaussian processes is derived by taking the expectation of the KL divergence between the current (prior) and posterior Gaussian process models at a set of collocation points, marginalizing out future measurements. Then, the approach is extended to develop a new information value function for Dirichlet process-Gaussian process mixture models of multiple dynamic targets. New theoretical results are presented to prove that the novel information theoretic functions are bounded, and to derive efficient estimators of expected information value that are proven to be unbiased, and characterized by an approximation error with a variance that decreases linearly with the number of samples. This dissertation analyzes the computational complexity of the proposed sensor planning and control problem, showing that the optimization of the DPGP information theoretic function subject to sensor dynamic constraints is NP-hard. A cumulative lower bound is then presented to reduce the computation required to polynomial time.

The information theoretic approach presented in this dissertation is demonstrated by developing three sensor planning methods for different target kinematics and sensor dynamics. When the control space of the sensor is discrete, a greedy algorithm that optimizes the information value of the next set of measurements can be used to effectively plan the sensor mode and motion at every time step. The efficiency of the greedy algorithm is demonstrated by a numerical experiment with data of ocean currents obtained by moored buoys. A sweep line algorithm is developed for applications where the sensor control space is continuous and unconstrained. Synthetic simulations as well as physical experiments with ground robots and a surveillance

camera are conducted to evaluate the performance of the sweep line algorithm. When the sensor is characterized by continuous control inputs and dynamic constraints, a lexicographic algorithm can be utilized to optimize an additive lower bound on the information value function, such that the sensor performance can be optimized over a finite time interval. The effectiveness of the lexicographic algorithm is demonstrated through numerical experiments involving measurements obtained from indoor pedestrians by a surveillance pan-tilt camera. Results from both numerical and physical experiments show that the information theoretic approach presented in this dissertation is highly effective at planning and controlling sensor measurements for learning BNP models of dynamic target or processes, with little or no prior knowledge.

To my parents and my wife.

Contents

List of Tables

List of Figures

List of Abbreviations and Symbols

Symbols

$\mathcal{A}$	Admissible set of sensor state.
d	Dimension of target state.
$D(\cdot\|\cdot)$	KullbackLeibler divergence.
$\mathbb{E}$	Expectation operator.
$\mathbf{f}_i(\cdot)$	ith target kinematics model.
$\mathcal{F}$	Set of target kinematics models.
G_j	Target-VF association index for the jth target.
$\mathcal{G}$	Set of target-VF association indices.
$H(\cdot)$	Differential entropy.
i	Index of kinematics model.
$I(\cdot;\cdot)$	Mutual information.
j	Index of target.
J	Cost/objective function.
k	Index of time step.
K	Finite control horizon.
l	Index of collocation points.
L	Number of collocation points.
M	Number of velocity fields.
$\mathbf{m}$	Sensor measurement.

$\mathcal{M}$	Set of sensor measurement.
N	Number of targets.
$\mathcal{N}$	Gaussian/Normal distribution.
$p(\cdot)$	Probability density/mass function.
$\phi(\cdot,\cdot)$, $\boldsymbol{\phi}(\cdot,\cdot)$	Gaussian process covariance function.
$\boldsymbol{\Phi}$	Gaussian process cross-covariance matrix.
$\pi_i, \boldsymbol{\pi}$	Mixture weights.
q	Dimension of sensor state.
r	Dimension of sensor control input.
$\mathbb{R}$	Set of real numbers.
$\mathbf{s}$	Sensor state.
s	Index of samples of target state.
S	Number of samples of target state.
$\mathcal{S}$	Sensor field of view.
σ_x	Standard deviation of position measurement noise.
σ_v	Standard deviation of velocity measurement noise.
σ_f	Standard deviation of output.
$\theta(\cdot)$, $\boldsymbol{\theta}(\cdot)$	Gaussian process mean function.
Θ	Gaussian process hyper-parameter.
υ, $\boldsymbol{\upsilon}$	Function evaluation.
$\mathbf{u}$	Sensor control input.
$\mathcal{U}$	Admissible control space.
ν, $\boldsymbol{\nu}$	Measurement noise.
w_{ij}	Probability that jth target follows ith kinematics model.
$\mathcal{W}$	Workspace.
$\mathbf{x}$	Target state.

$\boldsymbol{\chi}$	Sample of target state.
$\mathbf{X}$	Aggregated vector of target states.
$\boldsymbol{\xi}$	Collocation points.
$\mathbf{y}$	Measurement of target state.
$\mathbf{Y}$	Aggregated vector of target state measurements.
$\mathbf{z}$	Measurement of target velocity.
$\mathbf{Z}$	Aggregated vector of target velocity measurements.
$\mathbb{Z}$	Set of integers.

Abbreviations

BNP	Bayesian nonparametric.
Dir	Dirichlet distribution.
DP	Dirichlet process.
DPGP	Dirichlet process-Gaussian process.
EKLD	Expected KullbackLeibler divergence.
FOV	Field of view.
GP	Gaussian process.
MI	Mutual information.
MM	Mixture model.
MCMC	Markov chain Monte Carlo.
ODE	Ordinary differential equation.
PF	Particle filter.
RMSE	Root mean square error.
VF	Velocity field.

Acknowledgements

I would like to thank my advisor, Professor Silvia Ferrari, for her constant support, kindness, and instructions to help me with overcoming obstacles in my graduate study. Thanks to her, my graduate experience has been exciting and rewarding. I would also like to give special mention to my committee members who have generously sacrificed their time to help me along the way: Dr. Michael Zavlanos, Dr. Earl Dowell, and Dr. Xiaobai Sun. In addition, I would like to thank Dr. John Leonard and Dr. Guoquan Huang for collaborating with us and for proving me the opportunity to visit MIT. Moreover, I would like to thank our great collaborators: Dr. Jonathan P. How, Dr. Lawrence Carin, Dr. Miao Liu, Robert H. Klein, Shayegan Omidshafiei. A special thank you to my great labmates and fellow graduate students, who were always available and willing to help along the way: Wenjie Lu, Pingping Zhu, Xu Zhang, Guoxian Zhang, Wess Ross, Greg Foderaro, Ashleigh Swingler, Keith Rudd, Greyson Daugherty, Brian Bernard, Bernardo Fichera, Said Mansoor Wahab, Narit Tangsirivanich, Julian Morelli, Taylor Clawson.

I am indebted to those institutions which contributed financially to my graduate education: Duke University provided support through the department fellowship and teaching assistant; the office of naval research provided fanatical support through the ONR MURI Grant N000141110688; the WISeNet Integrative Graduate Education and Research Training (IGERT) program has provided me two years of trainee-ship with stipend and full tuition, normal fees and health insurance.

I wish to dedicate this dissertation to my family. To my parents and my parents in law. Your understanding is my resource of energy. To my wife, Yan. This will never happen without your support, understanding, and love.

1

Introduction

The problem of learning the kinematics of mobile targets by means of a mobile sensor is relevant to a wide range of applications, including security and surveillance [1], environmental monitoring [2, 3], and tracking of endangered species [4, 5]. There are a huge body of works related to this topic in the control literature that investigate various aspects of the sensor planning problem [6]. From a partial but extensive survey of existing works, it can be claimed that the sensor planning problem faces three key challenges. The first challenge is often if not always the problem of determining the model that describes the system under study, which is referred to as system modeling or system identification [7, 8, 9]. The answer to the first challenge can be seen as the interface between the real world of applications and the mathematical world of control theory and model abstractions [10]. Two lines of research can be pursued by adopting different philosophies. One philosophy is to use as simple models as possible, and the extreme case is the linear and time-invariant model, which has been successfully applied in a wide range of applications thanks to the rich set of theories and algorithms [11, 12]. The other philosophy is to use complex models that are adaptive for describing target kinematics from large amount of data. In this

line of research, Bayesian nonparametric models have been extensively applied due to their flexibility and resistance to overfitting or underfitting [13]. These properties are critical to problems where little knowledge of the targets is available. To this end, Bayesian nonparametric models are studied in this dissertation for developing efficient and manageable models of the target kinematics.

The second challenge that the sensor planning problem faces after the class of model is determined is how to design suitable reward functions to assess the sensor performance that would result from the sensor decisions before obtaining the future sensor measurements [14]. The reward function should truthfully represent the objectives of the research, which is learning the target kinematics. This topic is considered well studied for parametric models [15, 16]. One popular approach is to evaluate the utility of future measurements by their expected information value conditioned on the prior measurements and on the environmental variables [17]. Information theoretic functions are a natural choice for representing the information value because they measure the absolute or relative information content of probability mass functions or probability density functions associated with random variables in the stochastic model of the target kinematics [18]. A general approach is recently presented for estimating the *expected* information value of future sensor measurements in target classification problems [19]. However, not much work has been done on developing suitable objective functions for using Bayesian nonparametric models in control problems. To this end, this dissertation extends the approach in [19] to Bayesian nonparametric models by using the Kullback Leibler divergence to quantify the expected utility associated with future measurements in updating the Bayesian nonparametric models. The efficiency of the proposed information theoretic functions is demonstrated through a variety of examples where the target kinematics are modeled by various Bayesian nonparametric models.

Finally, optimization strategies need to be examined for the purpose of maximiz-

ing or minimizing the objective functions under various constraints on the sensor dynamics. The most suitable optimization approach often depends on the specific assumptions and formulations of the sensor planning problem. To this end, several scenarios where the proposed Bayesian nonparametric target kinematics models and the novel information theoretic functions can be applied are discussed. Corresponding control strategies are developed according to the assumptions in different scenarios.

This dissertation is organized as follows. Chapter 2 presents a comprehensive literature review on the Bayesian nonparametric models and primitive background knowledge of the Gaussian process and the Dirichlet process. Detailed discussions of the advantages and disadvantages of the Bayesian nonparametric models are also provided in Chapter 2 as the motivation of adopting them for describing target kinematics adaptive from data. Chapter 3 provides the assumptions and the mathematical formulations of the sensor planning problem and the research goals to be achieved. Chapter 4 presents the Bayesian nonparametric target kinematics models, and the corresponding inference, prediction and filtering algorithms. Chapter 5 proposes novel information theoretic functions for the Bayesian nonparametric target kinematics models and studies the properties of the proposed information theoretic functions. Chapter 6 presents a variety of sensor planning algorithms to several scenarios where the Bayesian nonparametric target kinematics models and the novel information theoretic functions can be applied. Chapter 7 provides examples of real-world applications to demonstrate the efficiency of the proposed sensor planning algorithms. Finally, conclusions are drawn in Chapter 8.

2

Background Knowledge

Bayesian models have been applied for describing processes associated with uncertainties arguably ever since the Bayes' theorem is derived in 1763 [20]. However, they are not widely used until the development of modern computers from about 1975, due to the high computational complexity for obtaining the posterior distributions [21]. The applicability of Bayesian models is widened significantly again in the 1990s with the development of numerical methods and software packages for numerical integration and Markov chain Monte Carlo (MCMC) sampling algorithms [21, 22]. In Bayesian statistics, observed data are assumed to be described by probability distributions with unknown parameters, where the observed data are treated as fixed quantities and the parameters of the probability distributions are treated as random variables [23]. In the Bayesian paradigm, current knowledge about the model parameters is expressed by a probability distribution on the parameters, referred to as the *prior distribution* [24]. The conditional probability distribution of the observed data given the model parameters is called the *likelihood*. The Bayes' theorem shows that the *posterior* probability density function of the model parameters is proportional to the product of the prior density with the likelihood function, and can be obtained

by proper normalization of the product. The whole process is referred to as the *Bayesian inference* and the adopted probabilistic models are the *Bayesian models*. The use of the prior distribution distinguishes the Bayesian inference from the *frequentist inference*, which treats the model parameters as unknown fixed quantities and learns the optimal parameter values by maximizing the likelihood functions [25]. Advantages of inferences with Bayesian models over frequentist inferences include the existence of posterior distributions so that probabilities of the parameters belonging to any set can be calculated. However, the biggest criticism about the Bayesian is also the use of the prior distribution, since it brings subjective information to the inference [26]. The argument between Bayesian and frequentist statistics has been discussed in many works, and no unifying view has been accepted by all the researchers [27]. Since Bayesian models are more straightforward and mathematically rigourous to cope with infinite number of parameters, they are used in this dissertation for the *nonparametric* target kinematics modeling when the number of classes of target kinematics can not be determined *a priori*, which is discussed as follows.

In both the Bayesian inference and the frequentist inference discussed above, a key question to be answered is *model selection*, that is how to choose a model at an appropriate level of complexity [28]. For Bayesian (parametric) models, the model complexity is determined by the number of model parameters specified by the prior distributions. For example, the most prominently asked model selection question is how to determine the proper number of components in a mixture model. If the Dirichlet distribution is used as the prior of mixture weights, the model selection question is simply to decide the dimension of the Dirichlet distribution. Working with Bayesian parametric models, the model selection problems are often addressed by first fitting several models, with different numbers of parameters, and then by selecting the best one using model comparison metrics [29]. These metrics are usually weighted summations of two parts. The first part evaluates the training error that

measures how well the model fits the data [30]. The second part is a complexity penalty that favors simpler models with less parameters according to the Occam's razor that can be interpreted as stating "Among competing hypotheses, the one with the fewest assumptions should be selected" [31]. A few problems exist with the parametric approach to model selection. The most dominating one is the difficulty in choosing the "correct" number of parameters. If the number of parameters is too large, Bayesian models are overfitted and tend to learn the noise of the training data instead of the underlying relationship. Therefore, overfitted Bayesian models have large predictive errors when given new test data. On the other hand, if the number of parameters is too small, underfitting occurs as a result of the excessively simple model. In this scenario, Bayesian models are not able to describe the training data well enough and the predictive performance is also poor. Validation or cross-validation using test data may alleviate the problems of overfitting and underfitting, however, it is also argued that when training a model, no test data should be used. In addition, high computational cost is also a disadvantage of such approaches, since only one of all the trained models is selected and the computational resource spent for the remaining models is wasted.

In contrast, Bayesian nonparametric (BNP) models provide a different approach to the model selection problem by assuming that the number of parameters is infinite. In fact, a commonly used definition of Bayesian nonparametric models is probability models with infinitely many parameters [32]. Equivalently, Bayesian nonparametric models are probability models on function spaces [33]. Rather than comparing models that vary in complexity, the BNP approach is to fit a single model that can adapt its complexity to the data [34]. The adaptation of model complexities in Bayesian nonparametric models is achieved by treating the number of parameters as part of the posterior distribution [21]. The number of parameters is then determined by analyzing the data and therefore only a finite subset of the available infinite parameters

are invoked for any given finite data set [13]. The development of Bayesian nonparametric models is made possible after the required mathematical tools for specifying distributions on function spaces are discovered, such as the Kolmogorov consistency theorem [35] and the de Finetti's theorem [36, 37]. The first Bayesian nonparametric model is commonly considered as the Dirichlet process developed in the 1970s from the study of distribution of random measures mainly for mathematical interest [38, 36, 39]. Since MCMC sampling algorithms for Dirichlet process mixtures become available in the 1990s and make latent variable models with Bayesian nonparametric components applicable to practical problems, the development of BNPs has experienced explosive growth [40, 41].

Comparing to parametric models that fixes the number of parameters in the prior distribution, Bayesian nonparametric models have several advantages. First of all, no assumptions on the number of parameters means that the burden of model selection is not placed on the users [34]. In addition, in many contexts of statistical modeling it is desirable to make fewer assumptions about the underlying populations from which the data are obtained [42]. Furthermore, Bayesian nonparametric models are considered to be more robust and easier for model diagnostics and sensitivity analysis [33]. The disadvantages of the Bayesian nonparametric models are also apparent. The mathematical complexities are more demanding, since placing well-defined probability distributions on potentially infinite-dimensional spaces is inherently harder than for Euclidean spaces [21]. In addition, posterior distributions in Bayesian nonparametric methods are more complicated and are more challenging for designing inference algorithms. However, the availability of powerful software packages have lessened the difficulty for using Bayesian nonparametric models [43]. Moreover, the increasing demand of flexible models in scientific works has been constantly boosting the popularity of Bayesian models. Successful applications of Bayesian nonparametric models include regression [44, 45, 46], classification [47, 48, 49, 50, 51],

clustering [37, 52, 53, 39, 40, 54], latent factor modeling [55, 56, 57, 58, 59], sequential modeling [60, 61, 62, 63], image processing [64, 65, 66, 67], and topic modeling [68, 69, 70, 71, 72], just to name a few.

Among all the successful applications of the Bayesian nonparametric models, target kinematics modeling is the most pertinent to the sensor planning. The most commonly utilized Bayesian nonparametric models for target kinematics modeling in the literature are Gaussian processes (GPs) and Dirichlet processes (DPs), since GPs provide a powerful technique for function regressions and DPs are useful for clustering similar kinematic equations. Based on the models applied and the modeling purpose, the Bayesian nonparametric models to target kinematics modeling can be categorized as follows. A group of studies utilize GPs to represent state transitions of dynamic systems and propose various filtering techniques based on the measurement models and the different assumptions on the dynamic systems [73, 74, 75, 76, 77, 78, 79, 80]. Applications of these filters include but not limited to multiple targets tracking and human pose learning. Another group of applications of GPs in target kinematics modeling assume that the target movements can be described as trajectories consisting of histories of target positions and/or velocities. For example, Ellis et al. proposed a non-parametric model for pedestrian motion based on Gaussian process regression, in which trajectory data are modelled by regressing relative motion against current position [81]. Mann et al. constructed and applied the GP for flight trajectory generation of pigeons trained to return home from specific release sites [82]. In [44], recognitions of motions and activities of targets in videos are performed by modeling target kinematics as a continuous dense flow field from a sparse set of vector sequences using Gaussian process regression, allowing for incrementally predicting possible paths and detecting anomalous events from online trajectories. In [83], a navigation algorithm for a car-like robot moving in a dynamic, uncertain environment populated with mobile obstacles is designed by representing

typical motion patterns of the obstacles using pre-learned Gaussian processes. In [84], the problem of safe navigation of a mobile robot through crowds of dynamic agents with uncertain trajectories is solved by estimating crowd interaction from data using a nonparametric statistical model based on dependent output Gaussian processes. Later, the approach is extended in [85] and a probabilistic predictive model of cooperative collision avoidance and goal-oriented behavior is developed by considering multiple goals and stochastic movement duration in the interacting Gaussian processes. In [86], an autoregressive Gaussian process motion model is applied in the problem of navigation through a partially observable dynamic environment. Besides a single Gaussian process, finite mixtures of GPs are also applied for target kinematics modeling. For example, Aoude et al. presented an efficient trajectory prediction algorithm for future collision avoidance by combining rapidly exploring random tree-reach algorithm with a finite mixtures of Gaussian processes [87, 88]. Reece et al. studied how groups of animals move collectively and how they effectively align their movements by using a mixture of GPs, where one GP is applied to model the distribution of an individual movement path [89]. In [90], multiple goal trajectories are modeled by a mixture of GPs over waypoints and the transition time between these waypoints. Furthermore, DP mixtures are also adopted for modeling target kinematics. Fox et al. considered the problem of data association for multi-target tracking in the presence of an unknown number of targets by using the Dirichlet process to provide a prior on partitions of the observations among targets whose dynamics are individually described by state space models [91]. In [92], a framework called Dual Hierarchical Dirichlet Processes is proposed for unsupervised trajectory analysis and semantic region modeling in surveillance settings. In [93], the Dirichlet process active region framework is developed that learns motion patterns from data and is able to group motion patterns with small planar shift in the same cluster. Finally, Joseph et al. proposed the Dirichlet process-Gaussian process mixture model for modeling

target kinematics, where the GPs provide a flexible representations for each individual motion pattern, while the DP assigns observed trajectories to particular motion patterns. Both automatically adjust the complexity of the motion model based on the available data [94, 95].

In summary, Bayesian nonparametric models have been extensively applied in sensor planning problems for modeling target kinematics or dynamics, due to their flexibility and resistance to over-fitting/under-fitting and ability to solve problems where domain knowledge is unavailable. However, existing works often treat the Bayesian nonparametric models simply as black boxes of target kinematics [73, 74, 75, 76, 77, 78, 79, 80, 83, 86, 94, 95]. Rather than being targeted at improving BNP models, to date the measurements used for updating the Bayesian nonparametric models are typically obtained while in pursuit of other objectives, such as tracking or estimation of target states. Previous methods that consider the improvement of Bayesian nonparametric models assumed targets are static or the sensor field-of-view is unbounded, such that target measurements are always available [81, 82, 44, 84, 85, 89, 90, 91, 92, 93]. This dissertation relaxes these assumptions and considers mobile targets observed by a sensor with a bounded field-of-view, such that when measurements are available only when the sensor planning problem takes into consideration both the field-of-view geometry and the target trajectory. In addition, new information-based sensor planning algorithms are proposed such that Bayesian nonparametric models can be utilized in applications where learning target kinematics with little or no priori knowledge efficiently is of great importance.

2.1 Gaussian Process

Target kinematics modeling often involves determining the equations that govern the motion of the target [96]. Traditionally, these kinematics equations can be derived from physical laws or geometry properties of the target system, and can be expressed

in the form of partial differential equations. This class of methods is accurate when the underlying physical law is easy to understand. However, in complex systems, simplifications are always required and, therefore, the model does not represent the system kinematics or dynamics exactly. In many applications, the values of target kinodynamic parameters are difficult to obtain *a priori*. In some cases, lookup tables from experimental data or empirical formulas are used, but this approach may lessen the accuracy of the model, when the conditions of the application are different from the ones considered by the lookup tables or the empirical formulas. Another approach to target kinematics modeling is using parametric models, such as hidden Markov models, to learn the parameters from data [93]. Markov models suffer from several disadvantages, including the Markov separation property [97]. This may be resolved by augmenting the system state, however, the state-space of the Markov model increases exponentially with the dimension of the system state. Markov models usually represent the system state by discrete distributions, and thus cannot cope well with continuous systems. Moreover, in some areas, the data available will be insufficient to estimate reliable probability or transfer rates, especially for rare transitions [98]. In contrast to the above methods, Bayesian nonparametric models provide an approach to adaptively learning the model parameters and dimensionality from data.

Among all Bayesian nonparametric models, the Gaussian process is arguably the most popular for describing target kinematics. The Gaussian process can be seen as an infinite-dimensional generalization of multivariate normal distributions. It can also been seen as a distribution over the function space, therefore, it can be used as the prior on the functions to be learned. The posteriors of the functions can be easily obtained in simple analytical form by the Gaussian process regression technique, provided the measurement noise is Gaussian distributed. Furthermore, properties of the underlying function, such as stationary and smoothness, can be easily specified

by the choice of the parameters of the GP. For the above three reasons, Gaussian processes have been successfully applied in learning spatial or temporal phenomena from noisy measurements, without *a-priori* specifications of model complexities [99, 100, 94, 101, 102]. One of the main drawbacks of Gaussian processes is the cubic complexity dependence on the number of training data [78]. Sparse GP regression approaches can be used to reduce the complexity by intelligently selecting a subset of the training data [103, 104, 105, 106, 107]. A formal definition of the Gaussian process is presented as follows,

Definition 1 (Gaussian process [43])**.** *A Gaussian process defines a multivariate Gaussian distribution over functions, $P(f)$, where $f : \mathcal{W} \to \mathbb{R}$. Let $F = \{f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n) \mid \mathbf{x}_i \in \mathcal{W}\}$ be a set of function values evaluated at n points in $\mathcal{W}$. Then, $P(f)$ is a Gaussian process if for any finite set $\{\mathbf{x}_1, \ldots, \mathbf{x}_n \mid \mathbf{x}_i \in \mathcal{W}\}$ the marginal distribution $P(F)$ is a joint multivariate Gaussian distribution.*

A Gaussian process is completely specified by its mean function and covariance function [108]. Let υ denote the latent random variable that represents the function evaluation at $\mathbf{x}$, that is, $\upsilon \triangleq f(\mathbf{x})$. Then, the mean function, $\theta : \mathcal{W} \to \mathbb{R}$, and the the covariance function, $\phi : \mathcal{W} \times \mathcal{W} \to \mathbb{R}$, of a GP are defined as follows:

$$\theta(\mathbf{x}) \triangleq \mathbb{E}_{\upsilon}[f(\mathbf{x})], \quad \forall \mathbf{x} \in \mathcal{W} \tag{2.1}$$

$$\phi(\mathbf{x}, \mathbf{x}') \triangleq \mathbb{E}_{\upsilon,\upsilon'}\left\{[f(\mathbf{x}) - \theta(\mathbf{x})][f(\mathbf{x}') - \theta(\mathbf{x}')]\right\}, \quad \forall \mathbf{x}, \mathbf{x}' \in \mathcal{W} \tag{2.2}$$

where $\mathbb{E}_{\upsilon}[\cdot]$ denotes the expectation operator with respect to the random variable υ [1]. In summary, the notation,

$$f(\mathbf{x}) \sim \mathrm{GP}\left(\theta(\mathbf{x}), \phi(\mathbf{x}, \mathbf{x}')\right) \tag{2.3}$$

can be used to indicate that f is "distributed as" the Gaussian process with mean function $\theta(\cdot)$ and covariance function $\phi(\cdot, \cdot)$. The existence of the Gaussian process is proved by the Kolmogorov's consistency theorem [35, 109].

2.1.1 Gaussian Process Regression

A major advantage of working with GPs is the existence of simple analytic formulas for mean and covariance of the posterior distribution, which allows easy implementation of algorithms [110]. Assume that $k \in \mathbb{Z}^+$ training input and output pairs, $\{\mathbf{x}_i, z_i \mid i = 1, \ldots, k\}$, are available, where z_i is a noisy measurement of $f(\cdot)$ taken at $\mathbf{x}_i \in \mathcal{W}$, such that,

$$z_i = f(\mathbf{x}_i) + \nu, \quad i = 1, \ldots, k \tag{2.4}$$

where ν is the measurement noise. If the measurement noise, ν, is zero mean Gaussian distributed with standard deviation σ_n, the posterior distribution of $f(\cdot)$ conditioned on the training data is also a GP, with modified mean and covariance functions, calculated as follows.

Without loss of generality, assume that the domain of $f(\cdot)$ is a subspace of the d-dimensional Euclidean space, such that $\mathcal{W} \subset \mathbb{R}^d$. For brevity, the d-dimensional column vector inputs for all k cases can be organized in the $dk \times 1$ vector,

$$\mathbf{X} \triangleq \begin{bmatrix} \mathbf{x}_1^T & \cdots & \mathbf{x}_k^T \end{bmatrix}^T \tag{2.5}$$

In addition, the measurements are collected in a $k \times 1$ vector,

$$\mathbf{z} \triangleq \begin{bmatrix} z_1 & \cdots & z_k \end{bmatrix}^T \tag{2.6}$$

Assume that $\upsilon_i \triangleq f(\mathbf{x}_i)$ is adopted to denote the function evaluation at $\mathbf{x}_i$, for $i = 1, \ldots, k$. Then, the function evaluations at all the k inputs can be organized in a $k \times 1$ vector,

$$\boldsymbol{\upsilon} \triangleq \begin{bmatrix} \upsilon_1 & \cdots & \upsilon_k \end{bmatrix}^T \tag{2.7}$$

Consider any other set of m input vectors also in the domain of $f(\cdot)$, denoted by $\{\mathbf{x}'_1, \ldots, \mathbf{x}'_m \mid \mathbf{x}'_i \in \mathcal{W}\}$. It follows that $\mathbf{X}'$ and $\boldsymbol{\upsilon}'$ can be defined similarly as (2.5) and (2.6), respectively. Then, the $k \times m$ cross-covariance matrix of random vectors

$\boldsymbol{\upsilon}$ and $\boldsymbol{\upsilon}'$ can be defined as,

$$\begin{aligned}\boldsymbol{\Phi}(\mathbf{X}, \mathbf{X}') &\triangleq \mathbb{E}_{\boldsymbol{\upsilon},\boldsymbol{\upsilon}'}\left[\left(\boldsymbol{\upsilon} - \mathbb{E}[\boldsymbol{\upsilon}]\right)\left(\boldsymbol{\upsilon}' - \mathbb{E}[\boldsymbol{\upsilon}']\right)^T\right] \\ &= \begin{bmatrix} \phi(\mathbf{x}_1, \mathbf{x}'_1) & \cdots & \phi(\mathbf{x}_1, \mathbf{x}'_m) \\ \vdots & \ddots & \vdots \\ \phi(\mathbf{x}_k, \mathbf{x}'_1) & \cdots & \phi(\mathbf{x}_k, \mathbf{x}'_m) \end{bmatrix}\end{aligned} \tag{2.8}$$

where $\phi(\cdot, \cdot)$ is defined in (2.2). Then, from GP regression, the joint distribution of $\mathbf{y}$ and $\boldsymbol{\upsilon}'$ is,

$$\begin{bmatrix} \mathbf{z} \\ \boldsymbol{\upsilon}' \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\theta}(\mathbf{X}) \\ \boldsymbol{\theta}(\mathbf{X}') \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma} & \boldsymbol{\Phi}(\mathbf{X}, \mathbf{X}') \\ \boldsymbol{\Phi}(\mathbf{X}', \mathbf{X}) & \boldsymbol{\Phi}(\mathbf{X}', \mathbf{X}') \end{bmatrix}\right) \tag{2.9}$$

where

$$\boldsymbol{\theta}(\mathbf{X}) \triangleq \begin{bmatrix} \theta(\mathbf{x}_1) & \cdots & \theta(\mathbf{x}_k) \end{bmatrix}^T \tag{2.10}$$

$$\boldsymbol{\Sigma} \triangleq \boldsymbol{\Phi}(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}_k \tag{2.11}$$

$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{K})$ denotes the multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{K}$, and $\mathbf{I}_k$ is the k-dimensional identity matrix. The marginalization over $\mathbf{y}$ in (2.9) shows that the posterior distribution of $\boldsymbol{\upsilon}'$ given the training data is still a multivariate Gaussian distribution with mean $\boldsymbol{\mu}'$ and covariance $\boldsymbol{\Sigma}'$, such that,

$$\boldsymbol{\mu}' = \boldsymbol{\theta}(\mathbf{X}') + \boldsymbol{\Phi}\left(\mathbf{X}', \mathbf{X}\right) \boldsymbol{\Sigma}^{-1}[\mathbf{z} - \boldsymbol{\theta}(\mathbf{X})] \tag{2.12}$$

$$\boldsymbol{\Sigma}' = \boldsymbol{\Phi}(\mathbf{X}', \mathbf{X}') - \boldsymbol{\Phi}(\mathbf{X}', \mathbf{X})\boldsymbol{\Sigma}^{-1}\boldsymbol{\Phi}(\mathbf{X}, \mathbf{X}') \tag{2.13}$$

Figure 2.1 shows two examples of Gaussian process regression with noisy training data, where the posterior mean function is treated as the function prediction. The ground-truth function in Fig. 2.1(b) is $f(\mathbf{x}) = \cos(x)\sin(y)$, where $\mathbf{x} = \begin{bmatrix} x & y \end{bmatrix}^T$.

2.1.2 Gaussian Process Hyper-Parameter Optimization

For Gaussian process regressions, the properties of $f(\cdot)$, such as smoothness and periodicity, can be enforced by the choice of the covariance function [111]. For

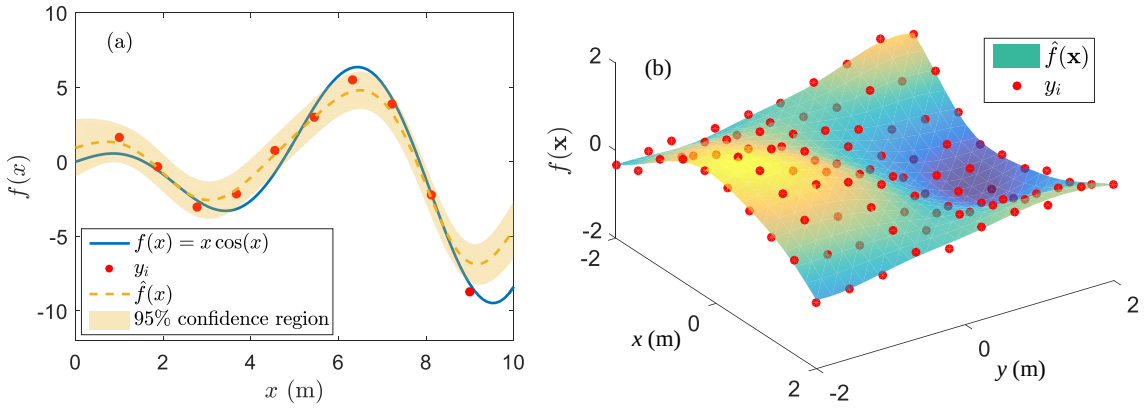

FIGURE 2.1: Gaussian process regression examples for (a) 1D domain and (b) 2D domain.

example, if the dependence between $f(\mathbf{x})$ and $f(\mathbf{x}')$ are invariant when $\mathbf{x}$ and $\mathbf{x}'$ are translated simultaneously, *stationary* covariance functions that depend on the relative position of its two inputs, $\mathbf{x} - \mathbf{x}'$, should be used. The covariance function, $\phi(\cdot, \cdot)$, is then specified by a set of *hyper-parameters*, denoted by Θ [108]. A common choice for a stationary covariance functions is the *squared-exponential* function,

$$\phi(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}')^T \mathbf{\Lambda}^{-1}(\mathbf{x} - \mathbf{x}')\right) \tag{2.14}$$

where σ_f^2 is the output variance that determines the average distance between $f(\cdot)$ and $\theta(\cdot)$, and $\mathbf{\Lambda} = \text{diag}([\lambda_1 \quad \cdots \quad \lambda_d])$ is a diagonal matrix of length-scales in all dimensions of $\mathbf{x} \in \mathbb{R}^d$, which control the smoothness of the covariance function [112]. $\text{diag}(\cdot)$ applied to a row vector denotes an operation that places elements of the row vector on the diagonal of a zero matrix. In addition, the measurement noise standard deviation, σ_n, can also be treated as a hyper-parameter. Therefore, the set of hyper-parameters for the squared-exponential covariance function in (2.14) is $\Theta = \{\sigma_f, \mathbf{\Lambda}, \sigma_n\}$.

In the Bayesian framework, the optimal hyper-parameters can be learned by

maximizing the *marginal likelihood* function [113],

$$\log p(\mathbf{z}|\mathbf{X},\Theta) = -\frac{1}{2}\mathbf{z}^T\boldsymbol{\Sigma}^{-1}\mathbf{z} - \frac{1}{2}\log\left[(2\pi)^k|\boldsymbol{\Sigma}|\right] - \frac{k}{2}\log 2\pi \tag{2.15}$$

where $\boldsymbol{\Sigma}$ is the covariance matrix of the noisy measurements $\mathbf{z}$ defined in (2.11), and $|\cdot|$ denotes the determinant of a matrix. The maximization of the marginal likelihood function can be solved by gradient-based algorithms. The search directions are defined by the partial derivatives of (2.15) with respect to the hyper-parameters Θ. In case of the squared-exponential covariance function (2.14), the partial derivatives can be found by matrix calculus, as follows,

$$\begin{aligned}
\frac{\partial \log p(\mathbf{z}|\mathbf{X},\Theta)}{\partial \sigma_f} &= \frac{1}{\sigma_f}\mathrm{tr}\left[(\boldsymbol{\alpha}\boldsymbol{\alpha}^T - \boldsymbol{\Sigma}^{-1})\boldsymbol{\Phi}(\mathbf{X},\mathbf{X})\right] \\
\frac{\partial \log p(\mathbf{z}|\mathbf{X},\Theta)}{\partial \sigma_n} &= \frac{1}{\sigma_n}\mathrm{tr}\left(\boldsymbol{\alpha}\boldsymbol{\alpha}^T - \boldsymbol{\Sigma}^{-1}\right) \\
\frac{\partial \log p(\mathbf{z}|\mathbf{X},\Theta)}{\partial \lambda_\ell} &= \frac{1}{4\lambda_\ell^2}\mathrm{tr}\left[(\boldsymbol{\alpha}\boldsymbol{\alpha}^T - \boldsymbol{\Sigma}^{-1})(\boldsymbol{\Phi}(\mathbf{X},\mathbf{X})\circ\mathbf{D})\right]
\end{aligned} \tag{2.16}$$

for $\ell = 1,\ldots,d$, where,

$$\boldsymbol{\alpha} \triangleq \boldsymbol{\Sigma}^{-1}\mathbf{z} \tag{2.17}$$

$$\mathbf{D}_{(i,j)} \triangleq \left(\mathbf{e}_\ell^T\mathbf{x}_i - \mathbf{e}_\ell^T\mathbf{x}_j\right)^2 \tag{2.18}$$

$$\mathbf{e}_\ell \triangleq \big[\underbrace{0 \quad \cdots \quad 0}_{\ell-1} \quad 1 \quad \underbrace{0 \quad \cdots \quad 0}_{d-\ell}\big]^T \tag{2.19}$$

$\mathrm{tr}(\cdot)$ calculates the trace of a matrix, and $\circ$ denotes the Hadamard product (element-wise product) of two matrices. Notice that the computational complexities for calculating the derivatives of σ_f, σ_n and λ_1 are $O(k^2)$ after $\boldsymbol{\Sigma}^{-1}$ is obtained for the k training data pairs. In addition, the computational complexities for $\lambda_2,\ldots,\lambda_d$ are only $O(1)$. Therefore, all the partial derivatives of the marginal likelihood function in (2.15) with respect to the hyper-parameters, Θ, can be calculated efficiently, which

justifies the usage of gradient-based methods in the hyper-parameter optimization. An example of the GP hyper-parameter optimization is shown in Fig. 2.2, demonstrating that optimizing the hyper-parameters can greatly reduce the regression error.

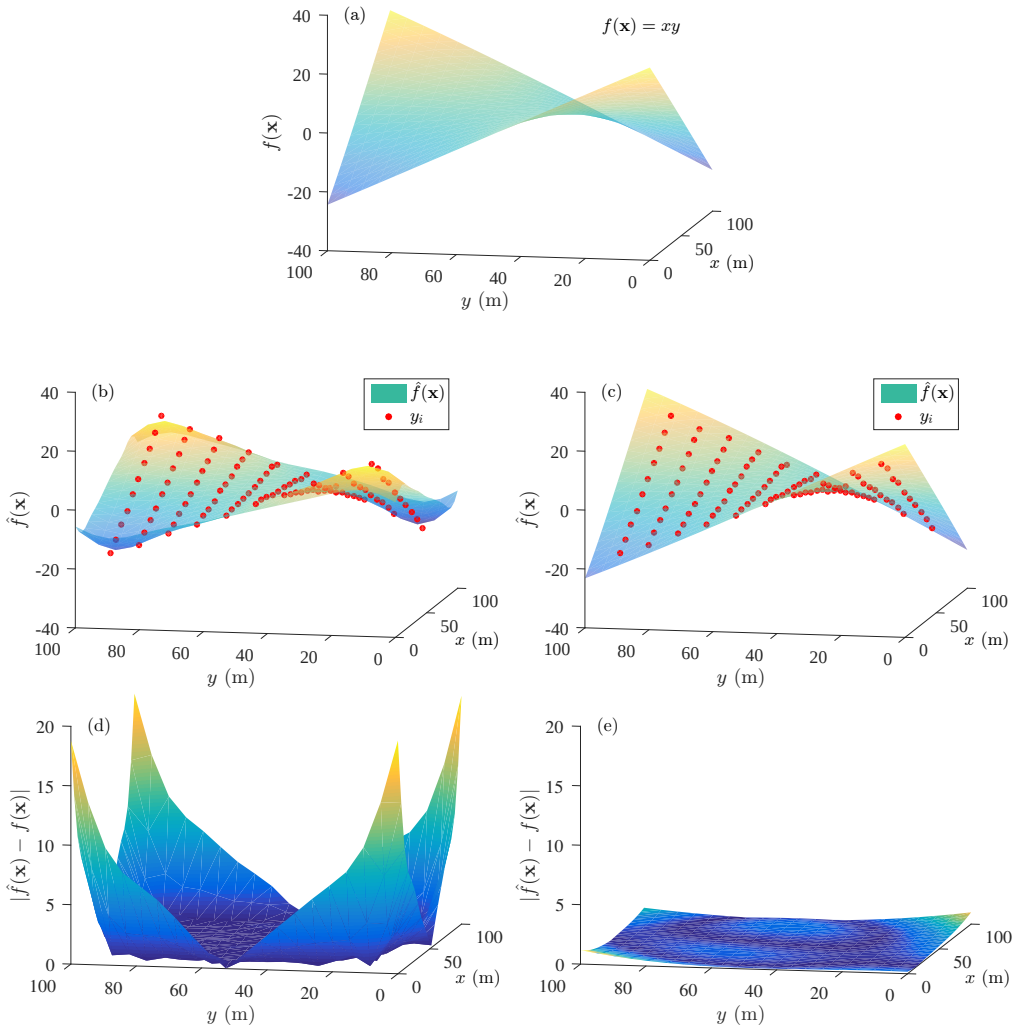

FIGURE 2.2: Example of Gaussian process hyper-parameter optimization with squared-exponential covariance function (2.14). (a) Ground-truth of the function $f(\mathbf{x})$; (b) Posterior mean function without hyper-parameter optimization; (c) Posterior mean function with hyper-parameter optimization; (d) Error of GP regression without hyper-parameter optimization; (e) Error of GP regression with hyper-parameter optimization.

2.2 Dirichlet Process

For problems involved with mixture models, such as data clustering, Dirichlet processes (DPs) have been successfully applied as priors of the mixture weights [94, 95]. DPs do not require the users to specify the number of clusters *a priori*, because they allow the creation of new clusters when necessary as data are obtained over time [114, 41, 115]. When the targets follow multiple classes of kinematics, a single Gaussian process is not sufficient for describing all observed trajectories, and a mixture of Gaussian process must be utilized. In this case, the Dirichlet process can be used to learn the total number of target kinematics from noisy sensor measurements, when this information is not available to the sensor *a priori*. Also, the target-kinematics associations can also be learned by the employment of the Dirichlet process.

The Dirichlet process was first developed in the 1970s [38, 39] and remains one of the most popular nonparametric model in the literature [33]. Its popularity derives from the development of posterior inference algorithms for the Dirichlet process, most of which are MCMC-based algorithms [114, 116, 41, 117, 118, 119], and variational inference methods [115, 120, 121, 122]. The Dirichlet process can be seen as a infinite-dimensional generalization of the Dirichlet distribution. It is called a Dirichlet process because it has Dirichlet distributed finite dimensional marginal distributions [123]. In the same way that the Dirichlet distribution is the conjugate prior for the categorical distribution (discrete probability distribution), the Dirichlet process is the conjugate prior for infinite discrete distributions. The Dirichlet process is a distribution over distributions, that is, samples from a Dirichlet process are distributions. In addition, the sampled distributions are discrete, but cannot be described using a finite number of parameters, thus the Dirichlet process is categorized as a nonparametric model. To this end, the most important application of the Dirichlet process is the prior probability distribution in infinite mixture models [123].

Since the Dirichlet process is the generalization of the Dirichlet distribution, it is helpful to study the properties of the simpler Dirichlet distribution. The Dirichlet distribution of order $d \geqslant 2$ with parameters, $\alpha_1, \ldots, \alpha_d > 0$, has a probability density function with respect to Lebesgue measure on the Euclidean space $\mathbb{R}^{d-1}$ given by,

$$p(\boldsymbol{\pi}) = \frac{\Gamma(\Sigma_{i=1}^{d}\alpha_i)}{\Pi_{i=1}^{d}\Gamma(\alpha_i)}\Pi_{i=1}^{d}\pi_i^{\alpha_i-1} \tag{2.20}$$

on the open $(d-1)$-dimensional simplex: $\left\{\boldsymbol{\pi} \in \mathbb{R}^d \mid \pi_1, \ldots, \pi_d > 0, \quad \sum_{i=1}^{d}\pi_i = 1\right\}$, and zero otherwise, where $\boldsymbol{\pi} \triangleq [\pi_1 \quad \cdots \quad \pi_d]^T$, and $\Gamma(\cdot)$ is the gamma function [109]. In order to demonstrate the meaning of the parameters, $\{\alpha_i\}_{i=1}^{d}$ can be normalized, such that,

$$\hat{\boldsymbol{\alpha}} \triangleq \frac{1}{\sigma}[\alpha_1 \quad \cdots \quad \alpha_d]^T, \quad \text{and} \quad \sigma \triangleq \sum_{i=1}^{d}\alpha_i \tag{2.21}$$

The normalized parameter, $\hat{\boldsymbol{\alpha}}$, also referred to as the base distribution, determines the mean of the Dirichlet distribution, as illustrated in Fig. 2.3. The normalization constant, σ, controls the variance of the Dirichlet distribution, therefore, it is also known as the concentration parameter. Figure 2.4 shows that by increasing σ, the mean of the Dirichlet distribution is not changed, however, distribution is more concentrated around the mean.

The Dirichlet process generalizes the Dirichlet distribution to an infinite dimension. In order to present the formal definition of the Dirichlet process, the σ-algebra and the measurable space need to be introduced first:

Definition 2 (σ-algebra [124])**.** *The σ-algebra on a nonempty set A is a collection B of subsets of A, which obeys the following properties:*

1. $\varnothing \in B$.

2. *If $E \in B$, then the complement of E, denoted by $E^c \triangleq A \backslash B$, is also in B.*

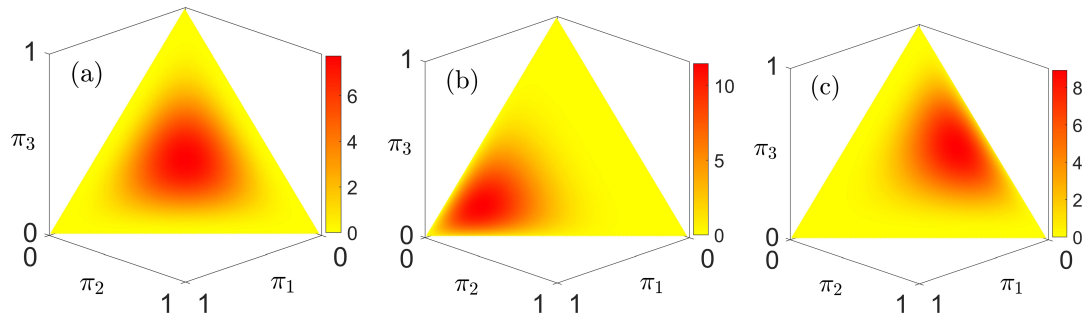

FIGURE 2.3: Example of Dirichlet distributions with concentration parameter $\sigma = 10$, and (a) $\hat{\boldsymbol{\alpha}} = [1/3 \quad 1/3 \quad 1/3]^T$; (b) $\hat{\boldsymbol{\alpha}} = [0.6 \quad 0.2 \quad 0.2]^T$; (c) $\hat{\boldsymbol{\alpha}} = [0.2 \quad 0.4 \quad 0.4]^T$.

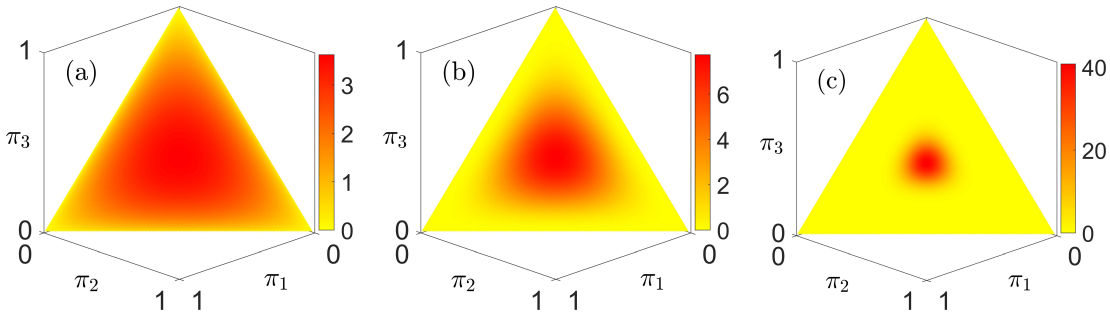

FIGURE 2.4: Example of Dirichlet distributions with base distribution $\hat{\boldsymbol{\alpha}} = [1/3 \quad 1/3 \quad 1/3]^T$, and (a) $\sigma = 5$; (b) $\sigma = 10$; (c) $\sigma = 50$.

3. *If* $E_1, E_2, \ldots \in B$, *then* $\bigcup_{i=1}^{\infty} E_i \in B$.

The pair (A, B) of a set A together with the σ-algebra B is referred to as a *measurable space*. In addition, a *finite measurable partition* of the set A is defined as a collection of sets, $\{B_i \in B \mid i = 1, \ldots, n\}$, such that $B_i \bigcap B_j = \varnothing$, if $i \neq j$; and $\bigcup_{i=1}^{n} B_i = A$. With the above definitions, a formal definition of the Dirichlet process mixture model can be given as follows:

Definition 3 (Dirichlet process [38]). *A Dirichlet process is a distribution over probability measures. Let* (A, B) *be a measurable space. Let* H *be a finite non-zero measure on the measurable space* (A, B)*, and let* α *be a positive real number. A Dirichlet process* p *with parameters* H *and* α*, denoted by* $p \sim \mathrm{DP}[\alpha, H(A)]$*, is a distribution*

of a random probability measure p, if for any finite measurable partition $\{B_i\}_{i=1}^n$ *of A, the following holds,*

$$[p(B_1), \ldots, p(B_n)]^T \sim \text{Dir}[\alpha H(B_1), \ldots, \alpha H(B_n)] \tag{2.22}$$

where "Dir" *denotes the Dirichlet distribution.*

Just like the Dirichlet distribution, the Dirichlet process can also been seen as a distribution of distributions. Because $\sum_{i=1}^n p(B_i) = 1$, samples from a Dirichlet process are discrete distributions that have the same *support* as H. The support of a function $f : \mathbb{R}^d \to \mathbb{R}$ is defined to be the set $\{\mathbf{x} \in \mathbb{R}^d \mid f(\mathbf{x}) \neq 0\}$ where f is non-zero [125]. In addition, these discrete distributions are made up of countably infinite number of weighted point masses [36]. Let $\{\beta_i\}_{i=1}^\infty$ denote the point masses and let $\{\pi_i\}_{i=1}^\infty$ denote the corresponding weight, a sample from the Dirichlet process can be expressed mathematically as,

$$p = \sum_{i=1}^{\infty} \pi_i \delta_{\beta_i}, \quad \text{where,} \quad \sum_{i=1}^{\infty} \pi_i = 1, \quad \text{and} \quad \pi_i > 0,\ i = 1, \ldots, \infty \tag{2.23}$$

where $\delta_\beta(\cdot)$ is the *Dirac's delta function* defined by the following properties:

$$\delta_\beta(x) = \begin{cases} 0, & x \neq \beta \\ \infty, & x = \beta \end{cases}, \quad \text{and} \quad \int_{-\infty}^{\infty} \delta_\beta(x) dx = 1 \tag{2.24}$$

Examples of samples from the Dirichlet processes with various parameters are shown in Fig. 2.5 and Fig. 2.6, where $\{\beta_i\}_{i=1}^\infty$ are indicated by the positions of the vertical bars and $\{\pi_i\}_{i=1}^\infty$ are shown by the heights of the vertical bars.

The parameters of the Dirichlet process have the same meaning as the those of the Dirichlet distribution. The base distribution, $H(\cdot)$, determines the mean of the Dirichlet process, as illustrated in Fig. 2.5. In Fig. 2.5, α is unchanged and the variance of the base distribution is increased, which affects the distribution of $\{\beta_i\}_{i=1}^\infty$.

The strength parameter α of the Dirichlet process controls the concentration of the sampled discrete distributions. For Dirichlet processes with large α, new clusters are more likely to be generated. Therefore, in the limit of $\alpha \to 0$, the realizations are all concentrated at a single value, while in the limit of $\alpha \to \infty$ the realizations become continuous. The effect of α of the Dirichlet process is shown in Fig. 2.6.

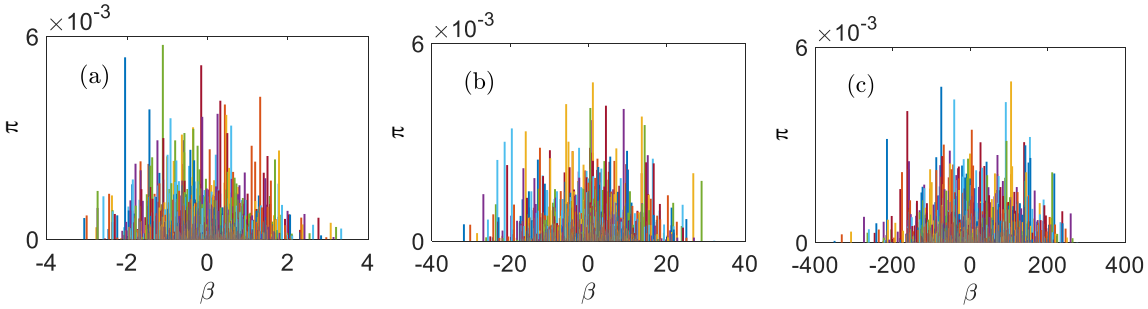

FIGURE 2.5: Example of samples from $\mathrm{DP}(\alpha, H)$, for $\alpha = 1000$ and (a) $H = \mathcal{N}(0, 1)$; (b) $H = \mathcal{N}(0, 10)$; (c) $H = \mathcal{N}(0, 100)$.

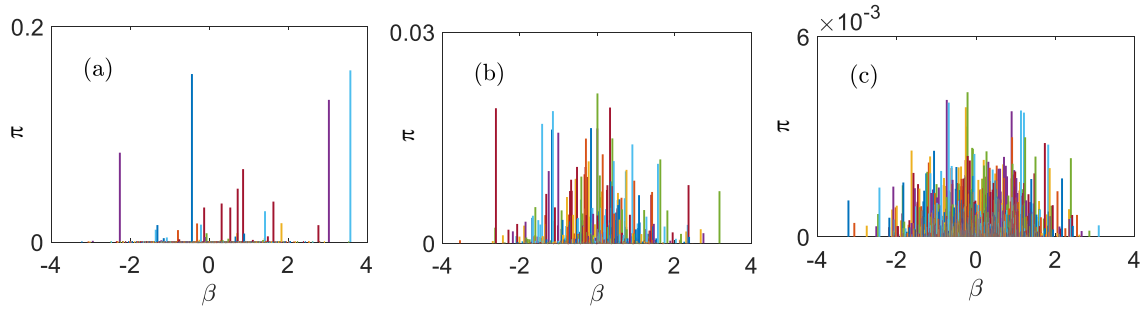

FIGURE 2.6: Example of samples from $\mathrm{DP}(\alpha, H)$, for $H = \mathcal{N}(0, 1)$ and (a) $\alpha = 10$; (b) $\alpha = 100$; (c) $\alpha = 1000$.

2.3 Dirichlet Process Gaussian Process Mixture Model

In applications where the dynamic process has a continuous domain, the Dirichlet process can not be used directly, since the samples from DPs are discrete distributions, as shown in (2.23). One popular solution to this problem is to convolve

the DP with a smooth distribution, resulting in DP mixture models, which provide an approach for estimating both the number of components in a mixture model and the parameters of the individual mixture components simultaneously from data [13, 63]. Therefore, the study for the DP mixture models, particularly the Dirichlet process-Gaussian process mixture models (DPGP-MMs), have been of interest for many years. Rasmussen et al. constructed a DPGP-MM through an extension to the Mixture of Experts model, where the individual experts are Gaussian Processes. Using an input-dependent adaptation of the Dirichlet Process, they implemented a gating network for an infinite number of Experts. Inference in this model may be done efficiently using a Markov Chain relying on Gibbs sampling. The model allows the effective covariance function to vary with the inputs, and may handle large data sets thus potentially overcoming two of the biggest hurdles with GP models, that is the high computational complexity and the limitation of stationary covariance function [54, 126]. Meeds et al. proposed an alternative infinite mixture Of Gaussian process experts, in which each component comprises a multivariate Gaussian distribution over an input space, and a Gaussian Process model over an output space. The model is neatly able to deal with non-stationary covariance functions, discontinuities, multi-modality and overlapping output signals. The work is similar to [126]; however, they used a full generative model over input and output space rather than just a conditional model, to deal with incomplete data, to perform inference over inverse functional mappings as well as for regression, and also leads to a more powerful and consistent Bayesian specification of the effective gating network for the different experts [127]. Yuan et al. presented in their work a new generative mixture of experts model. Each expert is still a Gaussian process but is reformulated by a linear model, which breaks the dependency among training outputs and enables them to use a much faster variational Bayesian algorithm for training. The resulting gating network is more flexible than previous generative approaches as inputs

for each expert are modeled by a Gaussian mixture model. The number of experts and number of Gaussian components for an expert are inferred automatically [128]. Jackson et al. presented a Bayesian technique aimed at classifying signals without prior training (clustering). The approach consists of modelling the observed signals, known only through a finite set of samples corrupted by noise, as Gaussian processes. As in many other Bayesian clustering approaches, the clusters are defined thanks to a mixture model. In order to estimate the number of clusters, they assumed a priori a countably infinite number of clusters, thanks to a Dirichlet process model over the Gaussian processes parameters. Computations are performed thanks to a dedicated Monte Carlo Markov Chain algorithm [129]. Gorur et al. studied the effect of the choice of base distributions in DPGP-MM, by compare computational efficiency and modeling performance of DPGP-MM defined using a conjugate and a conditionally conjugate base distribution. They showed that better density models can result from using a wider class of priors with no or only a modest increase in computational effort [130]. Joseph et al. used the DPGP-MM in modeling motion patterns, where the GP provides a flexible representation for each individual motion pattern, while the DP assigns observed trajectories to particular motion patterns. Both automatically adjust the complexity of the motion model based on the available data. They showed that DPGP-MM approach outperforms several parametric models on a helicopter-based car-tracking task on data grouped from the greater Boston area [95, 87].

The DPGP mixture is constructed by using a Dirichlet process as the prior of the distribution of the Gaussian process mean function, (2.1), as shown in Fig. 2.7. In order to ensure that the Dirichlet process prior is conjugate with the likelihood function, which is assumed to be Gaussian as in (2.4), the base distribution, H, of the Dirichlet process is chosen to be a known Gaussian process with zero mean, $\mathrm{GP}_0 = \mathrm{GP}[0, \phi(\cdot, \cdot)]$. For the purpose of learning the target kinematic models, $\mathcal{F}$,

GP_0 is defined for the measurable space (A, B), where $\mathcal{A}$ is chosen to be the function space $C^1(\mathcal{W})$, which is the space of continuously differentiable functions, $f : \mathcal{W} \to \mathbb{R}$, where $\mathcal{W}$ is the workspace. Then, DP-GP mixture model is formulated as follows, by adopting the infinite mixture model representation [41],

$$\begin{aligned} \{\theta_i, \boldsymbol{\pi}\} &\sim \mathrm{DP}(\alpha, \mathrm{GP}_0), \quad i = 1, \ldots, \infty \\ G_j &\sim \mathrm{Cat}(\boldsymbol{\pi}), \qquad j = 1, \ldots, N \\ \upsilon_j &\sim \mathrm{GP}(\theta_{G_j}, \phi), \quad j = 1, \ldots, N \end{aligned} \tag{2.25}$$

where "Cat" denote the categorical distribution.

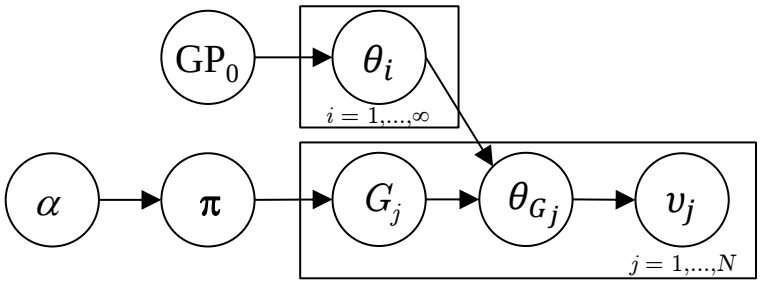

FIGURE 2.7: Graphical representation of the Dirichlet process Gaussian process mixture model.

2.4 Chapter Conclusion

Bayesian nonparametric models have recently become a popular choice for modeling dynamic processes, due to their flexibility and resistance to over-fitting/under-fitting. Rather than being targeted at improving BNP models, to date the measurements used for updating the Bayesian nonparametric models are typically obtained while in pursuit of other objectives, such as tracking or estimation of target states. Previous methods that consider the improvement of Bayesian nonparametric models assumed targets are static or the sensor field-of-view is unbounded, such that target measurements are always available. As shown in the problem formulation in Chapter 3, this

dissertation relaxes these assumptions and assumes the targets are mobile and the sensor field-of-view is bounded. Methodologies that aim at improving the accuracy of the BNP models by on-line measurements are presented in Chapters 4-6.

3

Sensor Planning Problem Formulation

Sensor planning consists of managing or controlling a sensor for the purpose of obtaining the most informative measurements in terms of a desired sensing objective. The problem class considered in this research focuses on the objective of learning the *unknown* kinematic models of multiple targets in a predefined workspace. The sensor planning problem formulation is described by first introducing all the necessary assumptions on the workspace, the sensor system, and the targets. These assumptions provide the link between the real-life applications and the abstract mathematical world. Subsequently, a mathematical description that adopts the aforementioned necessary assumptions is presented in order to rigorously define the sensor planing problem. Finally, the research goals that this dissertation proclaims to achieve are presented at the end of this chapter, which also serves as an outline of the following methodology chapters.

The targets under observation are assumed to move within a bounded workspace, $\mathcal{W}$, which is the same space that the sensor operates in, as shown in Fig. 3.1. It is assumed that the workspace is a connected and compact subset of the d-dimensional Euclidean space, such that $\mathcal{W} \subset \mathbb{R}^d$, where $d \in \mathbb{Z}^+$ is usually 2 or 3. For example,

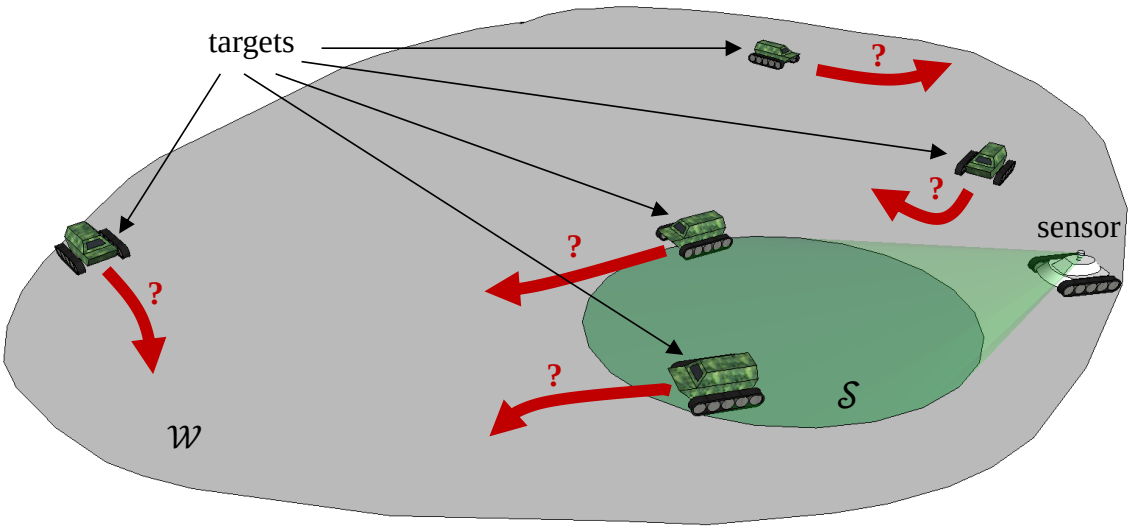

FIGURE 3.1: Schematic Diagram of the Sensor Planning Problem.

for targets moving in a flat plane, $d = 2$, and the workspace can be assumed to be a convex polygon. From the knowledge of the energy constraints of the sensor and the information about the region of interests, the workspace can be assumed known *a-priori* to the sensor.

Targets are modeled as rigid points in $\mathcal{W}$. The geometry of the target bodies is neglected because targets are assumed to be small comparing to the scale of the workspace. This assumption is valid for a wide range of real-life applications, such as the study of pedestrian movements in an indoor environment and the study of migrating animal trajectories in the wild environment. Let $\mathbf{x}_j \in \mathcal{W}$ denote the state of the jth target, for $j = 1, \ldots, N$, where N is the total number of targets in the workspace. The targets' kinematics are assumed to be modeled by a mixture of M *unknown* ordinary differential equations (ODEs),

$$\mathcal{F} \triangleq \{\mathbf{f}_1, \ldots, \mathbf{f}_M\} \tag{3.1}$$

with *unknown* mixture weights,

$$\boldsymbol{\pi} \triangleq [\pi_1 \quad \cdots \quad \pi_M]^T \tag{3.2}$$

where, $\pi_i \in [0,\ 1]$, for $i = 1, \ldots, M$, and $\sum_{i=1}^{M} \pi_i = 1$. It is worth noticing that the number of parameters (i.e. the model complexity) and the types (such as, linear, polynomial, radial basis functions) of the ODEs in $\mathcal{F}$ are also assumed *unknown*. The only available information of $\mathcal{F}$ is the domain and image of $\mathbf{f}_i(\cdot)$. Therefore, the set of ODEs need to be learned from sensor measurements adaptively. Because different targets may be described by the same ODE, the number of components, M, is also unknown and does not necessary equal to N. Let a discrete random variable $G_j \in \{1, \ldots, M\}$ denote target-ODE association, such that the event $\{G_j = i\}$ indicates that the kinematics of the jth target can be described by $\mathbf{f}_i(\cdot) \in \mathcal{F}$. Then, the state-space model for the N targets can be represented by the following autonomous systems,

$$\dot{\mathbf{x}}_j(t) = \mathbf{f}_{G_j}[\mathbf{x}_j(t)] \triangleq \mathbf{v}_j(t), \quad j = 1, \ldots, N \tag{3.3}$$

which are also known as *velocity fields* (VFs), since $\mathbf{f}_i$ is a mapping from the target position to the target velocity. The assumption in (3.3) is valid for various real-life applications, including traffic motion pattern modeling [94, 95, 104], pedestrian movement modeling [93, 77, 86, 131], semantic region modeling [92, 132], and aerial or ground robot tracking [133, 78, 80, 74, 75]. Let $\mathcal{G} \triangleq \{G_1, \ldots, G_N\}$ denote the set of target-ODE association indices for the N sensors. Then, the unknown target kinematic models can be completely described by the following sufficient statistics,

$$\mathcal{P} \triangleq \{\mathcal{F}, \mathcal{G}, \boldsymbol{\pi}, M\} \tag{3.4}$$

Notice that, point estimates of $\boldsymbol{\pi}$ and M can be derived from $\mathcal{F}$ and $\mathcal{G}$, such that,

$$\pi_i = \frac{\sum_{j=1}^{N} \mathbf{1}_{\{i\}}(G_j)}{N}, \quad i = 1, \ldots, N, \quad \text{and} \quad M = \text{card}(\mathcal{F}) \tag{3.5}$$

where $\mathbf{1}_A(x) = \begin{cases} 1, & \text{if } x \in A \\ 0, & \text{if } x \notin A \end{cases}$, is the indicator function, $\{i\}$ denotes a set with a single component i, and $\text{card}(\cdot)$ calculates the cardinality of a set. However, in full

Bayesian analysis, prior distributions can be placed on $\boldsymbol{\pi}$ and M, such that posterior distributions of $\boldsymbol{\pi}$ and M can be obtained rather than the point estimates in (3.5). For this reason, $\boldsymbol{\pi}$ and M are also included in the sufficient statistics $\mathcal{P}$ of the target kinematic models in (3.4).

The sensor is also modeled as a volume-less point in the workspace, $\mathcal{W}$. Let $\mathbf{s} \in \mathcal{A}$ denote the state of the sensor, where $\mathcal{A} \subset \mathbb{R}^q$ denote the admissible domain of the sensor state, and $q \in \mathbb{Z}^+$ is the dimension of the sensor state. The sensor state and its domain depend on the specific sensor dynamics model employed. For example, $\mathbf{s}$ can consist of the sensor position, and, accordingly, the domain of the sensor state equals to the workspace. Let $\mathbf{u} \in \mathcal{U}$ denote the control vector of the sensor, where $\mathcal{U} \subset \mathbb{R}^r$ is the admissible control space, and $r \in \mathbb{Z}^+$ is the dimension of the control input. The sensor dynamics are assumed to be discrete, linear and time-invariant (LTI). The state-space representation of the sensor dynamics is expressed as follows,

$$\mathbf{s}(k+1) = \mathbf{A}\mathbf{s}(k) + \mathbf{B}\mathbf{u}(k), \quad \mathbf{s}(k) \in \mathcal{A},\ \mathbf{u}(k) \in \mathcal{U} \tag{3.6}$$

where k is the time-index. The matrices $\mathbf{A} \in \mathbb{R}^{q \times q}$ and $\mathbf{B} \in \mathbb{R}^{r \times r}$ depend on the sensor dynamics model in consideration.

The sensor's field-of-view (FOV), denoted by $\mathcal{S} \subset \mathcal{W}$, is the subset of the workspace where the sensor is able to observe, as shown in Fig. 3.1. In other words, the sensor is able to obtain a noisy measurement of the target if and only if the target is in the FOV. For most real-life sensors, such as cameras, radars or lidars, the sensor FOV is bounded and much larger than the sensor itself. Therefore, the sensor FOV needs to be treated as a geometry object in the workspace, instead of a rigid point. Moreover, the sensor FOV is assumed to be attached rigidly to the body of the sensor, such that it is fully determined by the sensor state, $\mathbf{s}(k)$. To this end, the FOV of the sensor is modeled by a compact set, $\mathcal{S}[\mathbf{s}(k)] \subset \mathcal{W}$, which is abbreviated as $\mathcal{S}(k)$. In addition, in real-life applications, sensors often require a small but finite

constant time interval, Δt, to obtain and process new target measurements, subject to a finite sampling rate [134]. For simplicity, it is assumed that the sampling interval is equivalent to that determined by the sensor actuator in the discrete LTI model (3.6). It typically can be assumed that the sensor FOV is stationary during sampling and, therefore, measurements can be obtained from target j, provided $\mathbf{x}_j(t) \in \mathcal{S}$ for some $t \in [t_k,\ t_k + \Delta t)$. Then, during the kth sampling time interval, the jth target position can be approximated by $\mathbf{x}_j(k)$ [134]. Therefore, the sensor measurement model can be expressed as follows by considering the constant sampling interval,

$$\mathbf{m}_j(k) \triangleq [\mathbf{y}_j^T(k) \quad \mathbf{z}_j^T(k)]^T = \mathbf{h}[\mathbf{x}_j(k), \mathbf{v}_j(k)] + \boldsymbol{\nu}, \quad \text{if } \mathbf{x}_j(k) \in \mathcal{S}(k) \tag{3.7}$$

where $\boldsymbol{\nu}$ is an additive Gaussian distributed noise vector with zero mean and known covariance matrix $\text{diag}\big(\sigma_x^2\mathbf{I}_2, \sigma_v^2\mathbf{I}_2\big)$. $\text{diag}(\cdot)$ denotes the operator that places matrices on the diagonal blocks of a zero matrix [135]. If $\mathbf{x}_j(k) \notin \mathcal{S}(k)$, $\mathbf{m}_j(k)$ belongs to an empty set. It is worth noticing that $\mathbf{m}_j(k)$ consists of both the target position measurement, $\mathbf{y}_j(k) \in \mathbb{R}^d$, and the target velocity measurement, $\mathbf{z}_j(k) \in \mathbb{R}^d$. The specific form of the observation function, $\mathbf{h}(\cdot)$, depends on the type of sensor considered. Discussions of $\mathbf{h}(\cdot)$ in a variety of real-world applications can be found in Chapter 7.

For conciseness, the history of measurements of the jth target from time step k_1 to time step k_2 can be grouped in a set, such that,

$$\mathcal{M}_j(k_1, k_2) \triangleq \{\mathbf{m}_j(\ell) \mid k_1 \leqslant \ell \leqslant k_2\} \tag{3.8}$$

Similarly, the measurements of all the targets can also be grouped in a set, such that,

$$\mathcal{M}(k_1, k_2) \triangleq \bigcup_{j=1}^{N} \mathcal{M}_j(k_1, k_2) \tag{3.9}$$

Notice that, $\mathcal{M}_j(k) \triangleq \mathcal{M}_j(1, k)$ and $\mathcal{M}(k) \triangleq \mathcal{M}(1, k)$ will be used as short notations of the measurement history from the initial time to the current time. Then,

the posterior knowledge about the target kinematic models can be described by the joint distribution of the random variables in $\mathcal{P}$, which is $p[\mathcal{P}|\mathcal{M}(k)]$. Because the sensor planning problem aims at *learning* the target kinematic models, the instant reward, denoted by $\mathscr{L}[\mathbf{s}(k), \mathbf{u}(k)|\mathcal{M}(k)]$, should be defined as the amount of *information* brought about by the sensor measurements for improving the accuracy of the distribution of the statistics. In addition, the reward function, $\mathscr{L}$, should be *non-myopic*, which means that it considers all previous measurements, as indicated by the conditioning of $\mathcal{M}(k)$. Since the true distribution of the sufficient statistics of the target kinematics model is not known, no terminal cost or reward is considered. Then, the sensor planning problem can be formulated as the following optimal control problem:

Problem 1 (Sensor Planning)**.** *Given the reward function, $\mathscr{L}[\mathbf{s}(k), \mathbf{u}(k)|\mathcal{M}(k)]$, that evaluates the information value of the sensor state, $\mathbf{s}(k) \in \mathcal{A}$, and control vector, $\mathbf{u}(k) \in \mathcal{U}$, for learning the sufficient statistics of the target kinematic models, $\mathcal{P}$, find the optimal control, $\mathbf{u}^*(k)$, that maximizes the reward subject to the sensor dynamic constraints for a finite control horizon K,*

$$\begin{aligned} \max_{\mathbf{u}(k)\in\mathcal{U}} \quad & J = \sum_{k=1}^{K} \mathscr{L}[\mathbf{s}(k), \mathbf{u}(k)|\mathcal{M}(k)] \\ \text{s. t.} \quad & \mathbf{s}(k+1) = \mathbf{A}\mathbf{s}(k) + \mathbf{B}\mathbf{u}(k) \\ & \mathbf{s}(0) = \mathbf{s}_0, \quad \mathbf{s}(k) \in \mathcal{A}, \quad \mathbf{u}(k) \in \mathcal{U} \end{aligned} \tag{3.10}$$

where $\mathbf{s}_0$ is the initial sensor state.

3.1 Chapter Conclusion

From the sensor planning problem formulation summarized in Problem 1, it can be seen that the research goals of the sensor planning problem are threefold: (i) Determine a highly flexible target kinematics model that is able to adjust its complexity

and parameters according to the sensor measurements obtained over time adaptively; (ii) Design a novel objective function that evaluates the information value brought about by future sensor measurements for the purpose of reducing the uncertainties in the flexible target model from step (i); (iii) Develop efficient algorithms that optimize the novel objective functions derived in step (ii) subject to the sensor dynamics and FOV constraints. The relationship between the aforementioned three research goals are demonstrated in Fig. 3.2.

By following the flow of the research goals presented in Fig. 3.2, methodologies to the sensor planning problem can be presented in steps. To this end, Fig. 3.2 can also be treated as the outline of the subsequent methodology chapters (Chapters 4-6), where every methodology chapter addresses one research goal: Chapter 4 presents several novel flexible target kinematics models developed from Bayesian nonparametric models. Then, Chapter 5 derives new information theoretic functions for evaluating the utility of the sensor control in terms of improving the novel Bayesian nonparametric target kinematics models. Finally, efficient sensor control algorithms are presented in Chapter 6, that optimize the novel information value functions subject to the sensor dynamics and FOV constraints. Simulations and experimental results of real life applications of the sensor planning problem are discussed in Chapter 7, to demonstrate the efficiency of the proposed sensor planning algorithms.

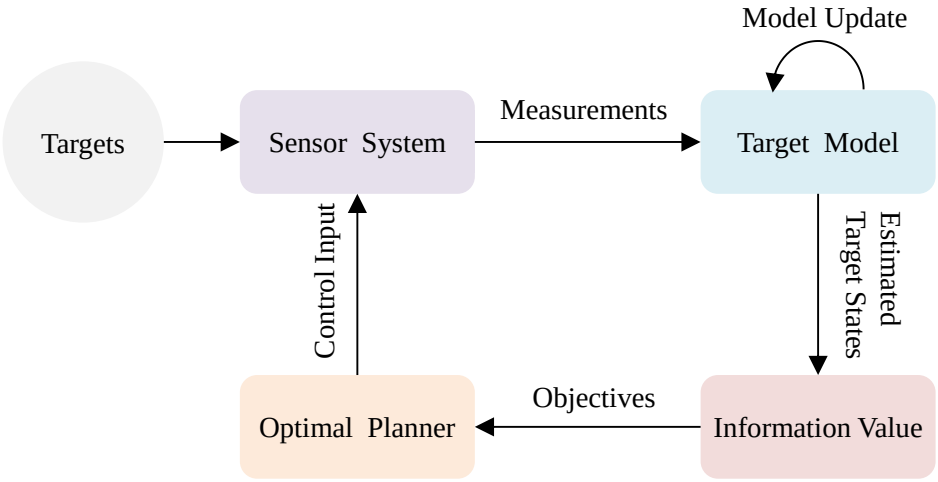

FIGURE 3.2: Block diagram of sensor planning.

4

Bayesian Nonparametric Target Modeling

Bayesian nonparametric models have been shown efficient at representing multiple dynamic processes adaptively from data, such as clinical identification [136], and gene expression time series analysis [137]. When data becomes available over time, parameters of Bayesian nonparametric models are expanded or compacted incrementally, as needed, to avoid growing the model dimensionality indefinitely as the size of the database increases. Due to the same reason, Bayesian nonparametric models, such as the Dirichlet process Gaussian process (DPGP) mixtures, have been extensively applied for modeling the kinematics of the mobile targets under surveillance, such as the movements of pedestrians [87], the patterns of ground transportation [132], and the trajectories of migrating animals [89]. Because of these characteristics, Bayesian nonparametric target kinematics models are particularly useful in lifelong learning and sensing problems, and present the opportunity for planning the measurement sequence so as to optimize the value of future data

In order to utilize the proposed Bayesian nonparametric target kinematics models for sensor planning, three problems need to be studied: *inference*, *prediction* and *filtering*. In Bayesian analysis, *inference* refers to learning the posterior distributions

or optimal values of model parameters. *Prediction* refers to the problem of deriving information about the target states at some time in the future by using data measured up to and including the current time [138]. Most prediction problems consider one-step ahead estimations. However, multiple steps predictions are of interest for finite-horizon sensor planning problems when the constraints of the sensor dynamics can not be neglected. Last but not least, *filtering* is defined as the operation that involves the estimation of target states at the current time by using data measured up to and including the current time [139].

4.1 GP Target Kinematics Model

Because the Gaussian process can be treated as a distribution of functions, a single GP can be used to model a class of target kinematics, $\mathbf{f}_i : \mathcal{W} \to \mathbb{R}^d$, for $i = 1, \ldots, M$, where M is the total number of functions in $\mathcal{F}$. Recall from (3.3) that the target kinematics are assumed to be velocity fields (VFs), that map the target position to the target velocity. Then, a natural choice of the training input and output for the GP target kinematics model consists of the measurement of the target position, $\mathbf{y}_j$, and the corresponding measurement of the target velocity at that position, $\mathbf{z}_j$, as defined in (3.7). Since the velocity measurement, $\mathbf{z}_j$, is d-dimensional, a *multioutput* Gaussian process should be applied, where the mean and covariance function are generalized from the definition of the single-output Gaussian process defined in (2.1)-(2.2), as follows,

$$\boldsymbol{\theta}_i(\mathbf{x}_j) = \mathbb{E}_{\mathbf{v}_j}[\mathbf{f}_i(\mathbf{x}_j)], \quad \forall \mathbf{x}_j \in \mathcal{W} \tag{4.1}$$

$$\boldsymbol{\phi}_i(\mathbf{x}_j, \mathbf{x}'_j) = \mathbb{E}_{\mathbf{v}_j}\left\{[\mathbf{f}_i(\mathbf{x}_j) - \boldsymbol{\theta}_i(\mathbf{x}_j)][\mathbf{f}_i(\mathbf{x}'_j) - \boldsymbol{\theta}_i(\mathbf{x}'_j)]^T\right\}, \quad \forall \mathbf{x}_j, \mathbf{x}'_j \in \mathcal{W} \tag{4.2}$$

for the ith velocity field, where $\mathbb{E}_{\mathbf{v}_j}[\cdot]$ denotes the expectation operator with respect to the velocity vector $\mathbf{v}_j$. For simplicity, it is assumed that the elements of $\mathbf{v}_j$ are independent, such that $\boldsymbol{\phi}_i$ is a diagonal and positive-definite matrix. It is

worth noticing that, by adopting this assumption, the multioutput Gaussian process formulation is equivalent to modeling every dimension of $\mathbf{f}_i(\cdot)$ by an independent single-output Gaussian processes, as suggested by various works [94, 95, 93]. If this assumption is violated, the following approach can be applied by carefully choosing the off-diagonal terms to reflect the velocity elements' correlation.

4.1.1 Inference with GP Target Kinematics Model

The GP regression introduced in Section 2.1.1 provides an effective technique for predicting the output of a function conditioned on all previous measurements. The target-VF associations, $\mathcal{G}$, can be learned by the DPGP inference algorithm presented in Section 4.2.1, and are considered as known information for the purpose of GP target kinematics modeling. Without loss of generality, it can be assumed that the target kinematics can be modeled by the ith VF. Now, let $\mathbf{Y}_i(k) = [\mathbf{y}_1^T(\cdot) \quad \mathbf{y}_2^T(\cdot) \cdots]^T$ and $\mathbf{Z}_i(k) = [\mathbf{z}_1^T(\cdot) \quad \mathbf{z}_2^T(\cdot) \cdots]^T$ denote two vectors containing all noisy measurements of target position and velocity, respectively, that have been associated with the ith velocity field up to time k. It follows that distribution of the target velocity at any target position, $\mathbf{x}_j(k) \in \mathcal{W}$, conditioned on all previous measurements of the N target, $\mathcal{M}(k)$, is a Gaussian distribution,

$$p_V\big(\mathbf{v}_j(k)|\mathcal{M}(k), G_j = i\big) = f_G\big(\mathbf{v}_j(k); \tilde{\boldsymbol{\mu}}_i(k), \tilde{\boldsymbol{\Sigma}}_i(k)\big) \tag{4.3}$$

where

$$\tilde{\boldsymbol{\mu}}_i(k) = \boldsymbol{\Phi}\left[\mathbf{x}_j(k), \mathbf{Y}_i(k)\right] \mathbf{A}^{-1} \mathbf{Z}_i(k) \tag{4.4}$$

$$\tilde{\boldsymbol{\Sigma}}_i(k) = \boldsymbol{\Phi}[\mathbf{x}_j(k), \mathbf{x}_j(k)] - \boldsymbol{\Phi}[\mathbf{x}_j(k), \mathbf{Y}_i(k)] \mathbf{A}^{-1} \boldsymbol{\Phi}[\mathbf{Y}_i(k), \mathbf{x}_j(k)] \tag{4.5}$$

$$\mathbf{A} \triangleq \boldsymbol{\Phi}[\mathbf{Y}_i(k), \mathbf{Y}_i(k)] + \sigma_n^2 \mathbf{I}_{2k} \tag{4.6}$$

the cross-covariance matrix $\boldsymbol{\Phi}$ is defined in (2.8), and f_G denotes the probability density function of a multivariate Gaussian distribution (see Appendix A.1).

After the GP target kinematics model is learned by the inference algorithm in (4.3), distributions of the target positions in the future time can be predicted by using the learned model. The following section introduces an efficient approach for the prediction with GP target kinematics model, which utilizes particle filter techniques.

4.1.2 Prediction with GP Target Kinematics Model

Prediction of the target position in the future time is essential for sensor planning, since the it allows the sensor to anticipate the movements of the targets to improve the performance of the planning algorithm. To this end, an efficient recursive method for the prediction with respect to the GP target kinematics model is proposed by utilizing the particle filter technique, which approximates continuous probability density function by using large number of samples. The idea of the recursive prediction method is to calculate the target position distribution at the next time step by the convolution of the target velocity distribution with the current target position distribution,

$$p_X\big(\mathbf{x}_j(k+1)|\mathcal{M}(k), G_j = i\big) = \tag{4.7}$$
$$\int_{\mathbb{R}^2} p_V\big(\mathbf{v}_j(k)|\mathcal{M}(k), G_j = i\big) p_X\big(\mathbf{x}_j(k+1) - \mathbf{v}_j(k)\Delta t|\mathcal{M}(k), G_j = i\big) d\mathbf{v}_j(k)$$

where Δt is the interval between two time steps. Notice that (4.7) provides the recursive relation between the target positions at consecutive time steps, and can be applied iteratively to predict the target position distributions in all future time steps. However, using (4.7) directly becomes computationally intractable in a few steps due to the convolution, since the target velocity distribution, $p_V(\cdot)$, depends on the position of the target through (4.4)-(4.6). Therefore, approximation algorithms need to developed in order to expedite the prediction. Since particle filters have been shown effective at predicting target states described by non-linear, non-Gaussian

systems, they are utilized to as an efficient approximation to the recursive prediction in (4.7) as follows.

The efficient prediction algorithm for the GP target kinematics model is developed by using the special case when the convolution in (4.7) can be calculated analytically. The special case assumes that the current target position is given, and the result is another Gaussian distribution,

$$p_X\big(\mathbf{x}_j(k+1)|\mathcal{M}(k), G_j = i\big) = f_G\big(\mathbf{x}_j(k+1); \mathbf{x}_j(k) + \tilde{\boldsymbol{\mu}}_i(k)\Delta t, \tilde{\boldsymbol{\Sigma}}_i(k)\Delta t^2\big) \tag{4.8}$$

where $\tilde{\boldsymbol{\mu}}_i(k)$ and $\tilde{\boldsymbol{\Sigma}}_i(k)$ are the mean and covariance of the target velocity distribution, defined in (4.4) and (4.5). Based on the discover in (4.8), a novel Gaussian process particle filter (GP-PF) can be developed to expedite the prediction of the target position distributions in the future time. In addition, in order to avoid using large number of particles, the GP-PF treats Gaussian distributions as "particles", and assumes that the distribution of the target position can be approximated by a Gaussian mixture model,

$$p_X\big(\mathbf{x}_j(k)|\mathcal{M}(k), G_j = i\big) \approx \sum_{\imath=1}^{n} \beta_\imath f_G\big(\mathbf{x}_j(k); \boldsymbol{\eta}_\imath(k), \boldsymbol{\Lambda}_\imath(k)\big) \tag{4.9}$$

where n is the number of components in the Gaussian mixture model, $\{\boldsymbol{\eta}_\imath(k),\, \boldsymbol{\Lambda}_\imath(k)\}$ are the mean and covariance matrix of the $\imath$th Gaussian component, and $\{\beta_\imath\}_{\imath=1}^{n}$ are the mixture weights that satisfy $\beta_\imath > 0$ and $\sum_{\imath=1}^{n} \beta_\imath = 1$ [140, 141]. The GP-GPF prediction based on (4.9) first draws S independent and identically distributed (i.i.d.) samples from every component of the Gaussian mixture model (4.9), such that,

$$\boldsymbol{\chi}_\imath^{(s)} \sim \mathcal{N}\big(\boldsymbol{\eta}_\imath(k), \boldsymbol{\Lambda}_\imath(k)\big), \quad s = 1, \ldots, S,\ \imath = 1, \ldots, n \tag{4.10}$$

These samples are then propagated by one time step using (4.8),

$$\hat{\boldsymbol{\chi}}_\imath^{(s)} \sim \mathcal{N}\big(\boldsymbol{\chi}_\imath^{(s)} + \boldsymbol{\mu}_\imath^{(s)}\Delta t, \boldsymbol{\Sigma}_\imath^{(s)}\Delta t^2\big), \quad s = 1, \ldots, S,\ \imath = 1, \ldots, n \tag{4.11}$$

where $\boldsymbol{\mu}_\imath^{(s)}$ and $\boldsymbol{\Sigma}_\imath^{(s)}$ are defined by substituting $\mathbf{x}_j(k)$ with $\boldsymbol{\chi}_\imath^{(s)}$ in (4.4) and (4.5), respectively. The final step of the GP-GPF prediction obtains the predicted mean and covariance matrix of every Gaussian component using the propagated samples in (4.11), such that,

$$\hat{\boldsymbol{\eta}}_\imath(k+1) = \frac{1}{S}\sum_{s=1}^{S}\hat{\boldsymbol{\chi}}_\imath^{(s)} \tag{4.12}$$

$$\hat{\boldsymbol{\Lambda}}_i(k+1) = \frac{1}{S}\sum_{s=1}^{S}[\hat{\boldsymbol{\chi}}_\imath^{(s)} - \hat{\boldsymbol{\eta}}_\imath(k+1)][\hat{\boldsymbol{\chi}}_\imath^{(s)} - \hat{\boldsymbol{\eta}}_\imath(k+1)]^T \tag{4.13}$$

for $\imath = 1, \ldots, n$. The predicted target position distribution at the next time step is then approximated by the Gaussian mixture model with the prorogated mean and covariance matrices,

$$p_X\big(\mathbf{x}_j(k+1)|\mathcal{M}(k), G_j = i\big) \approx \sum_{\imath=1}^{n}\hat{\beta}_\imath f_G\big(\mathbf{x}_j(k+1); \hat{\boldsymbol{\eta}}_\imath(k+1), \hat{\boldsymbol{\Lambda}}_\imath(k+1)\big) \tag{4.14}$$

where the estimated mixture weights are not changed, such that $\hat{\beta}_\imath = \beta_\imath$, for $\imath = 1, \ldots, n$. By repeating the steps from (4.9)-(4.14) multiple times, the prediction of the target position can be obtained for a finite horizon in the future. To summarize, the pseudocode of the GP-GPF prediction algorithm is compiled in Algorithm 1.

Algorithm 1 Gaussian process Particle Filter Time Update (GPPF-TU)

Input: Gaussian mixture model parameters, $\{\beta_\imath, \boldsymbol{\eta}_\imath(k), \boldsymbol{\Lambda}_\imath(k)\}_{\imath=1}^{n}$; Prediction time horizon, K.

Output: Predicted Gaussian mixture model parameters, $\{\hat{\beta}_\imath, \hat{\boldsymbol{\eta}}_\imath(k+\ell), \hat{\boldsymbol{\Lambda}}_\imath(k+\ell)\}_{\imath=1}^{n}$, for $\ell = 1, \ldots, K$.

1: **for** $\ell = 1, \ldots, K$ **do**
2: Draw S samples, $\{\boldsymbol{\chi}_\imath^{(s)}\}_{s=1}^{S}$, from (4.10), for $\imath = 1, \ldots, n$.
3: Obtain the propagated samples, $\{\hat{\boldsymbol{\chi}}_\imath^{(s)}\}_{s=1}^{S}$, by (4.11), for $\imath = 1, \ldots, n$.
4: Update weights as $\hat{\beta}_\imath = \beta_\imath$, for $\imath = 1, \ldots, n$.
5: Obtain the predicted mean, $\hat{\boldsymbol{\eta}}_\imath(k+\ell)$, and covariance $\hat{\boldsymbol{\Lambda}}_\imath(k+\ell)$, by (4.12) and (4.13), respectively, for $\imath = 1, \ldots, n$.
6: **end for**

4.1.3 Filtering with GP Target Kinematics Model

Bayesian filtering generally consists of an iterative time update-measurement update process [138]. In the time update step, one-step ahead prediction of the target state is calculated, and in the measurement update step, new measurements are incorporated to correct the predicted target state. For the GP target kinematics model, the time update step can be performed by the GP particle filter time update in Algorithm 1 with the length of the time horizon equal to one. Assuming the current time step is k, after the time update step, the target position distribution is estimated by a Gaussian mixture model with parameters, $\{\hat{\beta}_\imath, \hat{\boldsymbol{\eta}}_\imath(k+1), \hat{\boldsymbol{\Lambda}}_\imath(k+1)\}_{\imath=1}^{n}$.

The measurement update step of the GP-PF consists of two cases, depending on whether the data vector $\mathbf{m}_j(k+1)$ belongs to the empty set at time step $(k+1)$. If the sensor successfully obtains a measurement of the jth target at time step $(k+1)$, S i.i.d. samples can be drawn from the Gaussian mixture model,

$$\boldsymbol{\chi}_\imath^{(s)} \sim \mathcal{N}\big(\hat{\boldsymbol{\eta}}_\imath(k+1), \hat{\boldsymbol{\Lambda}}_\imath(k+1)\big), \quad s = 1, \ldots, S, \ \imath = 1, \ldots, n \tag{4.15}$$

After the samples are drawn, the measurement update step adjusts the weights and parameters of the Gaussian components by incorporating the measurement probability,

$$\begin{aligned} \gamma_\imath^{(s)} &\triangleq p(\mathbf{m}_j(k+1)|\boldsymbol{\chi}_\imath^{(s)}) \\ &= f_G(\mathbf{y}_j(k+1); \boldsymbol{\chi}_\imath^{(s)}, \sigma_x^2\mathbf{I}_2) \times f_G(\mathbf{z}_j(k+1); \tilde{\boldsymbol{\mu}}_i(k+1), \tilde{\boldsymbol{\Sigma}}_i(k+1)) \end{aligned} \tag{4.16}$$

for $\imath = 1, \ldots, n$ and $s = 1, \ldots, S$. Then, the mean vectors and covariance matrices of the Gaussian components are updated as the weighted sample mean and covariance matrices,

$$\boldsymbol{\eta}_\imath(k+1) = \frac{1}{\sum_{s=1}^{S} \gamma_\imath^{(s)}} \sum_{s=1}^{S} \gamma_\imath^{(s)} \boldsymbol{\chi}_\imath^{(s)} \tag{4.17}$$

$$\mathbf{\Lambda}_\imath(k+1) = \frac{1}{\sum_{s=1}^{S}\gamma_\imath^{(s)}}\sum_{s=1}^{S}\gamma_\imath^{(s)}[\boldsymbol{\chi}_\imath^{(s)} - \boldsymbol{\eta}_\imath(k+1)][\boldsymbol{\chi}_\imath^{(s)} - \boldsymbol{\eta}_\imath(k+1)]^T \tag{4.18}$$

for $\imath = 1, \ldots, n$. The previous steps from (4.15) to (4.18) can be applied to both the non-empty measurement case and the empty measurement case. However, cautious distinction is needed for updating the mixture weights. When the sensor obtains the measurement at time step $(k+1)$, the mixture weights can be updated by,

$$\beta_\imath = \frac{\hat{\beta}_\imath \sum_{s=1}^{S}\gamma_\imath^{(s)}}{\sum_{\imath=1}^{n}\sum_{s=1}^{S}\hat{\beta}_\imath\gamma_\imath^{(s)}}, \quad \imath = 1, \ldots, n \tag{4.19}$$

When no measurement is obtained, the calculation in (4.19) is not able to correctly reflect the rejection rate during the rejection sampling in step (4.15). However, the rejection rate is also related to the mixture weight in the sense that higher rejection rate means that the target position is less likely to be described by that Gaussian component. One solution is to numerically estimate the rejection rate by counting the number of successful and rejected samples and multiply this rate to every corresponding $\hat{\beta}_\imath$. Large numerical errors exist if the number of required samples, S, is small. On the other hand, if a large number of samples are required to represent every Gaussian component, the rejection sampling may take a long time to finish if the major part of the proposal density (4.15) overlaps with the sensor FOV, $\mathcal{S}(k+1)$. In order to avoid these problems in the case of empty measurement, the mixture weights can be updated by calculating the probability of the target lying out of the sensor FOV analytically, such that,

$$\begin{aligned}\beta_\imath &= \frac{\hat{\beta}_\imath}{c}p\big(\mathbf{x}_j \notin \mathcal{S}(k+1)|\hat{\boldsymbol{\eta}}_\imath(k+1), \hat{\mathbf{\Lambda}}_\imath(k+1)\big) \\ &= \frac{\hat{\beta}_\imath}{c}\int_{\mathcal{W}\setminus\mathcal{S}(k)} f_G(\mathbf{x}_j; \hat{\boldsymbol{\eta}}_\imath(k+1), \hat{\mathbf{\Lambda}}_\imath(k+1))d\mathbf{x}_j\end{aligned} \tag{4.20}$$

for $\imath = 1, \ldots, n$, where c is the normalizing constant that makes the summation of

$\beta_\imath$ equal to one. Notice that the integral in (4.20) can be calculated by the error function for rectangle-shaped sensor FOV with edges parallel to the x and y axes of a two-dimensional workspace. To summarize this section, the GP particle filter measurement update is summarized in Algorithm 2.

Algorithm 2 Gaussian Process Particle Filter Measurement Update (GPPF-MU)

Input: Predicted Gaussian mixture model parameters, $\{\hat{\beta}_\imath, \hat{\boldsymbol{\eta}}_\imath(k+1), \hat{\boldsymbol{\Lambda}}_\imath(k+1)\}_{\imath=1}^{n}$; Sensor measurement, $\mathbf{m}_j(k+1)$; Sensor FOV, $\mathcal{S}(k+1)$.

Output: Updated Gaussian mixture model parameters, $\{\beta_\imath, \boldsymbol{\eta}_\imath(k+1), \boldsymbol{\Lambda}_\imath(k+1)\}_{\imath=1}^{n}$.

1: Draw S samples, $\{\boldsymbol{\chi}_\imath^{(s)}\}_{s=1}^{S}$, using rejection sampling with proposal density (4.15), and reject the samples if $\boldsymbol{\chi}_\imath^{(s)} \in \mathcal{S}(k)$, for $\imath = 1, \ldots, n$.
2: Compute measurement weights, $\gamma_\imath^{(s)}$, by (4.16), for $\imath = 1, \ldots, n$, $s = 1, \ldots, S$.
3: Estimate the mean, $\boldsymbol{\eta}_\imath(k+1)$, and covariance matrix, $\boldsymbol{\Lambda}_\imath(k+1)$, by (4.17) and (4.18), respectively, for $\imath = 1, \ldots, n$.
4: **if** $\mathbf{m}(k+1) \notin \varnothing$ **then**
5: Update the mixture weights, $\{\beta_\imath\}_{\imath=1}^{n}$, by (4.19).
6: **else**
7: Update the mixture weights, $\{\beta_\imath\}_{\imath=1}^{n}$, by (4.20).
8: **end if**

4.2 Multiple Classes of Target Kinematics

When the targets' movements display multiple patterns, one Gaussian process is not enough for describing all the target kinematics. Therefore, a mixture model with GPs as the components should be applied. In addition, when the appropriate number of GPs can not be determined *a priori*, the use of infinite mixtures is appealing since it bypasses the need to determine the "correct" number of components in a finite mixture model. Among all the infinite mixture models, the Dirichlet process mixture model is particularly suitable for modeling the target kinematics since it leads to a few of the components dominating. Therefore, the Dirichlet process Gaussian process mixture model introduced in Chapter 2.3 is utilized to model multiple classes of target kinematics. The inference, prediction and filtering algorithms are also of interest for applying the DPGP target kinematics model in the sensor planning problem.

Therefore, this section first discusses the inference technique based on Markov Chain Monte Carlo sampling in Section 4.2.1. The prediction and filtering algorithms are then presented in 4.2.2.

4.2.1 Inference with DPGP Target Kinematics Model

Exact computation of the posterior DPGP mixture model can become infeasible when there are more than a few measurements, even with numerical approximations, since the joint distribution of the posterior DPGP mixture model does not have an analytical form. However, Markov Chain Monte Carlo (MCMC) sampling algorithms have been developed for sampling from the posterior distribution of the parameters of the component distributions and/or of the associations of mixture components with observations, by simulating a Markov chain that has the posterior as its equilibrium distribution [41]. These MCMC algorithms, especially the methods based on Gibbs sampling for conjugate priors, have made the implementation of Dirichlet process mixture models computationally feasible for problems with moderate or large number of measurements [118, 114, 116, 117, 40]. To this end, this section presents the MCMC algorithm that is able to sample from the posterior DPGP mixture for the modeling of multiple classes of target kinematics.

Since the measurement model is conjugate with the base distribution of the Dirichlet process, Gibbs sampling (or Gibbs sampler) can be used to sample from the posterior DPGP mixture model. The idea in Gibbs sampling is to generate posterior samples by sweeping through each variable (or block of variables) to sample from its conditional distribution with the remaining variables fixed to their current values [22]. For example, consider the generic Gibbs samplers that draws samples from the joint distribution of d random variables, $\{X_i\}_{i=1}^{d}$, and let $\{x_i^{(s)}\}_{i=1}^{d}$ denote the samples drawn in the sth iteration. The Gibbs sampler starts by randomly setting the initial values of the random variables, $\{x_i^{(0)}\}_{i=1}^{d}$. In every subsequent iteration,

the values of the random variables are updated by sampling from their conditional distributions. It is worth noticing that although the initial values of the samples are chosen randomly, the Ergodic theorem guarantees that the stationary distribution of the samples generated by the Gibbs sampler is the target joint posterior distribution [142]. Since the samples at the beginning of the iterations may not represent the joint posterior distribution truthfully, these samples are often discarded. This operation is referred to as the *burn-in* of the Gibbs sampler, and the number of discarded samples, denoted by $L_b \in \mathbb{Z}^+$, is referred to as the burn-in period. In addition, nearby samples draw by the Gibbs sampler are correlated, since the conditional distributions depend on the values of other variables from the previous iteration. Therefore, if independent samples are desired, the Gibbs sampler usually records the samples by every L_s iterations, where $L_s \in \mathbb{Z}^+$ is referred to as the *step size* of the Gibbs sampler. To summarize, the generic Gibbs samplers can be presented by Algorithm 3.

Algorithm 3 Generic Gibbs Sampler

Input: Conditional distributions, $p\big(X_i \mid x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_d\big)$; Burn-in period length, L_b; Step size, L_s.

Output: Samples from the joint distribution, $\{x_i^{(s)}\}_{i=1}^d$, for $s = 1, 2, \ldots$

```
Initialize {x_i^(0)}_{i=1}^d randomly.
for s = 1, 2, ..., do
    for i = 1, ..., d do
        x_i^(s) ~ p(X_i | x_1^(s), ..., x_{i-1}^(s), x_{i+1}^(s-1), ..., x_d^(s-1))
    end for
    Record {x_i^(s)}_{i=1}^d, if (s - L_b)/L_s ∈ Z^+.
end for
```

The Gibbs sampler in Algorithm 3 can be used to draw samples from the posterior DPGP mixture model (2.25) that describes multiple classes of target kinematics. It is worth noticing that (2.25) is the infinite mixture representation of the DPGP mixture model, that allows sampling of the target-VF association indices, $\{G_j\}_{j=1}^N$, directly by integrating out the mixture weights, $\boldsymbol{\pi}$. The advantage of sampling target-VF

association indices directly is that $\{\boldsymbol{\theta}_i\}_{i=1}^{M}$ can be used to represent the GP mean functions of the classes of target kinematics rather than those of every individual target kinematics. Therefore, when a GP mean function, $\boldsymbol{\theta}_i$, is changed, all the target kinematics associated with that class will be updated simultaneously, which expedites the convergence of the Markov chain in the Gibbs sampler greatly [41]. In order to calculate the posterior distributions of the target-VF association indices, the prior distributions and the likelihood functions of $\{G_j\}_{j=1}^{N}$ need to be studied. Since the number of GP mean functions in the DPGP mixture model is infinite, it is infeasible to explicitly represent all the GP mean functions. Therefore, the Gibbs sampling is only performed on the target-VF association indices corresponding to the GP mean functions associated with some measurements. The remaining infinite number of target-VF association indices can be grouped together to represent the case when the target is associated with a new VF that has never been seen before. Following this simplification, the prior distribution of the target-VF association indices is,

$$p\big(G_j = i \mid \{G_j\}^c\big) = \begin{cases} \dfrac{N_i}{N-1+\alpha}, & i = 1, \ldots, M \\ \dfrac{\alpha}{N-1+\alpha}, & i = M+1 \end{cases} \tag{4.21}$$

where $\{G_j\}^c \triangleq \mathcal{G} \backslash \{G_j\}$ denotes the complement of $\{G_j\}$, and N_i is the number of targets assigned to the ith VF without considering the jth target.

The likelihood function can be obtained by the marginalization over the function evaluations,

$$\mathbf{V}_j(k) \triangleq [\mathbf{v}_j^T(1) \quad \cdots \quad \mathbf{v}_j^T(k)]^T \tag{4.22}$$

where $\mathbf{v}_j \triangleq \mathbf{f}_i(\mathbf{x}_j)$ is defined in (3.3). The measurement history of a target, $\mathcal{M}_j(k)$, is treated as a group of data, since every target is assumed to be governed by one kinematics model for its entire movement in the workspace. Let $\tilde{\mathbf{Y}}_j(k) \triangleq [\mathbf{y}_j^T(1) \quad \cdots \quad \mathbf{y}_j^T(k)]^T$ and $\tilde{\mathbf{Z}}_j(k) \triangleq [\mathbf{z}_j^T(1) \quad \cdots \quad \mathbf{z}_j^T(k)]^T$ denote the aggregations

of the position measurements and velocity measurements of the jth target. Notice that $\tilde{\mathbf{Y}}_j(k)$ and $\tilde{\mathbf{Z}}_j(k)$ are defined with respect to the jth target, while $\mathbf{Y}_i(k)$ and $\mathbf{Z}_i(k)$ are defined for the ith VF. From the measurement model (3.7) and Gaussian process regression, it follows that,

$$\begin{aligned} p\big(\mathcal{M}_j|\boldsymbol{\theta}_i\big) &= \int_{\mathbf{V}_j} p\big(\mathcal{M}_j|\mathbf{V}_j\big)p\big(\mathbf{V}_j|\boldsymbol{\theta}_i\big)d\mathbf{V}_j \\ &= \int_{\mathbf{V}_j} f_G(\tilde{\mathbf{Z}}_j;\mathbf{0},\sigma_v^2\mathbf{I})f_G[\mathbf{V}_j;\boldsymbol{\theta}_i(\tilde{\mathbf{Y}}_j),\boldsymbol{\Phi}_i(\tilde{\mathbf{Y}}_j,\tilde{\mathbf{Y}}_j)]d\mathbf{V}_j \\ &= f_G[\tilde{\mathbf{Z}}_j;\boldsymbol{\theta}_i(\tilde{\mathbf{Y}}_j),\boldsymbol{\Phi}_i(\tilde{\mathbf{Y}}_j,\tilde{\mathbf{Y}}_j)+\sigma_v^2\mathbf{I}] \end{aligned} \tag{4.23}$$

where $\boldsymbol{\Phi}_i$ is defined by substituting $\phi(\cdot,\cdot)$ with $\boldsymbol{\phi}_i(\cdot,\cdot)$ in (2.8). Notice that $p(\mathcal{M}_j|\boldsymbol{\theta}_i)$ is equivalent to $p(\mathcal{M}_j|\{\boldsymbol{\theta}\}_{i=1}^M, G_j = i)$, and is used as a short notation. The GP covariance function, $\boldsymbol{\phi}_i(\cdot,\cdot)$, defined in (4.2), can also be adjusted adaptively to the measurements. A point estimate of the GP covariance function is computed by conditioning on the target measurements belonging to every class, such that,

$$\boldsymbol{\phi}_i(\mathbf{x},\mathbf{x}') \triangleq \boldsymbol{\phi}(\mathbf{x},\mathbf{x}') - \boldsymbol{\Phi}[\mathbf{x},\mathbf{Y}_i(k)]\{\boldsymbol{\Phi}[\mathbf{Y}_i(k),\mathbf{Y}_i(k)]+\sigma_v^2\mathbf{I}\}^{-1}\boldsymbol{\Phi}[\mathbf{Y}_i(k),\mathbf{x}'] \tag{4.24}$$

for$\forall \mathbf{x},\mathbf{x}' \in \mathcal{W}$. The likelihood function (4.23) is also well defined for $i = M+1$, when the target kinematics is assigned to a new group. In this scenario, $\boldsymbol{\theta}_{M+1}$ is obtained by drawing a new sample from GP_0 in (2.25), and $\boldsymbol{\phi}_{M+1}$ is initialized as $\boldsymbol{\phi}$. Other types of likelihood functions can also be used for the purpose of learning the target-VF associations. For example, the works in [94, 95, 87, 88, 93] assume that the measurements are independent given the GP mean function $\boldsymbol{\theta}_i$, such that the likelihood function is simplified to,

$$p\big(\mathcal{M}_j|\boldsymbol{\theta}_i\big) = f_G\left(\tilde{\mathbf{Z}}_j;\boldsymbol{\theta}_i(\tilde{\mathbf{Y}}_j),\sigma_v^2\mathbf{I}\right) \tag{4.25}$$

The computational complexity of (4.25) is $O(k)$, which is less than the $O(k^3)$ complexity of (4.23). Therefore, it is preferred when the number of measurements is large.

Experiments show that both the likelihood functions in (4.23) and (4.25) work well for real-world datasets. Since the likelihood function in (4.23) is more rigourous, it is preferred when dealing with small to moderate datasets.

Based on the above prior and likelihood functions, the posterior distribution of the target-VF association can be calculated,

$$p\big(G_j = i|\mathcal{M}, \{G_j\}^c, \{\boldsymbol{\theta}_i\}_{i=1}^M\big) = \begin{cases} c\left(\dfrac{N_i \cdot p(\mathcal{M}_j|\boldsymbol{\theta}_i)}{N-1+\alpha}\right), & i = 1, \ldots, M \\ c\left(\dfrac{\alpha \int p(\mathcal{M}_j|\boldsymbol{\theta}_i) d\mathrm{GP}_0(\boldsymbol{\theta}_i)}{N-1+\alpha}\right), & i = M+1 \end{cases} \tag{4.26}$$

where c is the normalizing constant that makes the summarization of (4.26) equal to one. The integral in (4.26) can be approximated by Monte Carlo integration [113]. Sampling $\boldsymbol{\theta}_i$ can be difficult. Since their posteriors are extremely peaked, some works suggest the use of their maximum likelihood values [94]. To this end, the posterior GP mean functions are calculated by conditioning on the target measurements belonging to that class, such that,

$$\boldsymbol{\theta}_i(\mathbf{x}) = \boldsymbol{\Phi}[\mathbf{x}, \mathbf{Y}_i(k)]\{\boldsymbol{\Phi}[\mathbf{Y}_i(k), \mathbf{Y}_i(k)] + \sigma_n^2\mathbf{I}\}^{-1}\mathbf{Z}_i(k), \quad \forall \mathbf{x} \in \mathcal{W} \tag{4.27}$$

After the conditional posterior distributions of the target-VF association variables are defined in (4.26), the Gibbs sampling algorithm in Algorithm 3 can be applied to the DPGP mixture model. The idea of the algorithm is similar to the second approach presented in [41], and is summarized by Algorithm 4 for convenience.

Algorithm 4 DPGP Gibbs Sampler (DPGP-Gibbs)

Input: Measurements, $\mathcal{M}(k)$; DP parameters, $\{\alpha, \mathrm{GP}_0\}$; Burn-in period length, L_b; Step size, L_s.

Output: Samples of target-VF association indices, $\{G_j^{(s)}\}_{j=1}^N$; Samples of GP parameters, $\{\boldsymbol{\theta}_i^{(s)}, \boldsymbol{\phi}_i^{(s)}\}_{i=1}^M$.

Initialize $\{G_j^{(0)}\}_{j=1}^N$ randomly.
for s=1,2,..., **do**
  for j=1,...,N **do**
    If $\boldsymbol{\theta}_{G_j^{(s-1)}}$ is associated with no other targets, remove it.
    Draw a new value of $G_j^{(s)}$ by (4.26).
  end for
  for i=1,...,M **do**
    Update $\boldsymbol{\theta}_i^{(s)}$ and $\boldsymbol{\phi}_i^{(s)}$ according to (4.27) and (4.24), respectively.
  end for
  Record $\{G_j^{(s)}\}_{j=1}^N$ and $\{\boldsymbol{\theta}_i^{(s)}, \boldsymbol{\phi}_i^{(s)}\}_{i=1}^M$, if $(s - L_b)/L_s \in \mathbb{Z}^+$.
end for

4.2.2 Prediction and Filtering with DPGP Target Kinematics Model

Once the posterior DPGP target kinematics model is learned from the current available measurements, it can be used to predict or estimate the target states. During the prediction or filtering phase, it can be assumed that the DP prior is not updated, since the target-VF associations are unlikely to change without obtaining a significant amount of more data. In addition, the high computational complexity of the Gibbs sampler in Algorithm 4 also disapproves updating the DP prior too often.

To this end, for the prediction and filtering of target states, the DPGP mixture model is treated as a finite Gaussian process mixture model, with parameters, $\{\pi_i, \boldsymbol{\theta}_i, \boldsymbol{\phi}_i\}_{i=1}^M$. The mixture weights, $\{\pi_i\}_{i=1}^M$, can be assumed to be proportional to the number of targets associated with a kinematics class, such that, $\pi_i = N_i/N$, for $i = 1, \ldots, M$. Then, the prediction and filtering by the DPGP target kinematics model can be achieved by a bank of GP particle filters presented in Section 4.1.2 and 4.1.3. Let $P_{ij}(k) \triangleq \{\beta_\imath, \boldsymbol{\eta}_\imath(k), \boldsymbol{\Lambda}_\imath(k) \mid G_j = i\}_{\imath=1}^n$ denote the GP-PF parameters of

the jth target corresponding to the ith VF, and let

$$w_{ij} \triangleq p(G_j = i), \quad i = 1, \ldots, M, \ j = 1, \ldots, N \tag{4.28}$$

denote the probability that the jth target follows the ith VF. Then, the prediction with respect to the DPGP target kinematics model can be performed by using the GP-PF time update in Algorithm 1 as sub-routines. The resulting prediction algorithm is presented as Algorithm 5.

Algorithm 5 DPGP Particle Filter Time Update (DPGP-PF-TU)

Input: Bank of Gaussian mixture model parameters, $\{w_{ij}, P_{ij}(k)\}_{i=1}^{M}$, for $j = 1, \ldots, N$; Prediction time horizon, K.

Output: Predicted Gaussian mixture model parameters, $\{\hat{w}_{ij}, \hat{P}_{ij}(k+\ell)\}_{i=1}^{M}$, for $j = 1, \ldots, N$, $\ell = 1, \ldots, K$.

1: $\hat{P}_{ij}(k) = P_{ij}(k)$, for $i = 1, \ldots, M$.
2: **for** $\ell = 1, \ldots, K$ **do**
3: **for** $i = 1, \ldots, M$ **do**
4: Update GP-PF parameters by Algorithm 1,
5: $\hat{P}_{ij}(k+\ell+1) = \text{GPPF-TU}\big(\hat{P}_{ij}(k+\ell)\big)$
6: **end for**
7: **end for**

The filtering with respect to the DPGP target kinematics model can also be achieved by using GP-PF as sub-routines. In the measurement update step, the target-VF association probabilities also need to be modified according to the new measurement. To this end, adjustments for the target-VF association probability can be calculated by the intermediate weights in (4.19) or (4.20) depending on if the new measurement is empty, such that,

$$w_{ij} = \begin{cases} \dfrac{1}{c_1}\hat{w}_{ij} \displaystyle\sum_{\imath=1}^{n}\sum_{s=1}^{S} \gamma_{\imath}^{(s)}, & \mathbf{m}_j(k+1) \notin \varnothing \\ \dfrac{1}{c_2}\hat{w}_{ij} \displaystyle\sum_{\imath=1}^{n} \int_{\mathcal{W}\setminus\mathcal{S}(k)} f_G(\mathbf{x}_j; \hat{\boldsymbol{\eta}}_{\imath}(k+1), \hat{\boldsymbol{\Lambda}}_{\imath}(k+1)) d\mathbf{x}_j, & \mathbf{m}_j(k+1) \in \varnothing \end{cases} \tag{4.29}$$

for $i = 1, \ldots, M$, where c_1 and c_2 are the constants that make the resulting weights

normalized, such that $\sum_{i=1}^{M} w_{ij} = 1$, for $j = 1, \ldots, N$. In summary, the measurement update of the DPGP target kinematics model can be described by Algorithm 6.

Algorithm 6 DPGP Particle Filter Measurement Update (DPGP-PF-MU)

Input: Bank of predicted Gaussian mixture model parameters, $\{\hat{w}_{ij}, \hat{P}_{ij}(k+1)\}_{i=1}^{M}$, for $j = 1, \ldots, N$; Sensor measurement, $\mathbf{m}_j(k+1)$; Sensor FOV, $\mathcal{S}(k+1)$.

Output: Bank of updated Gaussian mixture model parameters, $\{w_{ij}, P_{ij}(k+1)\}_{i=1}^{M}$.

1: Update GMM parameters by Algorithm 2, $P_{ij}(k+1) = \text{GPPF-MU}\big(\hat{P}_{ij}(k+1)\big)$.

2: Update target-VF association weights, $\{w_{ij}\}_{i=1}^{M}$, by (4.29).

4.3 Chapter Conclusion

This chapter has proposed two flexible Bayesian nonparametric models that can describe target kinematics adaptive from sensor measurements: the GP target kinematics model and the DPGP target kinematics model. The proposed target kinematics models can be treated as the 'Target Model' block in Fig. 3.2 to the sensor planning problem (Problem 1). Efficient inference, prediction, and filtering algorithms are developed for the purpose of utilizing the proposed Bayesian nonparametric target kinematics models in the sensor planning problem. To be specific, a new GP particle filter is developed for the prediction and filtering with the GP target kinematics model, consisting of the GP particle filter-time update algorithm (Algorithm 1) and the GP particle filter-measurement update algorithm (Algorithm 2). When the target kinematics are described by the DPGP kinematics model, the novel DPGP particle filter developed in this chapter can be used to solve the prediction and filtering problem. The DPGP particle filter also consists two parts. The DPGP particle fitler-time update algorithm is presented as Algorithm 5 and the DPGP particle filter-measurement update algorithm is described as Algorithm 6.

5

Information Value for Nonparametric Target Models

Information theory addresses the quantities of the information in terms of the probability mass functions (PMFs) for discrete random variables or the probability density functions (PDFs) for continuous random variables [143]. Information theoretic functions are a natural choice for representing the information value because they measure the absolute or relative information content of PMFs or PDFs. In sensor planning problems, expected information values can be utilized to estimate the utilities of the future measurements before they are actually obtained. The expected information values can be treated as the rewards for the control inputs that lead to the acquisitions of the future measurements. Therefore, optimal sensor planning algorithms can be developed by maximizing the expected information theoretic functions. This chapter proposes novel information theoretic functions for the Bayesian nonparametric target kinematics model developed in Chapter 4, and can be treated as the 'Information Value' block in the diagram (Fig. 3.2) to the sensor planning algorithm (Problem 1). This chapter is organized as follows. The information theoretic functions for parametric models are first reviewed in Section 5.1. An approach to deriving

the expected information value functions for the Gaussian process is subsequently introduced in Section 5.2. Finally, the expected information value functions for the DPGP mixture model is presented in Section 5.3.

5.1 Information Theoretic Functions

Information theoretic functions, such as mutual information and information divergence, have been shown very useful in representing the information value in sensing planning and other information gathering problems [18, 144, 145]. One of the most widely applied information theoretic functions is the *differential entropy* (also known as continuous entropy). The differential entropy can be seen as an extension of the Shannon entropy for a continuous random variable, X, with PDF $p(x) : \mathcal{X} \rightarrow \mathbb{R}^+$, where $p(x) \triangleq p(X = x)$. The differential entropy is defined as,

$$H(X) = -\int_{\mathcal{X}} p(x) \log p(x) dx \tag{5.1}$$

The definition of the differential entropy can be extended to multiple variables. For a pair of random variables, $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ with a joint distribution $p(x, y) : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$, the *joint entropy*, $H(X, Y)$, is defined as,

$$H(X, Y) = -\int_{\mathcal{X}} \int_{\mathcal{Y}} p(x, y) \log p(x, y) dy dx \tag{5.2}$$

From the joint distribution of the random variables, $p(x, y)$, the *conditional entropy* can be calculated as,

$$H(Y|X) = -\int_{\mathcal{X}} \int_{\mathcal{Y}} p(x, y) \log p(y|x) dy dx \tag{5.3}$$

The chain rule also applies to the differential entropy as long as all the terms are finite, such that,

$$H(X, Y) = H(X) + H(Y|X) \tag{5.4}$$

As shown in [143], the *mutual information* (MI) is a measure of the information content of one random variable regarding another random variable. From the joint density function, $p(x, y)$, the mutual information between random variables X and Y is,

$$I(X;Y) = \int_{\mathcal{X}} \int_{\mathcal{Y}} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} dydx \tag{5.5}$$

It is worth noticing that $I(X;Y) \geqslant 0$ and the equality is achieved only if X and Y are independent. In addition, the mutual information can be calculated from differential entropies,

$$\begin{aligned} I(X;Y) &= -H(X,Y) + H(X) + H(Y) \\ &= H(X) - H(X|Y) = H(Y) - H(Y|X) \end{aligned} \tag{5.6}$$

Since $I(X;Y) \geqslant 0$, the 'information never hurts' principle can be derived,

$$H(X,Y) \leqslant H(X) + H(Y) \tag{5.7}$$

and

$$H(X|Y) \leqslant H(Y) \tag{5.8}$$

The equalities in both (5.7) and (5.8) are achieved when X and Y are independent.

The *KullbackLeibler (KL) information divergence*, also known as relative entropy, can be viewed as a measure of the difference between two probability density (or mass) functions. Let $p(x)$ and $q(x)$ denote two known probability density functions of a continuous random variable $X \in \mathbb{R}$. Then, the KL divergence,

$$D\big(p(x) \parallel q(x)\big) = \int_{-\infty}^{\infty} p(x) \ln \frac{p(x)}{q(x)} dx \tag{5.9}$$

also known as relative entropy, can be used to represent the "distance" between $p(x)$ and $q(x)$. Although it does not constitute a true distance metric because it is nonadditive, nonsymmetric, and does not obey the triangle inequality, the KL

divergence as been shown useful at representing the change in a PDF brought about, for example, by a new measurement or observation [146]. It is worth noticing that the KL divergence also has the property to be always non-negative,

$$D\big(p(x) \parallel q(x)\big) \geqslant 0 \tag{5.10}$$

with equality if and only if $p(x) = q(x)$ almost everywhere (a.e.). The KL divergence can also be related to the mutual information,

$$I(X;Y) = D\big(p(x,y) \parallel p(x)p(y)\big) = \mathbb{E}_y\big[D\big(p(x|y) \parallel p(x)\big)\big] \tag{5.11}$$

The information theoretic functions require the knowledge of the posterior distribution of random variables, which is related to the measurement value in the sensor planning problem. Therefore, they can not be applied directly. A general framework for using the information theoretic function in the sensor planning problem is proposed in [19, 146]. However, little work has been done in applying information theoretic functions for Bayesian nonparametric models. To this end, approaches to applying the theoretic functions with respect to the Gaussian process and the Dirichlet process-Gaussian process are presented in the subsequent sections.

5.2 Information Value for Gaussian Process

In the control literature, information value functions have been used to estimate the reward of the sensor control prior to obtaining the measurements and therefore be used to determine the sensor control. This section presents an information value function for the Gaussian processes based on the Kullback-Leibler (KL) divergence. While the approach can be used to derive other information theoretic functions, the KL divergence is chosen here because it was found to be effective for planning future measurements. Due to the adoption of the KL divergence the information value presented in this section is referred to as 'Gaussian process-Expected KL divergence' (GP-EKLD), denoted by $\hat{D}$.

Without loss of generality, it is assumed that the target kinematics are described by the ith VF $\mathbf{f}_i : \mathcal{W} \to \mathbb{R}$, defined in (3.3). In order to extend the concept of information theoretic functions, such as the KL divergence, to the Gaussian process that models $\mathbf{f}_i$, the VF is evaluated at a finite number of collocation points in the domain of integration $\mathcal{W}$ of the differential equations in (3.3). Let $\boldsymbol{\xi}_l \in \mathcal{W}$ denote the lth collocation point chosen from a uniform grid of L points in $\mathcal{W}$, as in basic collocation methods [147], and group all points on the grid in the $2L \times 1$ vector,

$$\boldsymbol{\xi} = [\boldsymbol{\xi}_1^T \quad \dots \quad \boldsymbol{\xi}_L^T]^T \tag{5.12}$$

Then, the target kinematics can be discretized about every collocation point on the grid by evaluating the velocity field $\mathbf{f}_i$ at $\boldsymbol{\xi}_l$ for $l = 1, \dots, L$, such that the $2L \times 1$ vector,

$$\boldsymbol{\upsilon}_i \triangleq [\mathbf{f}_i(\boldsymbol{\xi}_1)^T \quad \cdots \quad \mathbf{f}_i(\boldsymbol{\xi}_L)^T]^T = \boldsymbol{\upsilon}_i(\boldsymbol{\xi}) \tag{5.13}$$

can be used to approximate the velocity field $\mathbf{f}_i$ in $\mathcal{W}$. Then, the KL divergence for the Gaussian process can be calculated as,

$$D\big(\boldsymbol{\upsilon}_i; \mathbf{m}(k+1)\big) \triangleq D\Big(p\big(\boldsymbol{\upsilon}_i|\mathcal{M}(k+1)\big) \parallel p\big(\boldsymbol{\upsilon}_i|\mathcal{M}(k)\big)\Big) \tag{5.14}$$

where $\mathcal{M}(k+1) = \mathcal{M}(k) \bigcup \{\mathbf{m}_j(k+1)\}$.

Since $\mathbf{m}_j(k+1)$ is not available before it is actually obtained, the KL divergence defined in (5.14) can not be used in the sensor planning problem. One approach is to take expectation of (5.14) with respect to the distribution of $\mathbf{m}_j(k+1)$. Therefore, assumptions on the distribution of $\mathbf{m}_j(k+1)$ are required for the calculation of the expectation. It is assumed that the distribution of the future velocity measurement of the target, $\mathbf{z}_j(k+1)$, is consistent with the GP regression result obtained by all previous measurements. In addition, it is worth noticing that the position measurement $\mathbf{y}_j(k+1)$ is determined by the sensor planning algorithm. Therefore, $\mathbf{y}_j(k+1)$ can be treated as part of the control input to the sensor planning algorithm. In other

words, the optimal value of $\mathbf{y}_j(k+1)$ is determined by the sensor planning algorithm and no expectation over $\mathbf{y}_j(k+1)$ is needed. Then, the GP-EKLD for a sensing action can be defined as follows:

$$\hat{D}\big(\boldsymbol{\upsilon}_i; \mathbf{m}(k+1)\big) = \int_{\mathbb{R}} D\Big(p\big(\boldsymbol{\upsilon}_i|\mathcal{M}(k+1)\big) \parallel p\big(\boldsymbol{\upsilon}_i|\mathcal{M}(k)\big)\Big) \times p\big(\mathbf{z}_j(k+1)|\mathcal{M}(k), \mathbf{y}_j(k+1)\big) d\mathbf{z}_j(k+1) \tag{5.15}$$

From the Gaussian process regression, it can be shown that the marginal distribution of $\boldsymbol{\upsilon}_i$ is a multivariate Gaussian distribution with the measurement mean vector, denoted by $\boldsymbol{\mu}_i(k)$, and measurement covariance matrix, denoted by $\boldsymbol{\Sigma}_i(k)$, calculated from the measurements in $\mathcal{M}(k)$. Recall that $\mathbf{Y}_i(k) = [\mathbf{y}_1^T(\cdot) \quad \mathbf{y}_2^T(\cdot) \cdots]^T$ and $\mathbf{Z}_i(k) = [\mathbf{z}_1^T(\cdot) \quad \mathbf{z}_2^T(\cdot) \cdots]^T$ denote two vectors containing all noisy measurements of target position and velocity, respectively, that have been associated with the ith velocity field up to time k. Then, the measurement mean vector is,

$$\boldsymbol{\mu}_i(k) = \boldsymbol{\Phi}\left[\boldsymbol{\xi}, \mathbf{Y}_i(k)\right] \left\{\boldsymbol{\Phi}[\mathbf{Y}_i(k), \mathbf{Y}_i(k)] + \sigma_v^2 \mathbf{I}_{2k}\right\}^{-1} \mathbf{Z}_i(k) \tag{5.16}$$

and the measurement covariance matrix is,

$$\boldsymbol{\Sigma}_i(k) = \boldsymbol{\Phi}(\boldsymbol{\xi}, \boldsymbol{\xi}) - \boldsymbol{\Phi}[\boldsymbol{\xi}, \mathbf{Y}_i(k)] \left\{\boldsymbol{\Phi}[\mathbf{Y}_i(k), \mathbf{Y}_i(k)] + \sigma_v^2 \mathbf{I}_{2k}\right\}^{-1} \boldsymbol{\Phi}[\mathbf{Y}_i(k), \boldsymbol{\xi}] \tag{5.17}$$

where $\boldsymbol{\xi}$ is the vector of chosen collocation points in the target workspace $\mathcal{W}$ defined in (5.12).

Since the prior and posterior distributions of $\boldsymbol{\upsilon}_i$ are both multivariate Gaussian distributions, the computation of the GP-EKLD can be simplified according to the following theorem:

Theorem 4. *Consider a Gaussian process GP_i with known covariance matrix function $\boldsymbol{\phi}$. The GP-EKLD defined in* (5.15) *affords the analytical solution,*

$$\hat{D}\big(\boldsymbol{\upsilon}_i; \mathbf{m}(k+1)\big) = \frac{1}{2}\left[\mathrm{tr}\left(\boldsymbol{\Sigma}_{i,k}^{-1}\boldsymbol{\Sigma}_{i,k+1} + \mathbf{Q}^{-1}\mathbf{R}^T\boldsymbol{\Sigma}_{i,k}^{-1}\mathbf{R}\mathbf{Q}^{-1}\sigma_v^2\right) - \ln\left(\frac{|\boldsymbol{\Sigma}_{i,k+1}|}{|\boldsymbol{\Sigma}_{i,k}|}\right) - 2L\right]$$

where $\mathbf{\Sigma}_{i,k} \triangleq \mathbf{\Sigma}_i(k)$, $\mathbf{\Sigma}_{i,k+1} \triangleq \mathbf{\Sigma}_i(k+1)$, *are the velocity covariance matrices at times* k *and* $k+1$, *respectively, and,*

$$\mathbf{\Sigma} \triangleq \mathbf{\Phi}[\mathbf{Y}_i(k), \mathbf{Y}_i(k)] + \sigma_v^2 \mathbf{I} \tag{5.18}$$

$$\mathbf{R} \triangleq \mathbf{\Phi}[\boldsymbol{\xi}, \mathbf{y}_j(k+1)] - \mathbf{\Phi}[\boldsymbol{\xi}, \mathbf{Y}_i(k)]\mathbf{\Sigma}^{-1}\mathbf{\Phi}[\mathbf{Y}_i(k), \mathbf{y}_j(k+1)] \tag{5.19}$$

$$\begin{aligned}\mathbf{Q} \triangleq\ & \mathbf{\Phi}[\mathbf{y}_j(k+1), \mathbf{y}_j(k+1)] + \sigma_v^2 \\ & - \mathbf{\Phi}[\mathbf{y}_j(k+1), \mathbf{Y}_i(k)]\mathbf{\Sigma}^{-1}\mathbf{\Phi}[\mathbf{Y}_i(k), \mathbf{y}_j(k+1)]\end{aligned} \tag{5.20}$$

for any $\mathbf{y}_j(k+1) \in \mathcal{S}(k+1)$, *where* $\boldsymbol{\xi}$ *is a vector of collocation points,* $\mathbf{Y}_i$ *is a vector of all past position measurements, and* $\mathbf{\Phi}$ *is the cross-covariance matrix.*

The proof is shown in Appendix A.2

5.3 Information Value for Dirichlet Process-Gaussian Process

Although the GP is effective at regression and posterior distribution prediction, one GP is often not enough for modeling multiple targets, since it is often the case that different groups of targets may display different behaviors. Therefore, the Dirichlet process is proposed to represent the prior knowledge of the distribution of GPs, resulting in a Dirichlet process-Gaussian process mixture model (DPGP-MM). Because DPGP-MMs can be viewed as distributions over probability distributions [148, 54], traditional information theoretic functions are not directly applicable. Therefore, this section presents a new information theoretic function that represents the information value of future measurements in closed form and, thus, can be optimized with respect to sensor planning and decision algorithms. The information value is developed as an extension to the GP-EKLD presented in Section 5.2. In particular, this section derives the DPGP KL divergence between measurements (Section 5.3.1), and then obtains a compact analytical form of DPGP expected KL divergence (Section 5.3.3), under the assumption that position measurement errors are negligible. Finally, in Section 5.3.4, an approximation of the DPGP-EKLD is obtained via Monte

Carlo integration, and the variance of the approximation error is shown to decrease linearly with the inverse of the number of samples.

5.3.1 DPGP KL-Divergence

In order to calculate the KL divergence for the DPGP mixture model in (2.25), all of the M velocity fields in $\mathcal{F}$ defined in (3.3) are evaluated at a finite number of collocation points in the domain of integration $\mathcal{W}$ of the differential equations in (3.3). Recall from (5.13) that $\boldsymbol{v}_i$ is used to represent the ith velocity field, for $i = 1, \ldots, M$. Similarly, all M velocity fields in $\mathcal{F}$ can be discretized and grouped into a $2LM \times 1$ vector of random velocity variables associated with the collocation points in $\boldsymbol{\xi}$ defined as

$$\boldsymbol{v} \triangleq [\boldsymbol{v}_1(\boldsymbol{\xi})^T \quad \cdots \quad \boldsymbol{v}_M(\boldsymbol{\xi})^T]^T = \boldsymbol{v}(\boldsymbol{\xi}) \tag{5.21}$$

Now, suppose $p(\boldsymbol{v})$ and $q(\boldsymbol{v})$ denote two joint PDFs of the $2LM$ elements of the random velocity vector $\boldsymbol{v}$ [149], where each PDF is obtained from the DPGP-MM of target dynamics in (2.25). Then, the "distance" between the two parameterizations of DPGP-MMs by collocation points can be represented by the DPGP KL divergence defined as,

$$D\big(p(\boldsymbol{v}) \parallel q(\boldsymbol{v})\big) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} p(\boldsymbol{v}) \ln \frac{p(\boldsymbol{v})}{q(\boldsymbol{v})} \, d\boldsymbol{v} \tag{5.22}$$

In order to evaluate (5.22), the joint PDF of $p(\boldsymbol{v})$ needs to be expressed in terms of the DPGP parameters in (2.25). Given $\{\boldsymbol{\pi}, \boldsymbol{\theta}_i, \boldsymbol{\Phi}\}$, the random vector $\boldsymbol{v}_i$ has a mixed multivariate Gaussian distribution with mean and covariance matrix calculated from $\boldsymbol{\theta}_i$ and $\boldsymbol{\Phi}$. However, the number of components in the mixture model is infinite. Therefore, computing the DPGP KL divergence from the above definition without any assumption can be computationally very expensive due to the multiple integrals of the joint PDFs in (5.22). The next subsections present several steps by which the KL divergence can be simplified and, then, approximated by an information function

that is computationally efficient, using conditional independence assumptions and Monte Carlo integration.

5.3.2 DPGP Conditional KL-Divergence

When using information theoretic functions to plan sensor measurements over time, the conditional probability densities must be taken into account because at any moment in time, t, a set of measurements is typically already available or *given*. Consider a DPGP-MM that is updated incrementally over time using a sensor that requires a small but finite constant time interval, Δt, to obtain and process new target measurements, subject to a finite sampling rate [134]. At every discrete time instant indexed by k, let $\mathbf{m}_j(k)$ denote a sample of noisy measurements, in the form (3.7), obtained from the jth target during the kth sampling time interval, $[t_k,\ t_k + \Delta t)$. It typically can be assumed that the sensor field-of-view is stationary during sampling and, therefore, measurements can be obtained from target j, provided $\mathbf{x}_j(t) \in \mathcal{S}(t)$ for some $t \in [t_k,\ t_k + \Delta t)$. Then, during the kth sampling time interval, the jth target position can be approximated by $\mathbf{x}_j(k)$, and the sensor FOV can be represented by $\mathcal{S}(k)$ [134].

For a non-myopic process, the information value of an additional measurement $\mathbf{m}_j(k+1)$ is to be conditioned on all prior measurements obtained from all the targets, $\mathcal{M}(k)$, because the DPGP cluster learning process is dependent on all N targets observed up to time k. By assuming that the DP prior is fixed during the planning process, the VF clusters can be assumed to remain unchanged. Then, the VF-target associations learned by the DPGP-MM at time step k, denoted by the set $\mathcal{G}(k) \triangleq \{G_j(k) \mid 1 \leqslant j \leqslant N\}$, can be considered part of the database used to learn the target dynamics, denoted by $Q(k) \triangleq \{\mathcal{M}(k), \mathcal{G}(k)\}$.

Under these assumptions, the change in the DPGP-MM brought about by a measurement $\mathbf{m}_j(k+1)$ can be represented by the conditional KL divergence (or

relative entropy),

$$D\big(\boldsymbol{v};\mathbf{m}_j(k+1)|Q(k)\big) \triangleq D\left(p\big(\boldsymbol{v}|\mathbf{m}_j(k+1),Q(k)\big) \parallel p\big(\boldsymbol{v}|Q(k)\big)\right) \tag{5.23}$$

in terms of the joint PDFs of the VF vector $\boldsymbol{v}$ obtained from the DPGP-MM. Because the VF vectors of two targets $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$, with $i \neq j$, are conditionally independent given the measurement database $Q(k)$, the joint PDF can be factorized as follows,

$$p(\boldsymbol{v}|Q(k)) = p(\boldsymbol{v}_1,\ldots,\boldsymbol{v}_M|Q(k)) = \prod_{i=1}^{M} p(\boldsymbol{v}_i|Q(k)) \tag{5.24}$$

where, each joint PDF $p(\boldsymbol{v}_i|Q(k))$ can be obtained from the GP covariance and mean as follows.

From GP regression [43], given the data $Q(k)$, $\boldsymbol{v}_i$ is characterized by the multivariate joint Gaussian PDF,

$$p(\boldsymbol{v}_i|Q(k)) = f_G\big(\boldsymbol{v}_i;\boldsymbol{\mu}_i(k),\boldsymbol{\Sigma}_i(k)\big) \tag{5.25}$$

where $\boldsymbol{\mu}_i(k)$ and $\boldsymbol{\Sigma}_i(k)$ are the measurement mean vector and covariance matrix of the measurements associated with the ith velocity field, defined in (5.16) and (5.16), respectively.

Suppose, without loss of generality, that a VF-target association is given as $G_j(k+1) = i$. Then, since measurements obtained from one velocity field do not contain information about other velocity fields, it follows that,

$$D\big(\boldsymbol{v};\mathbf{m}_j(k+1)|Q(k)\big) = D\big(\boldsymbol{v}_i;\mathbf{m}_j(k+1)|Q(k)\big) \tag{5.26}$$

for $G_j(k+1) = i$, as shown by the proof in Appendix A.3. Then, because when the measurement $\mathbf{m}_j(k+1)$ and the VF-target association $G_j(k+1)$ are both given, $\boldsymbol{v}_i$ is characterized by the multivariate joint Gaussian PDF in (5.25), the DPGP KL

divergence in (5.23) can be obtained in closed form as follows,

$$D\big(\boldsymbol{v};\mathbf{m}_j(k+1)|Q(k)\big) = \frac{1}{2}\left[\mathrm{tr}\left(\boldsymbol{\Sigma}_{i,k}^{-1}\boldsymbol{\Sigma}_{i,k+1}\right) - \ln\left(\left|\boldsymbol{\Sigma}_{i,k+1}\boldsymbol{\Sigma}_{i,k}^{-1}\right|\right) - 2L\right] + \frac{1}{2}\left(\boldsymbol{\mu}_{i,k+1}-\boldsymbol{\mu}_{i,k}\right)^T\boldsymbol{\Sigma}_{i,k}^{-1}\left(\boldsymbol{\mu}_{i,k+1}-\boldsymbol{\mu}_{i,k}\right) \tag{5.27}$$

where $\mathrm{tr}(\cdot)$ denotes the trace of a matrix, L is the number of collocation points, and $\boldsymbol{\mu}_{i,k} = \boldsymbol{\mu}_i(k)$ and $\boldsymbol{\Sigma}_{i,k} = \boldsymbol{\Sigma}_i(k)$ are abbreviations of the mean and covariance matrix defined in (5.16)-(5.17), respectively.

Because in active sensing and information gathering problems, the values of future measurements, $\mathbf{m}_j(k+1)$, are unavailable, the next section presents an approach for estimating the DPGP KL divergence from an existing DPGP-MM and past sensor measurements.

5.3.3 DPGP Expected KL-Divergence

Consider now the problem in which the DPGP KL divergence function derived in the previous section is to be used to determine the information value of a future measurement vector, $\mathbf{m}_j(k+1)$, such that, the sensor can be managed so as to minimize the uncertainty associated with the next DPGP-MM learned from data. When $\mathbf{m}_j(k+1)$ is unknown, the VF-target association $G_j(k+1)$ cannot be learned from data. Thus, the closed analytic form of the conditional KL divergence in (5.27) cannot be evaluated because the measurement mean vector and covariance matrix are not known at time $k+1$. An estimate of this information theoretic function, however, can be obtained by taking the expectation of the conditional KL divergence in (5.23) with respect to the unknown random variables $\mathbf{m}_j(k+1)$ and $G_j(k+1)$ as follows,

$$\hat{D}\left(\boldsymbol{v};\mathbf{m}_j(k+1)|Q(k)\right) = \mathbb{E}_{\mathbf{m}_j(k+1)}\left\{\mathbb{E}_{G_j(k+1)}\left\{D\left(\boldsymbol{v};\mathbf{m}_j(k+1)|Q(k)\right)\right\}\right\} \tag{5.28}$$

From the conditional independence property in (5.26) and the linearity of expectation operator, the above DPGP expected KL divergence (EKLD) can be written

as,

$$
\begin{aligned}
&\hat{D}\left(\boldsymbol{v};\mathbf{m}_j(k+1)|Q(k)\right)\\
&=\mathbb{E}_{\mathbf{m}_j(k+1)}\left\{\mathbb{E}_{G_j(k+1)}\left\{D\left(\boldsymbol{v}_i;\mathbf{m}_j(k+1)\middle|Q(k),G_j(k+1)=i\right)\right\}\right\}\\
&=\mathbb{E}_{\mathbf{m}_j(k+1)}\left[\sum_{i=1}^{M}D\left(\boldsymbol{v}_i;\mathbf{m}_j(k+1)\middle|Q(k),G_j(k+1)=i\right)p\left(G_j(k+1)=i\middle|Q(k)\right)\right]\\
&=\sum_{i=1}^{M}\mathbb{E}_{\mathbf{m}_j(k+1)}\left[D\left(\boldsymbol{v}_i;\mathbf{m}_j(k+1)|Q(k)\right)\right]\cdot p\left(G_j(k+1)=i|Q(k)\right)\\
&=\sum_{i=1}^{M}\hat{D}\left(\boldsymbol{v}_i;\mathbf{m}_j(k+1)|Q(k)\right)\cdot p\left(G_j(k+1)=i|Q(k)\right)
\end{aligned}
\tag{5.29}
$$

Let $\mathcal{M}_j^c = Q\backslash\mathcal{M}_j$ denote the complement set of $\mathcal{M}_j$ in Q. Then, the posterior probability of event $\{G_j(k+1)=i\}$ can be obtained from Bayes' rule,

$$
p\left(G_j(k+1)=i|Q(k)\right)=\frac{\pi_i\cdot p(\mathcal{M}_j(k)|\mathcal{M}_j^c(k),G_j(k+1)=i)}{\sum_{i=1}^{M}\pi_i\cdot p(\mathcal{M}_j(k)|\mathcal{M}_j^c(k),G_j(k+1)=i)}\triangleq w_{ij} \tag{5.30}
$$

where $\pi_i = p(G_j(k+1)=i)$, defined in (3.2), is the prior probability that the jth target follows the ith VF. The above posterior probability is obtained with respect to the measurement set $\mathcal{M}_j(k)$, defined in (3.8), because every target follows a VF and, thus, all prior measurements influence the posterior of $G_j(k+1)$.

Recall that the likelihood, $p\left(\mathcal{M}_j(k)|\mathcal{M}_j^c(k),G_j(k+1)=i\right)$, is calculated by (4.23). Then, the expected KL divergence of the ith VF in (5.29) is obtained by marginalizing the original KL divergence function over all possible values of $\mathbf{m}_j(k+1)$ as follows,

$$
\begin{aligned}
&\hat{D}\left(\boldsymbol{v}_i;\mathbf{m}_j(k+1)\right)\\
&=\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}D\left(\boldsymbol{v}_i;\mathbf{m}_j(k+1)\right)\cdot p\left(\mathbf{m}_j(k+1)|G_j(k+1)=i\right)d\mathbf{m}_j(k+1)
\end{aligned}
\tag{5.31}
$$

where $D(\cdot)$ is defined as in (5.27). For brevity, $Q(k)$ is omitted above and in the remainder of the dissertation, but all information functions and probabilities are assumed to be conditioned on all past data, $Q(k)$.

The measurement probability distribution in (5.31) can be derived from the measurement model (3.7) by marginalizing over the estimates of target positions in the sensor field-of-view $\mathcal{S}$ at time $k+1$,

$$p\left(\mathbf{m}_j(k+1)|G_j(k+1)=i\right) = \tag{5.32}$$
$$p(\mathbf{z}_j(k+1)|\mathbf{x}_j(k+1))p\left(\mathbf{x}_j(k+1)|G_j(k+1)=i\right)$$

where $p(\mathbf{z}_j(k+1)|\mathbf{x}_j(k+1))$ can be calculated from (3.7). Using Euler integration and the ODE (3.3), it follows that,

$$p\left(\mathbf{x}_j(k+1)|G_j(k+1)=i\right) = \tag{5.33}$$
$$\int_{\mathbb{R}^2} p\left(\mathbf{v}_j(k)|\mathbf{x}_j(k)\right) f_X\left(\mathbf{x}_j(k+1)-\mathbf{v}_j(k)\Delta t\right) d\mathbf{v}_j(k)$$

where $f_X(\cdot)$ represents the PDF of $\mathbf{x}_j(k)$, and is computed via filtering techniques [102]. The conditional joint probability distribution for the target speed, denoted by $p\left(\mathbf{v}_j(k)|\mathbf{x}_j(k)\right)$ in (5.33), is a multivariate Gaussian distribution with mean and covariance calculated according to (5.16)-(5.17) by replacing the collocation point vector $\boldsymbol{\xi}$ with the target position $\mathbf{x}_j(k)$.

In summary, the information value of a future measurement $\mathbf{m}_j(k+1)$ can be estimated using the DPGP EKLD function in (5.29), and equations (5.30)-(5.33). When the sensor can obtain multiple target measurements during the same sampling time interval $[t_k,\ t_k+\Delta t)$, the information value can be represented by the total expected KL divergence of all measurements that may be obtained subject to the sensor field-of-view $\mathcal{S}(k+1)$. Let the vector $\mathbf{m}(k) = [\mathbf{m}_j^T(k) \quad \mathbf{m}_l^T(k)\cdots]^T$ denote all measurements obtained from targets j, l, ..., in $\mathcal{S}(k)$ during the time interval $[t_k,\ t_k+\Delta t)$ (Section 5.3.2). Then, the DPGP-EKLD of a measurement vector $\mathbf{m}(k+1)$ that may be obtained subject to $\mathcal{S}(k+1)$ during a future time interval

$[t_{k+1},\ t_{k+1}+\Delta t)$ is given by,

$$\hat{D}\left(\boldsymbol{v};\mathbf{m}(k+1)\right)=\sum_{j,\ \mathbf{x}_j\in\mathcal{S}(k+1)}\hat{D}\left(\boldsymbol{v};\mathbf{m}_j(k+1)\right)= \tag{5.34}$$

$$\sum_{j,i}w_{ij}\int_{\mathcal{S}(k+1)}\int_{\mathbb{R}^2}D\left(\boldsymbol{v}_i;\mathbf{m}_j(k+1)\right)p\left(\mathbf{z}_j(k+1)|\mathbf{x}_j(k+1)\right)$$

$$\times\, p\left(\mathbf{x}_j(k+1)|G_j(k+1)=i\right)d\mathbf{z}_j(k+1)d\mathbf{x}_j(k+1)$$

where all quantities are defined and calculated as shown in (5.30)-(5.33).

Because the total DPGP-EKLD function in (5.34) involves an 6th-order integral, its computation is typically very burdensome and may become prohibitive particularly when the information value of all possible sensor decisions needs to be computed repeatedly over time. Using the methodology presented in the next subsection, it is possible to obtain a DPGP-EKLD approximation that reduces the integration required to a double integral and, thus, can be efficiently computed and implemented in real time for sensor planning.

5.3.4 Approximation to DPGP Expected KL-Divergence

By assuming that the noise in position measurements is negligible compared to the noise in velocity measurements, an approximate KL divergence function can be derived analytically with respect to a double integral of the velocity measurements. Subsequently, through the use of three lemmas introduced in this subsection, it is shown that the Monte Carlo integration of the remaining double integral provides an unbiased estimator of the approximate KL divergence with an error variance that decreases linearly with the number of samples in $\mathcal{W}$. The computation required by the resulting DPGP-EKLD approximation is far reduced compared to the original DPGP-EKLD function in (5.34) and, thus, can be implemented to plan the measurements of an active sensor in real time, as shown by the camera-planning application presented in chapter 7.

Since the probability $p\left(\mathbf{x}_j(k+1)|G_j(k+1)=i\right)$ is constant with respect to the measurements $\mathbf{m}_j(k+1)$, the expected KL divergence of a measurement vector $\mathbf{m}(k+1)$ in (5.34) can be written as,

$$\begin{aligned}&\hat{D}\left(\boldsymbol{\upsilon};\mathbf{m}(k+1)\right) \\ &= \sum_j \sum_i w_{ij} \int\limits_{\mathcal{S}(k+1)} \Big\{ h_i[\mathbf{x}_j(k+1)] \cdot p\left(\mathbf{x}_j(k+1)|G_j(k+1)=i\right) \Big\} d\mathbf{x}_j(k+1)\end{aligned} \tag{5.35}$$

where,

$$\begin{aligned}h_i[\mathbf{x}_j(k+1)] &\triangleq \int_{\mathbb{R}^2} D\big(\boldsymbol{\upsilon}_i;\mathbf{m}_j(k+1)\big) p\big(\mathbf{z}_j(k+1)|\mathbf{x}_j(k+1)\big) d\mathbf{z}_j(k+1) \\ &= \hat{D}\big(\boldsymbol{\upsilon}_i;\mathbf{m}_j(k+1)\big)\end{aligned} \tag{5.36}$$

From (5.36), it can be seen that the inner integral, $h_i[\mathbf{x}_j(k+1)]$, is equivalent to the GP-EKLD, $\hat{D}\big(\boldsymbol{\upsilon}_i;\mathbf{m}_j(k+1)\big)$, defined in (5.15). Therefore, $h_i[\mathbf{x}_j(k+1)]$ can be calculated analytically using Theorem 4, such that,

$$h_i[\mathbf{x}_j(k+1)] = \frac{1}{2}\left[\mathrm{tr}\left(\boldsymbol{\Sigma}_{i,k}^{-1}\boldsymbol{\Sigma}_{i,k+1}\right) - \ln\left(\frac{|\boldsymbol{\Sigma}_{i,k+1}|}{|\boldsymbol{\Sigma}_{i,k}|}\right) - 2L + \mathrm{tr}(\mathbf{Q}^{-1}\mathbf{R}^T\boldsymbol{\Sigma}_{i,k}^{-1}\mathbf{R}\mathbf{Q}^{-1})\sigma_v^2\right]$$

Even with the above simplification, the computational complexity associated with evaluating $h_i[\mathbf{x}_j(k+1)]$ is $O(L^3+k^3)$, where L is the number of collocation points and k is the time index. Therefore, it is often infeasible to compute the approximate DPGP-EKLD in (5.35) for all possible sensor fields-of-view, $\mathcal{S}(k+1)$, in $\mathcal{W}$. Because the integrand of (5.35) goes to zero when the probability $p\left(\mathbf{x}_j(k+1)|G_j(k+1)=i\right)$ goes to zero and $h_i[\mathbf{x}_j(k+1)]$ is finite, the computation of (5.35) can be significantly reduced by the approach known as Monte Carlo integration [150]. Let $\boldsymbol{\chi}^{(1)},\ldots,\boldsymbol{\chi}^{(S)}$ denote S values of $\mathbf{x}_j(k+1)$ drawn identically and independently from the target state distribution $p\left(\mathbf{x}_j(k+1)|G_j(k+1)=i\right)$ in (5.33). The samples can be obtained from the DPGP particle filter described by Algorithm 5 and Algorithm 6 in Chapter

4.2.2. Then, the integral in (5.35) can be computed numerically by evaluating its integrand at each sample with nonzero probability, such that,

$$\hat{D}\left(\boldsymbol{v};\mathbf{m}(k+1)\right) \approx \sum_{i=1}^{M}\sum_{j=1}^{N}\frac{w_{ij}}{S}\sum_{s=1}^{S} h_i(\boldsymbol{\chi}^{(s)})\mathbf{1}_{\mathcal{S}(k+1)}(\boldsymbol{\chi}^{(s)}) \tag{5.37}$$

where $\boldsymbol{\chi}^{(s)}$, $s = 1,\ldots,S$, denote S target-position samples drawn independently and identically from the target state distribution (5.33), and the indicator function is defined as:

$$\mathbf{1}_{\mathcal{S}(k+1)}(\boldsymbol{\chi}^{(s)}) \triangleq \begin{cases} 1, & \boldsymbol{\chi}^{(s)} \in \mathcal{S}(k+1) \\ 0, & \boldsymbol{\chi}^{(s)} \notin \mathcal{S}(k+1) \end{cases} \tag{5.38}$$

Similarly to the collocation points used to discretize the DPGP information value (Section 5.3.1), these samples represent points in $\mathcal{W}$. But, unlike collocation points, which are chosen from a uniform grid, the Monte Carlo integration samples are chosen by sampling a known distribution.

The remainder of this subsection shows that, with the help of the three following lemmas, the approximation in (5.37) is proven to be an unbiased estimator of the approximate DPGP-EKLD in (5.35), and that the error variance for this approximation decreases linearly with S. The first lemma provides a bound for the covariance matrix of the velocity at the predicted target position, $\mathbf{Q}$, defined in (5.20). This result is used in a second lemma to derive a matrix inequality between consecutive covariance matrices $\boldsymbol{\Sigma}_{i,k}$ and $\boldsymbol{\Sigma}_{i,k+1}$, defined in (5.17), such that $\mathrm{tr}(\boldsymbol{\Sigma}_{i,k}^{-1}\boldsymbol{\Sigma}_{i,k+1})$ and $\ln(|\boldsymbol{\Sigma}_{i,k+1}\boldsymbol{\Sigma}_{i,k}^{-1}|)$ in $h_i[\mathbf{x}_j(k+1)]$ in (5.35) can be shown to be bounded. The third and last lemma present a bound on the trace of the matrix $\mathbf{R}^T\mathbf{R}$, such that $\mathrm{tr}(\mathbf{Q}^{-1}\mathbf{R}^T\boldsymbol{\Sigma}_{i,k}^{-1}\mathbf{R}\mathbf{Q}^{-1})\sigma_v^2$ in $h_i[\mathbf{x}_j(k+1)]$ can be shown finite-valued. Let $\mathbf{A} \preceq \mathbf{B}$ denote the element-wise inequality between matrices $\mathbf{A}$ and $\mathbf{B}$. Then, the following lemma provides a bound on the elements of $\mathbf{Q}$ that is later used to establish the matrix inequality between $\boldsymbol{\Sigma}_{i,k}$ and $\boldsymbol{\Sigma}_{i,k+1}$.

Lemma 5. *The velocity covariance matrix* $\mathbf{Q}$*, defined in* (5.20)*, obeys the element-wise inequality* $\boldsymbol{\phi}_0 + \{\sigma_v^2/[k\ \mathrm{tr}(\boldsymbol{\phi}_0) + \sigma_v^2]\}\,\mathbf{I}_2 \preceq \mathbf{Q} \preceq \boldsymbol{\phi}_0 + \sigma_v^2\mathbf{I}_2$*, where* $\boldsymbol{\phi}_0 = \boldsymbol{\phi}(\mathbf{0},\mathbf{0})$ *is a constant matrix obtained by evaluating the stationary covariance matrix function at zero,* $\sigma_v \in \mathbb{R}$ *is the standard deviation of the measurement noise of velocity, and* $k \in \mathbb{Z}_+$ *is the time index.*

Proof. From definition (5.18), the matrix $\boldsymbol{\Sigma}$ is symmetric and positive definite ($\boldsymbol{\Sigma} > 0$) because it is the sum of a symmetric, positive semi-definite matrix, $\boldsymbol{\Phi}[\mathbf{Y}_i(k), \mathbf{Y}_i(k)]$, and a symmetric, positive definite matrix, $\sigma_v^2\mathbf{I}_{2k}$. Then, the inverse $\boldsymbol{\Sigma}^{-1}$ also is positive definite and $\mathbf{C}^T\boldsymbol{\Sigma}^{-1}\mathbf{C} \succeq \mathbf{0}_{2\times 2}$, for all $\mathbf{C} \in \mathbb{R}^{k\times 2}$. Thus, from (5.20) the following element-wise inequality holds,

$$\mathbf{Q} \preceq \boldsymbol{\phi}[\mathbf{x}_j(k+1), \mathbf{x}_j(k+1)] + \sigma_v^2\mathbf{I}_2 \tag{5.39}$$

Since the covariance matrix function is stationary, $\boldsymbol{\phi}[\mathbf{x}_j(k+1), \mathbf{x}_j(k+1)] = \boldsymbol{\phi}(\mathbf{0},\mathbf{0}) = \boldsymbol{\phi}_0$ for any $\mathbf{x}_j \in \mathcal{W}$, and thus,

$$\mathbf{Q} \preceq \boldsymbol{\phi}_0 + \sigma_v^2\mathbf{I}_2 \tag{5.40}$$

Since $\boldsymbol{\Phi}[\mathbf{Y}_i(k), \mathbf{Y}_i(k)]$ is real, symmetric, and positive semi-definite, there exists an eigenvalue decomposition, $\boldsymbol{\Phi}[\mathbf{Y}_i(k), \mathbf{Y}_i(k)] = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{-1}$, with orthogonal eigenvectors, where $\boldsymbol{\Lambda}$ is a diagonal matrix obtained by placing the k eigenvalues of $\boldsymbol{\Phi}[\cdot]$ on the diagonal, and $\mathbf{U}$ is a $k \times k$ matrix whose columns are the eigenvectors of $\boldsymbol{\Phi}[\cdot]$, i.e.,

$$\boldsymbol{\Lambda} \triangleq \mathrm{diag}[\lambda_1 \quad \cdots \quad \lambda_k] \quad \text{and} \quad \mathbf{U} \triangleq [\mathbf{u}_1 \quad \cdots \quad \mathbf{u}_k]^T. \tag{5.41}$$

and, thus, $\mathbf{U}^T\mathbf{U} = \mathbf{I}$. Since the kth column of $\mathbf{Y}_i(k)$ is equal to $\mathbf{x}_j(k)$, the matrix $\mathbf{C}$ defined in (5.20) can be written as,

$$\mathbf{C} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{u}_k \tag{5.42}$$

and, by substituting (5.42) into in (5.20), the matrix $\mathbf{Q}$ can be written as,

$$\mathbf{Q} = \boldsymbol{\phi}_0 + \left[\sigma_v^2 - \sum_{\ell=1}^{k} \frac{\lambda_\ell}{\lambda_\ell + \sigma_v^2}\lambda_\ell(\mathbf{U}_{(\ell,k)})^2\right]\mathbf{I}_2 \tag{5.43}$$

where λ_ℓ is the ℓth eigenvalue of $\mathbf{\Phi}[\mathbf{Y}_i(k), \mathbf{Y}_i(k)]$, and $\mathbf{U}_{(\ell,k)}$ denotes the element in the ℓth row and kth column of $\mathbf{U}$. Because $\mathbf{\Phi}[\cdot]$ is symmetric and positive semi-definite $\lambda_\ell \geqslant 0$ for all ℓ, and, since $\mathbf{u}_k^T \mathbf{\Lambda} \mathbf{u}_k = 1$, it follows that

$$\sum_{\ell=1}^{k} \lambda_\ell (\mathbf{U}_{(\ell,k)})^2 = 1 \tag{5.44}$$

Substituting (5.44) into (5.43), it follows that,

$$\mathbf{Q} \succeq \boldsymbol{\phi}_0 + \left(\sigma_v^2 - \max_\ell \left\{ \frac{\lambda_\ell}{\lambda_\ell + \sigma_v^2} \right\} \right) \mathbf{I}_2 = \boldsymbol{\phi}_0 + \left(\sigma_v^2 - \frac{\max_\ell \{\lambda_\ell\}}{\max_\ell \{\lambda_\ell\} + \sigma_v^2} \right) \mathbf{I}_2 \tag{5.45}$$

providing a lower bound on the elements of $\mathbf{Q}$ that can be simplified as follows. Because the trace of a real, symmetric matrix equals the sum of its eigenvalues,

$$\operatorname{tr} \{\mathbf{\Phi}[\mathbf{Y}_i(k), \mathbf{Y}_i(k)]\} = \sum_{\ell=1}^{k} \lambda_\ell \tag{5.46}$$

and since the diagonal blocks of $\mathbf{\Phi}[\mathbf{Y}_i(k), \mathbf{Y}_i(k)]$ are equal to $\boldsymbol{\phi}(\mathbf{y}_l, \mathbf{y}_l) = \boldsymbol{\phi}_0$, then $\sum_{\ell=1}^{k} \lambda_\ell = k \operatorname{tr}(\boldsymbol{\phi}_0)$. Furthermore, $\max_\ell \{\lambda_\ell\} \leqslant k \operatorname{tr}(\boldsymbol{\phi}_0)$, and thus,

$$\mathbf{Q} \succeq \boldsymbol{\phi}_0 + \left[\frac{\sigma_v^2}{k \operatorname{tr}(\boldsymbol{\phi}_0) + \sigma_v^2} \right] \mathbf{I}_2 \succeq \boldsymbol{\phi}_0 \succ \mathbf{0} \tag{5.47}$$

which completes the proof. □

The above result is used in the following lemma to establish a matrix inequality on consecutive covariance matrices $\mathbf{\Sigma}_{i,k}$ and $\mathbf{\Sigma}_{i,k+1}$.

Lemma 6. *Under the assumptions in Theorem* 4*, two consecutive covariance matrices* $\mathbf{\Sigma}_{i,k}$ *and* $\mathbf{\Sigma}_{i,k+1}$*, defined according to* (5.17)*, obey the element-wise inequality* $\mathbf{0} \prec \mathbf{\Sigma}_{i,k+1} \preceq \mathbf{\Sigma}_{i,k}$.

Proof. Under the assumptions in Theorem 4, $\mathbf{\Sigma}_{i,k}$ and $\mathbf{\Sigma}_{i,k+1}$ are positive definite matrices because they represent Gaussian covariance matrices. From the block matrix inversion in (A.3) the difference between two consecutive covariance matrices

can be written as

$$\boldsymbol{\Sigma}_{i,k} - \boldsymbol{\Sigma}_{i,k+1} = \mathbf{R}\mathbf{Q}^{-1}\mathbf{R}^T \tag{5.48}$$

From Lemma 5, $\boldsymbol{\phi}_0 \preceq \mathbf{Q}$ and, since $\mathbf{Q}$ is a diagonal positive-definite matrix, it follows that $\mathbf{Q}^{-1}$ is diagonal and positive definite. Then, there exists a diagonal positive-definite matrix, $\mathbf{V}$, such that $\mathbf{Q}^{-1} = \mathbf{V}\mathbf{V}^T$. Plugging $\mathbf{V}$ in (5.48) yields that

$$\boldsymbol{\Sigma}_{i,k} - \boldsymbol{\Sigma}_{i,k+1} = \mathbf{R}\mathbf{V}\mathbf{V}^T\mathbf{R}^T = (\mathbf{R}\mathbf{V})(\mathbf{R}\mathbf{V})^T \succeq \mathbf{0} \tag{5.49}$$

□

The third and final lemma provides a bound on the trace of the quadratic form $\mathbf{R}^T\mathbf{R}$ that is later used to show that the Monte Carlo integration (5.37) is an unbiased estimator of the approximate DPGP-EKLD in (5.35).

Lemma 7. *Under the assumptions in Theorem 4, the cross-covariance matrix* $\mathbf{R}$*, obtained from the target velocity and position estimates at the collocation points, and defined in* (5.19)*, obeys the inequality* $0 \leqslant \operatorname{tr}(\mathbf{R}\mathbf{R}^T) \leqslant 4k[(\boldsymbol{\phi}_0) + 2\sigma_v^2]$*, where* $\boldsymbol{\phi}_0 = \boldsymbol{\phi}(\mathbf{0}, \mathbf{0})$ *is a constant matrix obtained by evaluating the stationary covariance matrix function at zero,* $\sigma_v \in \mathbb{R}$ *is the standard deviation of the measurement noise of velocity, and* $k \in \mathbb{Z}_+$ *is the time index.*

Proof. Since $\mathbf{R}^T\mathbf{R}$ is a positive semi-definite matrix ($\mathbf{R}^T\mathbf{R} \succcurlyeq 0$), it has positive or zero eigenvalues, and

$$\operatorname{tr}(\mathbf{R}\mathbf{R}^T) \geqslant 0 \tag{5.50}$$

From GP regression, the joint probability distribution of a vector $[\boldsymbol{\upsilon}_i^T \quad \mathbf{v}_j(k+1)^T \quad \mathbf{f}_i(\mathbf{y}_1)^T \quad \mathbf{f}_i(\mathbf{y}_2)^T \quad \cdots]^T$, comprised of target velocities at the collocation points (5.13) and at the measured target positions, is a multivariate Gaussian distribution with covariance matrix,

$$\begin{bmatrix} \boldsymbol{\Phi}(\boldsymbol{\xi}, \boldsymbol{\xi}) + \sigma_v^2\mathbf{I}_{2L} & \boldsymbol{\Phi}[\boldsymbol{\xi}, \mathbf{x}_j(k+1)] & \boldsymbol{\Phi}[\boldsymbol{\xi}, \mathbf{Y}_i(k)] \\ \boldsymbol{\Phi}[\mathbf{x}_j(k+1), \boldsymbol{\xi}] & \boldsymbol{\Phi}[\mathbf{x}_j(k+1), \mathbf{x}_j(k+1)] + \sigma_v^2\mathbf{I}_2 & \boldsymbol{\Phi}[\mathbf{x}_j(k+1), \mathbf{Y}_i(k)] \\ \boldsymbol{\Phi}[\mathbf{Y}_i(k), \boldsymbol{\xi}] & \boldsymbol{\Phi}[\mathbf{Y}_i(k), \mathbf{x}_j(k+1)] & \boldsymbol{\Phi}[\mathbf{Y}_i(k), \mathbf{Y}_i(k)] + \sigma_v^2\mathbf{I}_{2k} \end{bmatrix}$$

The conditional marginal distribution of the vector $[\boldsymbol{\upsilon}_i^T \quad \mathbf{v}_j(k+1)^T]^T$, given the vector of target position measurements $\mathbf{Y}_i(k)$, is a multivariate Gaussian distribution with covariance matrix,

$$\begin{bmatrix} \boldsymbol{\Phi}(\boldsymbol{\xi},\boldsymbol{\xi}) + \sigma_v^2\mathbf{I}_{2L} & \boldsymbol{\Phi}[\boldsymbol{\xi}, \mathbf{x}_j(k+1)] \\ \boldsymbol{\Phi}[\mathbf{x}_j(k+1), \boldsymbol{\xi}] & \boldsymbol{\Phi}[\mathbf{x}_j(k+1), \mathbf{x}_j(k+1)] + \sigma_v^2\mathbf{I}_2 \end{bmatrix}$$

$$- \begin{bmatrix} \boldsymbol{\Phi}[\boldsymbol{\xi}, \mathbf{Y}_i(k)] \\ \boldsymbol{\Phi}[\mathbf{x}_j(k+1), \mathbf{Y}_i(k)] \end{bmatrix} \{\boldsymbol{\Phi}[\mathbf{Y}_i(k), \mathbf{Y}_i(k)] + \sigma_v^2\mathbf{I}_{2k}\}^{-1} \begin{bmatrix} \boldsymbol{\Phi}[\boldsymbol{\xi}, \mathbf{Y}_i(k)] \\ \boldsymbol{\Phi}[\mathbf{x}_j(k+1), \mathbf{Y}_i(k)] \end{bmatrix}^T$$

$$= \begin{bmatrix} \boldsymbol{\Sigma}_{i,k} & \mathbf{R} \\ \mathbf{R}^T & \mathbf{Q} \end{bmatrix} \tag{5.51}$$

Because the covariance matrix in (5.51) is symmetric and positive definite, the off-diagonal elements of $\mathbf{R}$ are smaller than the corresponding diagonal elements of $\mathbf{Q}$, or

$$\mathbf{R}_{(i,j)} < \mathbf{Q}_{(j,j)} < \text{tr}(\mathbf{Q}), \quad \forall i, j \tag{5.52}$$

From Lemma 5, $\mathbf{Q} \preceq \boldsymbol{\Phi}_0 + \sigma_v^2\mathbf{I}_2$, therefore, it follows that,

$$\mathbf{R}_{(i,j)} < \text{tr}(\boldsymbol{\phi}_0) + 2\sigma_v^2 \tag{5.53}$$

and from the properties of quadratic forms the following holds,

$$\text{tr}(\mathbf{R}\mathbf{R}^T) = \sum_{i=1}^{M}\sum_{j=1}^{N}[\mathbf{R}_{(i,j)}]^2 < \sum_{i=1}^{M}\sum_{j=1}^{N}[\text{tr}(\boldsymbol{\phi}_0) + 2\sigma_v^2] = 4k[\text{tr}(\boldsymbol{\phi}_0) + 2\sigma_v^2] \tag{5.54}$$

completing the proof.

□

We are now ready to prove the following theorem on the properties of the DPGP-EKLD approximation presented in this subsection:

Theorem 8. *Under the assumptions in Theorem 4, the Monte Carlo integration* (5.37) *is an unbiased estimator of the approximate DPGP-EKLD function in* (5.35)*, and the variance of the difference between* (5.35) *and* (5.37) *decreases linearly with*

the number of samples, S, that are drawn independently and identically from the target state distribution (5.33).

Proof. From the linearity of the expectation operation, the expected value of the DPGP-EKLD obtained via Monte Carlo integration in (5.37), is given by,

$$\begin{aligned}\bar{D} = \mathbb{E}\{\hat{D}\left(\boldsymbol{v}; \mathbf{m}(k+1)\right)\} =& \mathbb{E}\left\{\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{w_{ij}}{S}\sum_{s=1}^{S}\left[h_i(\boldsymbol{\chi}^{(s)})\mathbf{1}_{\mathcal{S}(k+1)}(\boldsymbol{\chi}^{(s)})\right]\right\} \\ =& \sum_{i=1}^{M}\sum_{j=1}^{N}\frac{w_{ij}}{S}\sum_{s=1}^{S}\mathbb{E}\left\{h_i(\boldsymbol{\chi}^{(s)})\mathbf{1}_{\mathcal{S}(k+1)}(\boldsymbol{\chi}^{(s)})\right\}\end{aligned} \tag{5.55}$$

where $\mathbb{E}\{\cdot\}$ denotes the expectation with respect to $\boldsymbol{\chi}^{(s)}$. Since the Monte Carlo integration samples, $\boldsymbol{\chi}^{(s)}$, $s = 1, \ldots, S$, are drawn identically and independently (iid) from the target state distribution $p\left(\mathbf{x}_j(k+1)|G_j(k+1) = i\right)$ in (5.33), the following equation holds,

$$\mathbb{E}\left\{h_i(\boldsymbol{\chi}^{(s)})\mathbf{1}_{\mathcal{S}(k+1)}(\boldsymbol{\chi}^{(s)})\right\} = \mathbb{E}\left\{h_i\big[\mathbf{x}_j(k+1)\big]\mathbf{1}_{\mathcal{S}(k+1)}[\mathbf{x}_j(k+1)]\right\} \tag{5.56}$$

for all $s = 1, \ldots, S$, where $h_i\big[\mathbf{x}_j(k+1)\big]$ is defined in (5.36) for $G_j(k+1) = i$, and thus the estimator (5.37) is unbiased.

Now, let $\mathrm{var}(\cdot)$ denote the variance of a random variable. Because the samples

$\boldsymbol{\chi}^{(s)}$, $s = 1, \ldots, S$, are drawn i.i.d., the following holds,

$$\begin{aligned}
\mathrm{var}(\hat{D}) &= \mathbb{E}\{(\hat{D} - \bar{D})^2\} \\
&= \mathbb{E}\Bigg\{ \sum_{j=1}^{N} \sum_{i=1}^{M} w_{ij} \Bigg[\frac{1}{S} \sum_{s=1}^{S} h_i\big(\boldsymbol{\chi}^{(s)}\big) \mathbf{1}_{\mathcal{S}(k+1)}\big(\boldsymbol{\chi}^{(s)}\big) \\
&\qquad\qquad - \mathbb{E}\Big\{ h_i\big[\mathbf{x}_j(k+1)\big] \mathbf{1}_{\mathcal{S}(k+1)}\big[\mathbf{x}_j(k+1)\big] \Big\} \Bigg]^2 \Bigg\} \\
&= \sum_{j=1}^{N} \sum_{i=1}^{M} w_{ij} \frac{1}{S^2} \sum_{s=1}^{S} \mathbb{E}\Bigg\{ \Bigg[h_i\big(\boldsymbol{\chi}^{(s)}\big) \mathbf{1}_{\mathcal{S}(k+1)}\big(\boldsymbol{\chi}^{(s)}\big) \\
&\qquad\qquad - \mathbb{E}\Big\{ h_i\big[\mathbf{x}_j(k+1)\big] \mathbf{1}_{\mathcal{S}(k+1)}\big[\mathbf{x}_j(k+1)\big] \Big\} \Bigg]^2 \Bigg\} \\
&= \frac{1}{S} \sum_{j=1}^{N} \sum_{i=1}^{M} w_{ij} \mathrm{var}\Big\{ h_i\big[\mathbf{x}_j(k+1)\big] \mathbf{1}_{\mathcal{S}(k+1)}\big[\mathbf{x}_j(k+1)\big] \Big\} \\
&= \frac{1}{S} \mathrm{var}\Big\{ h_i\big[\mathbf{x}_j(k+1)\big] \mathbf{1}_{\mathcal{S}(k+1)}\big[\mathbf{x}_j(k+1)\big] \Big\}
\end{aligned} \tag{5.57}$$

where $\mathrm{var}\left\{ h_i\big[\mathbf{x}_j(k+1)\big] \mathbf{1}_{\mathcal{S}(k+1)}\big[\mathbf{x}_j(k+1)\big] \right\}$ is a finite constant, and can be proved by showing that $h_i[\cdot]$ is finite-valued as follows. First, we prove that $h_i[\cdot]$ is greater than or equal to zero. From (5.36), $h_i[\cdot]$ is the integral of the KL divergence weighted by a Gaussian distribution. Since the KL divergence is always greater than or equal to zero, and $p(\mathbf{z}_j(k+1)|\mathbf{x}_j(k+1)) \geqslant 0$, it also follows that

$$h_i[\cdot] \geqslant 0 \tag{5.58}$$

Second, we prove that $h_i[\cdot]$ is less than infinity, by showing that every term of $h_i[\cdot]$ is finite-valued. Recalling the second term in (5.36), the first term of $h_i[\cdot]$ can be shown finite-valued by following Lemma 6 as follows,

$$\begin{aligned}
\mathrm{tr}(\boldsymbol{\Sigma}_{i,k}^{-1} \boldsymbol{\Sigma}_{i,k+1}) &= \mathrm{tr}[\boldsymbol{\Sigma}_{i,k}^{-1} (\boldsymbol{\Sigma}_{i,k} - \mathbf{R}\mathbf{Q}^{-1}\mathbf{R}^T)] \\
&= \mathrm{tr}(\mathbf{I}_{2L} - \boldsymbol{\Sigma}_{i,k}\mathbf{R}\mathbf{Q}^{-1}\mathbf{R}^T) \leqslant 2L
\end{aligned} \tag{5.59}$$

where L is the number of collocation points in $\mathcal{W}$. The second term of $h_i[\cdot]$ can also be shown finite-valued. From Lemma 6, it follows that $|\mathbf{\Sigma}_{i,k}| > 0$ and $|\mathbf{\Sigma}_{i,k+1}| > 0$, since the two covariance matrices are both positive definite. Since $\mathbf{\Sigma}_{i,k}$ and $\mathbf{\Sigma}_{i,k+1}$ are Hermitian and $\mathbf{\Sigma}_{i,k} \succeq \mathbf{\Sigma}_{i,k+1}$, then $|\mathbf{\Sigma}_{i,k}| > |\mathbf{\Sigma}_{i,k+1}|$ and $0 < |\mathbf{\Sigma}_{i,k+1}|/|\mathbf{\Sigma}_{i,k}| < 1$, such that $0 < -\ln(|\mathbf{\Sigma}_{i,k+1}|/|\mathbf{\Sigma}_{i,k}|) < \infty$. The last term of $h_i[\cdot]$ in (5.36) can be shown finite-valued from the property that the trace of a product of matrices is invariant under cyclic permutations [151], such that

$$\mathrm{tr}(\mathbf{Q}^{-1}\mathbf{R}^T\mathbf{\Sigma}_{i,k}^{-1}\mathbf{R}\mathbf{Q}^{-1})\sigma_v^2 = \mathrm{tr}(\mathbf{\Sigma}_{i,k}^{-1}\mathbf{R}\mathbf{Q}^{-1}\mathbf{Q}^{-1}\mathbf{R}^T)\sigma_v^2 \tag{5.60}$$

In addition, for positive semi-definite matrices of the same size, the trace of the products is less than or equal to the product of traces [151], such that

$$\begin{aligned}
\mathrm{tr}(\mathbf{\Sigma}_{i,k}^{-1}\mathbf{R}\mathbf{Q}^{-1}\mathbf{Q}^{-1}\mathbf{R}^T)\sigma_v^2 &\leqslant \mathrm{tr}(\mathbf{\Sigma}_{i,k}^{-1})\mathrm{tr}(\mathbf{R}\mathbf{Q}^{-1}\mathbf{Q}^{-1}\mathbf{R}^T)\sigma_v^2 \\
&= \mathrm{tr}(\mathbf{\Sigma}_{i,k}^{-1})\mathrm{tr}(\mathbf{Q}^{-1}\mathbf{Q}^{-1}\mathbf{R}^T\mathbf{R})\sigma_v^2 \\
&\leqslant \mathrm{tr}(\mathbf{\Sigma}_{i,k}^{-1})[\mathrm{tr}(\mathbf{Q}^{-1})]^2\mathrm{tr}(\mathbf{R}^T\mathbf{R})\sigma_v^2 \\
&= \mathrm{tr}(\mathbf{\Sigma}_{i,k}^{-1})[\mathrm{tr}(\mathbf{Q}^{-1})]^2\mathrm{tr}(\mathbf{R}\mathbf{R}^T)\sigma_v^2 \\
&\leqslant 4\mathrm{tr}(\mathbf{\Sigma}_{i,k}^{-1})\left[4k\mathrm{tr}(\mathbf{\Phi}_0) + 2\sigma_v^2\right]\sigma_v^2
\end{aligned} \tag{5.61}$$

where the last inequality is the result of Lemmas 5-7. Therefore, every term of $h_i[\cdot]$ has been proved to be finite-valued, which proves that $h_i[\cdot]$ is also finite-valued for all $\mathbf{x}_j(k+1) \in \mathbb{R}^2$. Due to the indicator function $\mathbf{1}_{\mathcal{S}(k+1)}[\cdot]$, the variance $\mathrm{var}\left\{h_i\big[\mathbf{x}_j(k+1)\big]\mathbf{1}_{\mathcal{S}(k+1)}\big[\mathbf{x}_j(k+1)\big]\right\}$ in (5.57) is a definite integral over $\mathcal{S}(k+1)$ of finite-valued $h_i[\cdot]$, therefore, it is a finite constant. Thus the error variance of the Monte Carlo estimator in (5.37) is proportional to $1/S$ [152].

□

5.3.5 Cumulative Lower Bound of DPGP Expected KL-Divergence

The DPGP-EKLD discussed in Sections 5.3.1-5.3.4 only considers one future sensor measurement, $\mathbf{m}_j(k+1)$, therefore, it can only be applied to develop greedy sensor planning algorithms. The performance of the greedy sensor planning algorithms is acceptable if no constraints on the sensor dynamics are imposed, and the sensor can be treated as a free-flying object. However, when constraints of the sensor dynamics, such as the linear dynamics with constrained input in (3.6), greedy algorithms suffer from local minima and the performance is usually impaired greatly. Therefore, DPGP-EKLD considering multiple future measurements needs to be examined.

Recall that $\mathcal{M}_j(k_1,k_2) \triangleq \{\mathbf{m}_j(\ell) \mid k_1 \leqslant \ell < k_2\}$ denote the measurements obtained from the jth target between time steps k_1 and k_2, and that $\mathcal{M}(k_1,k_2) \triangleq \bigcup_{j=1}^{N} \mathcal{M}_j(k_1,k_2)$. Then, the DPGP-EKLD for K future measurements of the jth target until time step $k' = k+K$, $\mathcal{M}_j(k,k')$, is defined by the prior and posterior distributions of $\boldsymbol{\upsilon}$ as follows,

$$\hat{D}\left(\boldsymbol{\upsilon}; \mathcal{M}_j(k,k')\right) \triangleq \mathbb{E}_{\mathcal{M}_j(k,k')}\left\{\mathbb{E}_{G_j}\left\{D\big(\boldsymbol{\upsilon}; \mathcal{M}_j(k,k')\big)\right\}\right\}$$

where,

$$D\big(\boldsymbol{\upsilon}; \mathcal{M}_j(k,k')\big) \triangleq D\left(p\big(\boldsymbol{\upsilon}|\mathcal{M}_j(k,k')\bigcup \mathcal{M}(1,k)\big) \parallel p\big(\boldsymbol{\upsilon}|\mathcal{M}(1,k)\big)\right) \tag{5.62}$$

It is worth noticing that $\hat{D}\left(\boldsymbol{\upsilon}; \mathcal{M}_j(k,k')\right)$ can also be expressed in terms of the mutual information (MI). Recall that $I(X;Y) \triangleq \mathbb{E}_Y\{D[p_{X|Y}(x|y)\|p_X(x)]\}$ denote the mutual information between two continuous random variables, X and Y. Then, the DPGP-EKLD is equivalent to,

$$\hat{D}\left(\boldsymbol{\upsilon}; \mathcal{M}_j(k,k')\right) = \sum_{i=1}^{M} w_{ij} I\big(\boldsymbol{\upsilon}_i; \mathcal{M}_j(k,k')\big) \tag{5.63}$$

where w_{ij} is the posterior probability of $\{G_j = i\}$. Let $\mathcal{M}_j^c(1,k) = \mathcal{M}(1,k)\backslash\mathcal{M}_j(1,k)$ denote the complement set of $\mathcal{M}_j(1,k)$ in $\mathcal{M}(1,k)$. The value of w_{ij} can be obtained

from Bayes' rule as follows,

$$w_{ij} = \frac{\pi_i \cdot p(\mathcal{M}_j(1,k)|\mathcal{M}_j^c(1,k), G_j = i)}{\sum_{i=1}^{M} \pi_i \cdot p(\mathcal{M}_j(1,k)|\mathcal{M}_j^c(1,k), G_j = i)} \tag{5.64}$$

where $\pi_i = p(G_j = i)$, is the prior probability that the jth target follows the ith VF.

However, it can be proved as follows that optimizing the DPGP-EKLD (5.63) is NP-hard under the constraints of the camera dynamics (3.6) and the bounded FOV. Thus, using the DPGP-EKLD directly as the design objective for controlling the sensor for a finite horizon is computationally intractable. In order to study the complexity for optimizing the DPGP-EKLD, Lemma 17 is introduced in Appendix A.4 for integrity [153]. Based on Lemma 17, Theorem 9 is proposed as follows.

Theorem 9. *Given a rational number m and a rational covariance matrix $\mathbf{\Lambda}$ of a set of Gaussian random variables $V = S \bigcup U$, deciding whether there exists a subset $A \subset S$ of cardinality d such that $I(A;U) \geqslant m$ is NP-complete.*

Proof. Let Q_1 and Q_2 denote the problems studied in Lemma 17 and Theorem 9, respectively. The NP-completeness of Q_2 can be shown by the reduction from Q_1 to Q_2 as follows. Recall that S denotes the set of Gaussian random variables in Q_1, with rational covariance matrix $\mathbf{\Sigma}$. Let σ_c denote a positive constant that is less than the square root of the smallest eigenvalue of $\mathbf{\Sigma}$. For any S, there exits a set U, such that the random variables in $V = S \cup U$ are multivariate Gaussian distributed with covariance matrix $\mathbf{\Lambda} = \begin{bmatrix} \mathbf{\Sigma} & \mathbf{I} \\ \mathbf{I} & (\mathbf{\Sigma} - \sigma_c^2\mathbf{I})^{-1} \end{bmatrix}$. Then, the conditional distribution of S given U is a multivariate Gaussian distribution with covariance matrix $\sigma_c^2\mathbf{I}$. Therefore, the random variables in S are conditionally independent given U.

The mutual information studied in Q_2 can be decomposed into two parts, that is, $I(A;U) = H(A) - H(A|U)$, where $H(\cdot)$ denotes the entropy function. The first part, $H(A)$, is the same quantity to be maximized in Q_1. The second part, $H(A|U)$, can

be proven to be a constant as follows. Since $A \subset S$, the random variables in A are also conditionally independent given U. Thus, $H(A|U) = d\log(2\pi e\sigma^2)/2$, which is a constant since the cardinality of A is part of the question. Therefore, maximizing $I(A;U)$ is equivalent to maximizing $H(A)$, which proves that the solution to Q_2 can be used as a black-box to solve Q_1 in polynomial steps. Since Q_1 is NP-complete, Q_2 is NP-hard. In addition, calculating the mutual information given a set of Gaussian random variables requires polynomial time. Therefore, Q_2 is NP-complete. □

Based on Theorem 9, the complexity of optimizing the DPGP-EKLD in (5.63) can be addressed by Theorem 10 as follows.

Theorem 10. *Determining the optimal control trajectory,* $\mathbf{u}^*(\ell),\ \ell = k, \ldots, k'$*, that maximizes the DPGP-EKLD,* $\hat{D}\left(\boldsymbol{\upsilon}; \mathcal{M}_j(k,k')\right)$*, is NP-hard, under the constraints of the sensor dynamics* (3.6) *and the bounded FOV.*

Proof. An NP-hardness proof by restriction for a given problem consists of showing that the given problem contains another known NP-complete problem as a special case, by placing additional restrictions on the given problem [154]. Let Q_3 denote the problem studied in Theorem 10. Let Q_2 denote the known NP-complete problem in Theorem 9. In order to prove the NP-hardness of Q_3 by restriction, the following three additional restrictions are required for indicating that Q_2 is a special case of Q_3. The first restriction is that the number of velocity fields equals one, that is, $M = 1$. Without loss of generality, let $\mathbf{f}_1$ denote the VF that all the N targets are associated with. Then, $\{\mathbf{f}_1[\mathbf{x}_j(\ell)] \mid 1 \leqslant j \leqslant N,\ k \leqslant \ell \leqslant k'\}$ can be selected to be equivalent to S. The second restriction is that the camera FOV is small enough such that the camera can obtain only one target measurement at every time step. Then, the cardinality of $\mathcal{M}_j(k,k')$ is $K = k' - k$, which can be chosen to be the cardinality d in Q_2. The last restriction is placed on $\boldsymbol{\xi}_l$, such that $\{\mathbf{f}_1(\boldsymbol{\xi}_l) \mid 1 \leqslant l \leqslant L\} = U$.

After the specification of the above restrictions, the resulting restricted problem of Q_3 is identical to Q_2. Since Q_2 is NP-complete, Q_3 is NP-hard. $\square$

Since optimizing the DPGP-EKLD (5.63) is NP-hard as shown in Theorem 10, approximation techniques are required in order to reduce the computational complexity for determining the optimal control trajectory. One approach is to maximize the lower bound of the DPGP-EKLD, which optimizes the worst-case performance, as shown in the next section [155].

The complexity of optimizing the DPGP-EKLD stems from that $\hat{D}\left(\boldsymbol{\upsilon}; \mathcal{M}_j(k,k')\right)$ is a function of the set of future measurements, $\mathcal{M}_j(k,k')$. Therefore, maximizing the DPGP-EKLD is a combinatorial optimization problem. The complexity can be greatly reduced if the objective function can be written as a summation of the reward at every time step. To this end, the cumulative lower bound of the DPGP-EKLD is studied, and is presented in the following theorem.

Theorem 11. *The DPGP-EKLD, $\hat{D}\left(\boldsymbol{\upsilon}; \mathcal{M}_j(k,k')\right)$, is lower bounded by the discounted summation of the mutual information between the random variables indexed at the collocation points, $\boldsymbol{\upsilon}$, and the measurement at every time step, $\mathbf{m}_j$, such that,*

$$\hat{D}\big(\boldsymbol{\upsilon}; \mathcal{M}_j(k,k')\big) \geqslant \sum_{i=1,\ell=k}^{M,k'} w_{ij}(1-\gamma)\gamma^{\ell-k} I\big(\boldsymbol{\upsilon}_i; \mathbf{m}_j(\ell)\big) \triangleq \hat{D}_L\big(\boldsymbol{\upsilon}; \mathcal{M}_j(k,k')\big) \quad (5.65)$$

where $\gamma \in (0,1)$ is a user-defined discount factor.

Proof. Let V, A_1, and A_2 denote any three sets of random variables. Let $A \triangleq A_1 \cup A_2$ denote the union of A_1 and A_2. Since $A_1, A_2 \subset A$, it follows that $H(V|A) \leqslant H(V|A_1)$, and that $H(V|A) \leqslant H(V|A_2)$ [143]. Thus, for any $\gamma \in (0,1)$, the following inequality holds,

$$I(V;A) \geqslant \gamma I(V;A_1) + (1-\gamma) I(V;A_2) \quad (5.66)$$

By applying the inequality (5.66) to $I\big(\boldsymbol{\upsilon}_i; \mathcal{M}_j(k,k')\big)$ for K times, it follows that,

$$I\big(\boldsymbol{\upsilon}_i; \mathcal{M}_j(k,k')\big) \geqslant \sum_{\ell=k}^{k'} (1-\gamma)\gamma^{\ell-k} I\big(\boldsymbol{\upsilon}_i; \mathbf{m}_j(\ell)\big) \tag{5.67}$$

Finally, substituting (5.67) in (5.63) yields the cumulative lower bound of the DPGP-EKLD,

$$\hat{D}\big(\boldsymbol{\upsilon}; \mathcal{M}_j(k,k')\big) \geqslant \sum_{i=1,\ell=k}^{M,k'} w_{ij}(1-\gamma)\gamma^{\ell-k} I\big(\boldsymbol{\upsilon}_i; \mathbf{m}_j(\ell)\big) = \hat{D}_L\big(\boldsymbol{\upsilon}; \mathcal{M}_j(k,k')\big) \tag{5.68}$$

which completes the proof. □

5.4 Chapter Conclusion

Novel information theoretic functions for the GP target kinematics model and the DPGP target kinematics model presented in Chapter 4 have been developed. The GP expected KL divergence (5.15) is derived to evaluate the improvement of the GP target kinematics model brought about by a single future measurement. The analytical expression of the GP expected KL divergence is derived and presented in Theorem 4. In addition, the DPGP expected KL divergence (5.28) is proposed for calculating the information value of a single future measurement for learning the DPGP the DPGP target kinematics model. New theoretic results (Theory 8) are used to obtain a computationally efficient approximation of DPGP expected KL divergence via Monte Carlo integration. The new analysis proves that this approximation is an unbiased estimator of the original DPGP value function, and is characterized by an error covariance that decreases linearly with the number of samples. Moreover, the DPGP expected KL divergence is extended to (5.62) that considers multiple future measurements. A novel theorem (Theorem 10) shows that optimizing the DPGP expected KL divergence for multiple sensor measurements in (5.62) subject to sensor dynamics and FOV constraints is NP-hard. A new cumulative lower bound is

then derived in Theorem 11 for the purpose of reducing the computational complexity of the sensor planning problem. The novel Bayesian nonparametric information theoretic functions and their bounds are used to develop efficient sensor planning algorithms in subsequent chapters.

6

Sensor Planning for Bayesian Nonparametric Target Modeling

The problem of determining the optimal control sensor planning strategy often consists of maximizing or minimizing an objective function subject to the constraints on the sensor dynamics. In the case of sensor planning for Bayesian nonparametric target modeling, the objectives can be expressed by the novel information value functions presented in Chapter 5. Specific forms of the assumptions on the targets and the sensor dynamics also need to be considered for the purpose of developing the most suitable control strategy. Moreover, these assumptions need to be the abstract representations of a broad range of real-life applications, such that the theoretic works can be applied in practice. In order to demonstrate that the sensor planning algorithms with respect to the target kinematics models proposed in Chapter 4 and the information values proposed in Chapter 5 can be broadly applied, a variety of application scenarios are considered according to various practical assumptions on the sensor dynamics and the target kinematics. First, when the target kinematics can be described by a time-invariant velocity field, such as the ocean current in a

relative short period of time, a single GP can be applied. In this case, a novel greedy algorithm is developed to determine the optimal sensing sequence, which is presented in Section 6.1. Second, problems involving mobile targets are addressed by the scenario in Section 6.2, where a sweep line algorithm is developed for unconstrained sensor dynamics with continuous state space. Third, constrained sensor dynamics are studied in Section 6.3, and a lexicographic algorithm is proposed.

6.1 Scenario 1: Always Observable Target and Discrete Control Space

In many applications, the measurement of the target kinematics can be assumed available for any target state, $\mathbf{x} \in \mathcal{W}$, and for any time step, k. This assumption is valid for applications where the target kinematics are modeled by a steady velocity field, $\mathbf{f}_i : \mathcal{W} \to \mathbb{R}^d$ defined in (3.3), where d is the dimension of the target state. For example, in problems of studying the kinematics of drifters in the ocean or the kinematics of airborne robots (such as the "smart dust" particles [156]), the ocean surface current and the wind velocity can be treated as time-invariant velocity fields for a short period of time [157]. The measurement of the ocean surface current or the wind velocity is available at any position in the workspace as long as the sensor is deployed at that position. In these applications, the set of admissible sensor states $\mathcal{A}$, defined in (3.6), is often finite and known *a-priori* [101]. For example, measurements of the ocean current or the wind velocity may be only available at a finite number of moored buoys or climate stations. Comparing to the area of the workspace, the size of the moored buoy or the climate station is negligible. Therefore, the sensor FOV can be treated as a volume-less point overlapping with the sensor state, $\mathbf{s}(k)$, and the measurement of the target state, $\mathbf{y}(k)$, is noise-free. Nevertheless, noise of the measurement of the target kinematics, $\mathbf{z}(k)$, can be considered, and the general

measurement model in (3.7) can be specialized to,

$$\mathbf{m}(k) = [\mathbf{y}^T(k) \quad \mathbf{z}^T(k)]^T = [\mathbf{s}^T(k) \quad \mathbf{v}^T(k)]^T + [\mathbf{0}_d^T \quad \boldsymbol{\nu}_v^T]^T \tag{6.1}$$

where $\mathbf{0}_d$ is a $d \times 1$ vector of zeros, and $\boldsymbol{\nu}_v \in \mathbb{R}^d$ is the Gaussian distributed velocity measurement noise with zero mean and covariance matrix, $\sigma_v^2 \mathbf{I}_d$. Notice that the FOV constraint is implied by substituting $\mathbf{x}$ with $\mathbf{s}(k)$ in (6.1).

No constraint on the sensor dynamics is considered, such that the sensor can be treated as a free-flying object. In other words, the sensor is able to obtain one measurement at any state in $\mathcal{A}$ at every time step, regardless of the history of the sensor states. Then, the general state-space representation of the sensor dynamics in (3.6) can be specialized to,

$$\mathbf{s}(k+1) = \mathbf{u}(k), \quad \text{and,} \ \ \mathbf{u}(k) \in \mathcal{U} = \mathcal{A} \tag{6.2}$$

where the admissible control space, $\mathcal{U}$, defined in (3.6), is equivalent to $\mathcal{A}$.

One Gaussian process is sufficient for describing the target kinematics for the scenario discussed in this section. Therefore, the GP-EKLD presented in Section 5.2 can be utilized as the objective function in the sensor planning problem, that is,

$$\mathscr{L}[\mathbf{s}(k), \mathbf{u}(k) \mid \mathcal{M}(k)] = \hat{D}\big(\boldsymbol{\upsilon}_i; \mathbf{m}(k+1)\big) \tag{6.3}$$

where $\boldsymbol{\upsilon}_i$ is the vector define in (5.13) that represents the velocity field $\mathbf{f}_i$. The left hand side and the right hand side of (6.3) are related by the sensor dynamics, (6.2), and the measurement model, (6.1).

Sensor planning in the scenario characterized by (6.1)-(6.3). A greedy algorithm can be developed. First, the GP-EKLDs (6.3) for all admissible sensor states in $\mathcal{A}$ are calculated. Then, the optimal sensor state is selected by maximizing the GP-EKLD (6.3). Thereafter, the estimation of the target kinematics $\hat{\mathbf{f}}_i[\mathbf{x}|\mathcal{M}(k)]$ is calculated given the new measurement $\mathbf{m}(k)$, and the previous estimation $\hat{\mathbf{f}}_i[\mathbf{x}|\mathcal{M}(k-$

1)]. Hyper-parameters of the GP, Θ, can be optimized after the new measurement is obtained. The algorithm determines the measurement sequence one at a time, and is referred to as the GP-EKLD-Greedy algorithm. Let K denote the maximum number of observations, the GP-EKLD-Greedy algorithm is summarized as follows:

Algorithm 7 GP-EKLD-Greedy

Input: GP parameters, $\{\boldsymbol{\theta}_i(\cdot), \boldsymbol{\phi}_i(\cdot,\cdot)\}$; GP hyper-parameters, Θ_i; Set of admissible sensor state, $\mathcal{A}$; Collocation points, $\boldsymbol{\xi}$; Control horizon, K

Output: Optimal control inputs, $\{\mathbf{u}^*(k)\}_{k=1}^K$

1: **for** $k = 1, \ldots, K$ **do**
2: $\quad \mathbf{u}^*(k+1) = \underset{\mathbf{s}\in\mathcal{A}}{\operatorname{argmax}}\ \hat{D}\big(\boldsymbol{\upsilon}_i; \mathbf{m}(k+1)\big)$
3: $\quad \mathbf{s}(k+1) = \mathbf{u}^*(k+1)$
4: $\quad$ Obtain $\mathbf{m}(k+1)$ according to (6.1)
5: $\quad \mathbf{Y}(k+1) = [\mathbf{Y}^T(k) \quad \mathbf{y}^T(k+1)]^T$
6: $\quad \mathbf{Z}(k+1) = [\mathbf{Z}^T(k) \quad \mathbf{z}^T(k+1)]^T$
7: $\quad$ Obtain optimal GP hyper-parameters, Θ_i^*, by (2.16)
8: **end for**

6.2 Scenario 2: Mobile Targets and Unconstrained Sensor Dynamics

Although the scenario considered in the previous section with target kinematics that are always observable can be applied in certain problems such as ocean surface current modeling, the assumption is too restricted and limits the applications of the Greedy-GP-EKLD algorithm (Algorithm 7). For problems involving mobile targets, such as pedestrians or ground vehicles, target tracking needs to be considered by the sensor planning algorithm, since the sensor is only able to obtain non-empty measurement of the target kinematics if the target is in the sensor FOV, as determined by the measurement model (3.7). To cope with the mobile targets, a novel sweep line algorithm is proposed as follows by using the GP particle filter or the DPGP particle filter presented in Chapter 4.

The target kinematics are assumed to be the same as presented in the problem formulation in (3.3), where the target kinematics are modeled by a mixture of M

unknown ODEs, $\mathcal{F} \triangleq \{\mathbf{f}_1, \ldots, \mathbf{f}_M\}$, with unknown mixture weights, $\boldsymbol{\pi}$. Notice that the following approach applies to the case where the target kinematics is modeled by a single VF automatically by imposing the constraint that $M = 1$.

The sensor is still assumed to be a free-flying object as in the previous section, however, continuous sensor state is utilized, such that the sensor dynamics are,

$$\mathbf{s}(k+1) = \mathbf{u}(k), \quad \mathbf{u}(k) \in \mathcal{U} = \mathcal{W} \tag{6.4}$$

Notice that the admissible control space is assumed equivalent to the admissible state space, such that $\mathcal{U} = \mathcal{W}$. The workspace is assumed to be two-dimensional, and the sensor FOV is assumed to be an axis-parallel rectangle with fixed shape, such that, $\mathcal{S}(k) \subset \mathcal{W}$, as shown in Fig. 6.3. This assumption is valid for a variety of applications, such as autonomous driving vehicle with Lidar [158]. The sensor is assumed to be able to take measurements of the target position and the target velocity,

$$\mathbf{m}_j(t) \triangleq [\mathbf{y}_j^T(t) \quad \mathbf{z}_j^T(t)]^T = [\mathbf{x}_j^T(t), \mathbf{v}_j^T(t)] + \boldsymbol{\nu}, \quad \text{if } \mathbf{x}_j(t) \in \mathcal{S}(t) \tag{6.5}$$

where $\boldsymbol{\nu}$ is an additive Gaussian distributed noise vector with zero mean and known covariance matrix $\text{diag}\big(\sigma_x^2\mathbf{I}_2, \sigma_v^2\mathbf{I}_2\big)$.

According to the above formulation of the problem, the target kinematics can be described by the DPGP mixture model (2.25). A general framework to the sensor planning with the DPGP mixture model is presented in Fig. 6.1. Without loss of generality the time index k is used to denote the time at which the DPGP-MM update is complete, and the position of the sensor FoV is decided for the next time step, $k+1$. With the DPGP-EKLD (5.37), the expected information value can be maximized with respect to the next sensor FoV location, $\mathcal{S}(k+1)$, such that the utility of the next sensor measurements is maximized. After a sufficient number of measurements $\mathcal{M}_j(k)$ is obtained from multiple targets, the DPGP-MM is updated

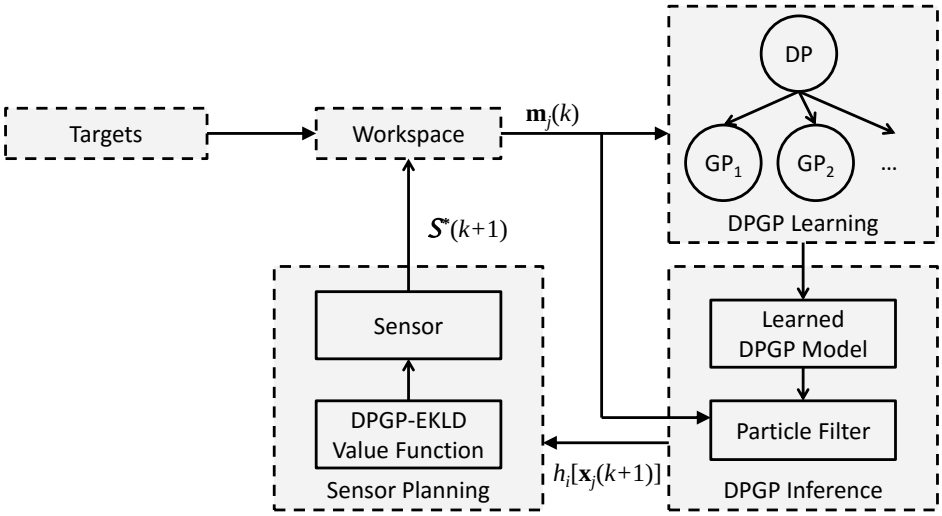

FIGURE 6.1: Diagram of DPGP-EKLD sensor planning framework and algorithms.

based on the complete database $Q(k)$, and the position of the sensor FoV for future time steps is decided according to the updated DPGP mixture model to further improve the DPGP-MM.

The sensor planning algorithm in the general framework in Fig. 6.1 is also a greedy algorithm, and can be developed by the reduction to a rectangle translation problem in the field of computational geometry. Recall from (5.37) that the DPGP-EKLD can be approximated by the weighted summation,

$$\hat{D}\left(\boldsymbol{v};\mathbf{m}(k+1)\right) \approx \sum_{j=1}^{N}\sum_{i=1}^{M}\sum_{s=1}^{S}\frac{w_{ij}}{S}h_i(\boldsymbol{\chi}^{(s)})\mathbf{1}_{\mathcal{S}(k+1)}(\boldsymbol{\chi}^{(s)}) \tag{6.6}$$

where the samples, $\{\boldsymbol{\chi}^{(s)}\}_{s=1}^{S}$, can be obtained from the DPGP particle filter (Algorithm 5 and Algorithm 6). Since all the samples for different targets following different VFs are summed together, do not need to be distinguished for the purpose of sensor planning. From the indicator function in (6.6), the optimal control $\mathbf{u}^*(k+1)$ can be obtained by the reduction to the following rectangle translation problem:

Problem 2 (Rectangle Translation [159])**.** *Given a two-dimensional, finite workspace*

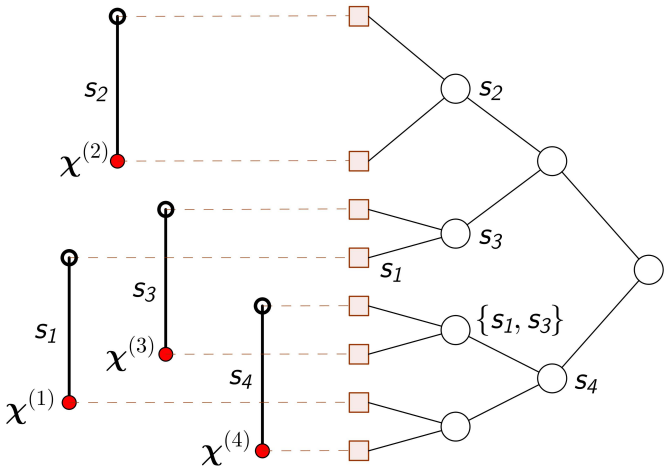

FIGURE 6.2: Example of segment a tree.

$\mathcal{W} \subset \mathbb{R}^2$, and a set of points, $\bigcup_{j=1}^{N}\bigcup_{i=1}^{M}\bigcup_{s=1}^{S}\{\boldsymbol{\chi}^{(s)}\} \subset \mathcal{W}$, each associated with a weight $\omega_{ij}h_i(\boldsymbol{\chi}^{(s)})/S$, find the translation of a rectangle, $\mathcal{S} \subset \mathcal{W}$, with fixed shape, such that the summation of the weights of the points covered by the rectangle is maximum.

There exists an efficient algorithm to Problem 2, with $O(MNS \log MNS)$ time complexity and $O(MNS \log MNS)$ space complexity, based on a data structure known as *segment tree* [160]. Assume that the sensor FOV is a $L_x \times L_y$ rectangle, the segment tree is build for all vertical segments of length L_y with their bottom ends at particles' y coordinates. An example of the segment tree is shown in Fig. 6.2, with four samples, $\{\boldsymbol{\chi}^{(s)}\}_{s=1}^{4}$. The segment tree is used to efficiently query the segments according to the following theorem:

Theorem 12 ([160])**.** *A segment tree for a set of n intervals uses $O(n \log n)$ storage can be built in $O(n \log n)$ time. Using the segment tree, all intervals that contain a query point can be reported in $O(\log n + m)$ time, where m is the number of reported intervals.*

The algorithm to solving Problem 2 a sweep-line algorithm, and is schematized in Fig. 6.3. It consists of five steps: (i) sort x-coordinates of all the samples

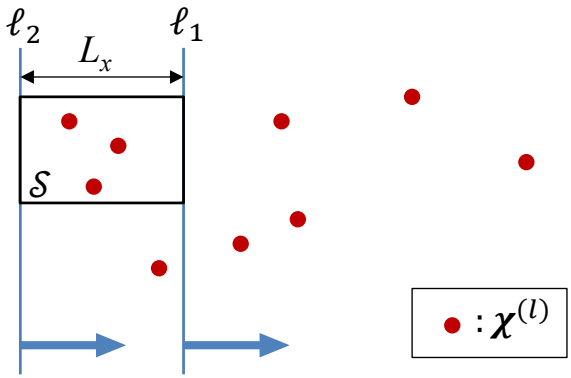

FIGURE 6.3: Schematic plot of the sweep-line algorithm.

$\bigcup_{j=1}^{N}\bigcup_{l=1}^{M}\bigcup_{s=1}^{S}\{\boldsymbol{\chi}^{(s)}\} \subset \mathcal{W}$; (ii) build a segment tree in for all vertical segments of length L_y with their bottom end at samples' y-coordinates, respectively, as shown in Fig. 6.2; (iii) associate a value to each vertical segment and initialize it as zero; (iv) sweep along sorted x-coordinates with two vertical lines, ℓ_1 and ℓ_2 with infinite height, as shown in Fig 6.3. The horizon distance between the two vertical lines stays as L_x throughout the sweeping. If a sample is swept by line ℓ_1, add the weight associated with the sample to all vertical segments containing it. If a sample is swept by line ℓ_2, remove the weight associated with the sample to all vertical segments containing it. (v) obtain the maximum value among all segments.

By using the sweep-line algorithm presented above as a sub-routine, the sensor planning to the scenario described by (6.4)-(6.5) can be developed by following the framework in 6.1. In order to present the sensor planning algorithm clearly, the pseudocode is summarized in Algorithm 8 as follows,

6.3 Scenario 3: Mobile Targets and Constrained Sensor Dynamics

The scenarios discussed in the previous two sections do not consider consider constraints on the sensor dynamics and model the sensor as free-flying object in the workspace either with a discrete or a continuous admissible control space. However, in real world applications, the sensor dynamics are always constrained with limits of

Algorithm 8 DPGP-EKLD-Greedy

Input: GP parameters, $\{\boldsymbol{\theta}_i(\cdot), \boldsymbol{\phi}_i(\cdot,\cdot)\}_{i=1}^{M}$; Collocation points, $\boldsymbol{\xi}$; Control horizon, K
Output: Optimal control inputs, $\{\mathbf{u}^*(k)\}_{k=1}^{K}$

1: **for** $k = 1, \ldots, K$ **do**
2: Obtain samples, $\bigcup_{j=1}^{N}\bigcup_{l=1}^{M}\bigcup_{s=1}^{S}\{\boldsymbol{\chi}^{(s)}\}$, from DPGP Particle Filter Time Update (Algorithm 5)
3: Calculate the DPGP-EKLD for every sample
4: Solve the Rectangle Translation problem by the sweep-line algorithm
5: Report the optimal FOV position, $\mathbf{u}^*(k)$
6: DPGP Particle Filter Measurement Update (Algorithm 6)
7: **end for**

the state and/or limits of the control vector. Considering the constraints on the sensor dynamics, the greedy algorithm is insufficient since it suffers from local optima. The performance of the greedy algorithm is especially deficient when mobile targets are under observation, since the local optima often result in empty measurements that contain no information of the target kinematics as shown by the simulation results in Chapter 7. To overcome the disadvantages of the greedy algorithm, this section proposes a lexicographic method presented as follows.

As stated by the sensor planning formulation (Problem 1), the targets' kinematics are assumed to be modeled by the mixture of unknown VFs, $\mathcal{F}$, with unknown mixture weights, $\boldsymbol{\pi}$, where M is also unknown. The DPGP mixture model (2.25) is adopted to describe the VFs adaptively from noisy measurements, $\mathbf{m}_j(k)$, which are only available when targets enter the bounded field-of-view (FOV) of the camera, $\mathcal{S}(k)$. The expected information value of $\mathbf{m}_j(k)$ for improving the accuracy of the ith VF in $\mathcal{F}$ is then accessed by the DPGP information theoretic function derived from the cumulative lower bound of the DPGP-EKLD (see Section 6.3.1), denoted by J_{ij}, for $i = 1, \ldots, M$, and $j = 1, \ldots, N$. The set of information theoretic functions is denoted by $\mathcal{J} \triangleq \bigcup_{i=1}^{M}\bigcup_{j=1}^{N}\{J_{ij}\}$.

The sensor dynamics are modeled by the linear time-invariant difference equation (3.6), with linear constraints on the sensor state, $\mathbf{s} \in \mathbb{R}^q$, and the control vector,

$\mathbf{u} \in \mathbb{R}^r$, such that,

$$\mathbf{s}(k+1) = \mathbf{A}\mathbf{s}(k) + \mathbf{B}\mathbf{u}(k), \quad \mathbf{b}_1 \leqslant \mathbf{s} \leqslant \mathbf{b}_2, \quad -\mathbf{1}_r \leqslant \mathbf{u} \leqslant \mathbf{1}_r \tag{6.7}$$

where $\mathbf{A} \in \mathbb{R}^{q\times q}, \mathbf{B} \in \mathbb{R}^{q\times r}, \mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^{q\times 1}$ depend on the sensor dynamics in consideration, and the physical scaling parameters of $\mathbf{u}$ are absorbed into $\mathbf{B}$. It can be seen from (6.7) that $\mathcal{A} = \{\mathbf{s} \in \mathbb{R}^q \mid \mathbf{b}_1 \leqslant \mathbf{s} \leqslant \mathbf{b}_2\}$, and $\mathcal{U} = \{\mathbf{u} \in \mathbb{R}^r \mid -\mathbf{1}_r \leqslant \mathbf{u} \leqslant \mathbf{1}_r\}$. Examples of values of matrices $\mathbf{A}$, $\mathbf{B}$, and the constraints $\mathbf{b}_1$, $\mathbf{b}_2$ are presented in Section 7.3 for a real-world application involving a pan-tilt camera.

From the above problem formulation and assumptions, the optimal control strategy can be obtained by solving a multi-objective optimization (MOO) problem at every time step, where the set of DPGP information theoretic functions, $\mathcal{J}$, are optimized simultaneously, such that camera obtains the most informative measurements for learning the target kinematics, $\{\mathcal{F}, \boldsymbol{\pi}\}$ [161]. Assuming $K = k' - k$ denotes the length of the control horizon, the MOO problem to be solved at time step k can be stated as follows,

$$\begin{aligned}
&\underset{\mathbf{u}(\ell),\ k\leqslant\ell\leqslant k'}{\text{maximize}} \ \begin{bmatrix} J_{11} & \cdots & J_{MN} \end{bmatrix}^T \\
&\text{subject to} \ \ \mathbf{s}(k) = \mathbf{s}_0 \\
&\qquad\qquad \mathbf{s}(\ell+1) = \mathbf{A}\mathbf{s}(\ell) + \mathbf{B}\mathbf{u}(\ell),\ \ell = k, \ldots, k' \\
&\qquad\qquad \mathbf{b}_1 \leqslant \mathbf{s}(\ell) \leqslant \mathbf{b}_2, \quad \ell = k, \ldots, k' \\
&\qquad\qquad -\mathbf{1}_r \leqslant \mathbf{u}(\ell) \leqslant \mathbf{1}_r,\ \ell = k, \ldots, k'
\end{aligned} \tag{6.8}$$

where $\mathbf{s}_0$ is the state of the camera at time step k.

A lexicographic method solution to the MOO problem (6.8) is presented in the following sections by assigning relative importance to the information theoretic functions and by exploiting the geometry properties of the camera FOV. The lexicographic method belongs to the methods with *a priori* articulation of preference,

which assume that the objectives, J_{ij}, can be ranked in order of importance [161]. The lexicographic method is claimed to be the most suitable solution to (6.8) for the following reasons. First, compared to other methods also with *a priori* articulation of preference, such as weighted global criterion methods, lexicographic methods avoid the unfavorable local optima [162]. In addition, compared to methods with *a posteriori* articulation of preference, lexicographic methods are computationally more efficient [163]. Furthermore, unlike methods without articulation of preference, such as compromise solutions, lexicographic methods do not require the definition of closeness between solutions [164].

6.3.1 Lexicographic Approach to Sensor Planning

Since it is computationally intractable to optimize the DPGP-EKLD, the cumulative lower bound (5.65) proposed in Theorem 11 is optimized to obtain sub-optimal control trajectories. In addition, the camera FOV constraint is treated as a potential function, $P(\mathbf{s}, \mathbf{x}_j)$ [165]. An example of $P(\mathbf{s}, \mathbf{x}_j)$ for an omnidirectional sensor FOV is shown in Fig. 6.4. One type of widely used potential function is the quadratic function, with a shape parameter $\sigma_g > 0$, such that,

$$P(\mathbf{s}, \mathbf{x}_j) \triangleq 1 - \|\mathbf{G}\mathbf{s} - \mathbf{g}(\mathbf{x}_j)\|^2 / \sigma_g^2 \tag{6.9}$$

where $\mathbf{G} \in \mathbb{R}^{d' \times q}$ transforms the sensor state, $\mathbf{s} \in \mathbb{R}^q$ into another coordinate system of dimension d'. If no coordinate transformation is required, $d' = q$ and $\mathbf{G} = \mathbf{I}_{d'}$. The vector function $\mathbf{g}(\cdot) : \mathbb{R}^d \to \mathbb{R}^{d'}$ maps the target state, $\mathbf{x}_j \in \mathbb{R}^d$, to the new coordinate system. The matrix $\mathbf{G}$ and the function $\mathbf{g}$ are adopted to resolve the dimensional mismatch between the sensor state and the target state, if there is any, for the potential function. From the cumulative lower bound (5.65) and the potential function (6.9) the objective function that evaluates the improvement of the ith VF

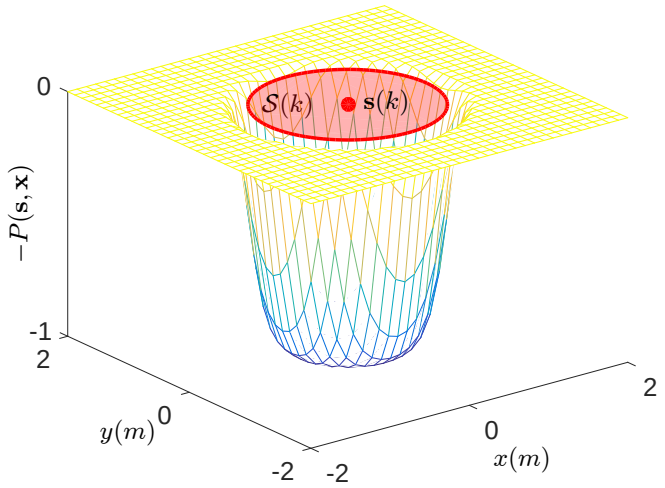

FIGURE 6.4: Example of potential function, $P(\mathbf{s}, \mathbf{x}_j)$.

by measurements of the jth target can be formulated as follows,

$$J_{ij} \triangleq w_{ij} \sum_{\ell=k}^{k'} (1-\gamma)\gamma^{\ell-k} I\big(\boldsymbol{\upsilon}_i; \mathbf{m}_j(\ell)\big) P\big(\mathbf{s}(\ell), \mathbf{x}_j(\ell)\big) \tag{6.10}$$

for $i = 1, \ldots, M$, and $j = 1, \ldots, N$.

In order to describe the lexicographic algorithm, it is assumed that the objective functions in $\mathcal{J}$ can be rearranged in order of decreasing importance (Section 6.3.2) into $\{J'_1, \ldots, J'_{MN}\}$. In other words, $J'_\imath$ is more important than $J'_\jmath$ if $\imath < \jmath$, for $\forall \imath, \jmath \in \{1, \ldots, MN\}$. For brevity, let

$$\boldsymbol{\rho} \triangleq [\mathbf{s}^T(k) \ \cdots \ \mathbf{s}^T(k') \ \ \mathbf{u}^T(k) \ \cdots \ \mathbf{u}^T(k')]^T \tag{6.11}$$

denote the vector consisting of the camera states and control inputs between time steps k and k'. By adopting $\boldsymbol{\rho}$, the constraints on the camera states and control inputs in (6.8) can be expressed as,

$$\mathcal{V} \triangleq \{\boldsymbol{\rho} \in \mathbb{R}^{(q+r)K} \mid \mathbf{C}\boldsymbol{\rho} = \mathbf{d}_1, \ \mathbf{D}\boldsymbol{\rho} \leqslant \mathbf{d}_2\} \tag{6.12}$$

where,

$$\mathbf{C} \triangleq \begin{bmatrix} \mathbf{I}_q & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ -\mathbf{A} & \mathbf{I}_q & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{B} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & -\mathbf{A} & \mathbf{I}_q & \ddots & \vdots & \mathbf{0} & \mathbf{B} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \mathbf{0} & \vdots & \ddots & \ddots & \mathbf{0} \\ \underbrace{\mathbf{0}}_{qK} & \cdots & \mathbf{0} & -\mathbf{A} & \mathbf{I}_q & \underbrace{\mathbf{0}}_{rK} & \cdots & \mathbf{0} & \mathbf{B} \end{bmatrix} \tag{6.13}$$

$$\mathbf{D} \triangleq \begin{bmatrix} -\mathbf{I}_{qK} & \mathbf{I}_{qK} & \mathbf{0}_{qK\times rK} & \mathbf{0}_{qK\times rK} \\ \mathbf{0}_{rK\times qK} & \mathbf{0}_{rK\times qK} & -\mathbf{I}_{rK} & \mathbf{I}_{rK} \end{bmatrix}^T \tag{6.14}$$

$$\mathbf{d}_1 \triangleq \begin{bmatrix} \mathbf{s}_0^T & \mathbf{0}_{1\times q(K-1)} \end{bmatrix}^T \tag{6.15}$$

$$\mathbf{d}_2 \triangleq \begin{bmatrix} \underbrace{\mathbf{b}_1^T \ \cdots \ \mathbf{b}_1^T}_{qK} & \underbrace{\mathbf{b}_2^T \ \cdots \ \mathbf{b}_2^T}_{rK} & \mathbf{1}_{1\times qK} \end{bmatrix}^T \tag{6.16}$$

and $\mathbf{0}_{m\times n}$ denotes a $m \times n$ matrix of zeros. Then, the lexicographic method obtains the solution to (6.8) by solving a sequence of single-objective optimization problems,

$$\begin{aligned} \max_{\boldsymbol{\rho}} \quad & J_\imath'(\boldsymbol{\rho}) \\ \text{s. t.} \quad & J_\jmath'(\boldsymbol{\rho}) \geqslant (J_\jmath')^*, \ \jmath = 1, \ldots, \imath - 1 \\ & \boldsymbol{\rho} \in \mathcal{V} \end{aligned} \tag{6.17}$$

for $\imath = 1, \ldots, MN$, where $(J_\jmath')^*$ is the optimum of the $\jmath$th objective function, found in the $\jmath$th iteration [161].

6.3.2 Order of Importance of Objectives

This section presents the approach for determining the order of importance of the objective functions in $\mathcal{J}$. The notation $J_{ij} \succ J_{i'j'}$ ($J_{ij} \prec J_{i'j'}$) is adopted to denote that J_{ij} is more (less) important than $J_{i'j'}$. First, it is assumed that, for the same target, objective functions corresponding to higher target-VF association weights are more important, such that,

$$J_{ij} \succeq J_{i'j} \Leftrightarrow w_{ij} \geqslant w_{i'j}, \quad 1 \leqslant i \neq i' \leqslant M \tag{6.18}$$

for $j = 1, \ldots, N$. Without loss of generality, the objective functions for the same target can be rearranged in order of decreasing importance according to (6.18), such that,

$$J_{1j} \succeq J_{2j} \succeq \cdots \succeq J_{Mj}, \quad j = 1, \ldots, N \tag{6.19}$$

Second, after the rearrangement of objective functions according to (6.19), objective functions with smaller VF association indices for different targets are assumed to be more important, such that,

$$J_{ij} \succeq J_{i'j'}, \quad 1 \leqslant i < i' \leqslant M,\ 1 \leqslant j \neq j' \leqslant N \tag{6.20}$$

Notice that (6.19) and (6.20) can be combined into,

$$J_{ij} \succeq J_{i'j'}, \quad 1 \leqslant i < i' \leqslant M,\ 1 \leqslant j, j' \leqslant N \tag{6.21}$$

The last assumption copes with the case where $i = i'$. Let J_{ij}^I denote the *ideal value* of the objective function J_{ij}, obtained by optimizing J_{ij} alone, such that [166],

$$J_{ij}^I \triangleq \max_{\boldsymbol{\rho}} \left\{ J_{ij}(\boldsymbol{\rho}) \mid \boldsymbol{\rho} \in \mathcal{V} \right\} \tag{6.22}$$

Then, it is assumed that, when $i = i'$, objective functions corresponding to larger ideal values are more important,

$$J_{ij} \succeq J_{ij'} \Leftrightarrow J_{ij}^I \geqslant J_{ij'}^I, \quad 1 \leqslant j, j' \leqslant N \tag{6.23}$$

for $i = 1, \ldots, M$. In summary, (6.21) and (6.23) together determine the order of importance of the objectives, J_{ij}, completely, as required by the lexicographic method (6.17).

The following two sections present how the sequence of the single-objective optimization problems in (6.17) can be solved after the order of importance of the objective functions, J_{ij}, is determined.

6.3.3 Optimization for the First Iteration

This section presents the solution to the first iteration of the lexicographic method (6.17). Since the mutual information values, $I\big(\boldsymbol{\upsilon}_i; \mathbf{m}_j(\ell)\big)$, $\ell = k, \ldots, k'$, are independent of $\boldsymbol{\rho}$, the single-objective optimization problem in every iteration of (6.17) is equivalent to maximizing a weighted summation of the potential functions, $P(\mathbf{s}, \mathbf{x}_j)$ [102]. Let

$$\beta(\ell) \triangleq w_{ij}(1-\gamma)\gamma^{\ell-k} I\big(\boldsymbol{\upsilon}_i; \mathbf{m}_j(\ell)\big) \tag{6.24}$$

denote the weight multiplied to the ℓth potential function in J_{ij}, for $\ell = k, \ldots, k'$. Then, J_{ij} can be written as a quadratic function of $\boldsymbol{\rho}$,

$$\begin{aligned} J_{ij} &= \sum_{\ell=k}^{k'} \beta(\ell) - \sum_{\ell=k}^{k'} \frac{\beta(\ell)}{\sigma_g^2} \big\| \mathbf{G}\mathbf{s}(\ell) - \mathbf{g}[\mathbf{x}_j(\ell)] \big\|^2 \\ &= \left(\sum_{\ell=k}^{k'} \beta(\ell) - \mathbf{c}^T\mathbf{c} \right) - \left(\boldsymbol{\rho}^T \mathbf{Q}^T \mathbf{Q} \boldsymbol{\rho} - \mathbf{c}^T \mathbf{Q} \boldsymbol{\rho} \right) \end{aligned} \tag{6.25}$$

where,

$$\mathbf{Q} \triangleq \left[\begin{array}{cccc|c} \sqrt{\frac{\beta(k)}{\sigma_g^2}}\mathbf{G} & \mathbf{0} & \cdots & \mathbf{0} & \\ \mathbf{0} & \sqrt{\frac{\beta(k+1)}{\sigma_g^2}}\mathbf{G} & \ddots & \vdots & \\ \vdots & \ddots & \ddots & \mathbf{0} & \\ \mathbf{0} & \cdots & \mathbf{0} & \sqrt{\frac{\beta(k')}{\sigma_g^2}}\mathbf{G} & \end{array} \right| \mathbf{0}_{rK \times rK} \Bigg] \tag{6.26}$$

$$\mathbf{c} \triangleq \left[\sqrt{\frac{\beta(k)}{\sigma_g^2}} \mathbf{g}^T[\mathbf{x}_j(k)] \quad \cdots \quad \sqrt{\frac{\beta(k')}{\sigma_g^2}} \mathbf{g}^T[\mathbf{x}_j(k')] \right]^T \tag{6.27}$$

In the first iteration of the lexicographic method (6.17), the camera states and control inputs are only subject to linear constraints, $\boldsymbol{\rho} \in \mathcal{V}$ (6.12). Therefore, the first iteration of the lexicographic method (6.17) can be solved by convex quadratic programming algorithms, such as the interior-point algorithm, in polynomial time [167].

6.3.4 Optimization for the Remaining Iterations

The remaining iterations in (6.17) require solving the same quadratic programming problem (6.25)-(6.27), but are subject to additional constraints. For the ($\imath > 1$)th iteration, the additional constraints are,

$$J'_\jmath(\boldsymbol{\rho}) \geqslant (J'_\jmath)^*, \quad \jmath = 1, \ldots, \imath - 1 \tag{6.28}$$

The additional constraints in (6.28) are nonlinear, however, they can be simplified to linear constraints that are only slightly more constricted.

Consider the $\jmath$th additional constraint in (6.28) for example, that is, $J'_\jmath(\boldsymbol{\rho}) \geqslant (J'_\jmath)^*$. Recalling the measurement model (3.7), the camera obtains the same measurements, independent of $\boldsymbol{\rho}$, for targets in the camera FOV. In addition, the cumulative lower bound (5.65) is also independent of $\boldsymbol{\rho}$. Let $\mathcal{S}^*(\ell)$, $\ell = k, \ldots, k'$ denote the optimal camera FOV trajectory determined in the $\jmath$th iteration. Then, the $\jmath$th additional constraint in (6.28) can be relaxed without decreasing the cumulative lower bound (5.65) to the set of the following K geometric constraints,

$$\begin{cases} \mathbf{x}_j(\ell) \in \mathcal{S}(\ell), & \text{if } \mathbf{x}_j(\ell) \in \mathcal{S}^*(\ell) \\ \text{no constraint}, & \text{if } \mathbf{x}_j(\ell) \notin \mathcal{S}^*(\ell) \end{cases} \tag{6.29}$$

for $\ell = k, \ldots, k'$. Examples of geometric constraints (6.29) and their properties can be found in Chapter 7.3 for a real-world application involving a pan-tilt camera. The general lexicographic method (6.17) to the MOO formulation (6.8) of the sensor planning problem can be summarized by Algorithm 9.

6.4 Chapter Conclusion

Various scenarios where the target kinematics models developed in Chapter 4 and the information values developed in Chapter 5 can be utilized are discussed and

Algorithm 9 Lexicographic Algorithm

Input: Predicted Gaussian mixture model parameters, $\{\hat{w}_{ij}, \hat{P}_{ij}(\ell)\}_{i=1}^{M}$, for $j = 1, \ldots, N$, $\ell = k, \ldots, k'$; Current sensor state, $\mathbf{s}(k)$; Sensor dynamics parameters, $\{\mathbf{C}, \mathbf{D}, \mathbf{d}_1, \mathbf{d}_2\}$.

Output: Optimal sensor states and control inputs for K future steps, $\boldsymbol{\rho}^*$.

for $j = 1, \ldots, N$ **do**
  Sort $\{w_{ij}\}$ such that $w_{1j} \geqslant w_{2j} \geqslant \cdots \geqslant w_{Mj}$
  Rearrange $\mathcal{J}$ according to sorted $\{w_{ij}\}$
end for
$\mathcal{V} \leftarrow \{\boldsymbol{\rho} \in \mathbb{R}^{(q+r)K} \mid \mathbf{C}\boldsymbol{\rho} = \mathbf{d}_1,\ \mathbf{D}\boldsymbol{\rho} \leqslant \mathbf{d}_2\}$
for $i = 1, \ldots, M$ **do**
  Sort $\{J_{ij}^I\}$, such that $J_{i1}^I \geqslant J_{i2}^I \geqslant \cdots \geqslant J_{iN}^I$
  Rearrange $\{J_{ij}\}$ according to sorted $\{J_{ij}^I\}$
  for $j = 1, \ldots, N$ **do**
    $\boldsymbol{\rho}^* \leftarrow \arg\max_{\boldsymbol{\rho}} \{J_{ij}(\boldsymbol{\rho}) \mid \boldsymbol{\rho} \in \mathcal{V}\}$
    for $\ell = k, \ldots, k'$ **do**
      if $\mathbf{x}_j(\ell) \in \mathcal{S}[\mathbf{s}^*(\ell)]$ **then**
        $\mathcal{V} \leftarrow \mathcal{V} \bigcap \{\boldsymbol{\rho} \in \mathbb{R}^{(q+r)K} \mid \mathbf{x}_j(\ell) \in \mathcal{S}[\mathbf{s}^*(\ell)]\}$
      end if
    end for
  end for
end for

corresponding optimal sensor planning algorithms are developed. The algorithms developed can be treated as the 'Optimal Planner' block in the diagram (Fig. 3.2) to the sensor planning problem (Problem 1). First, when the target kinematics can be described by a time-invariant velocity field modeled by a single GP, a novel greedy algorithm (GP-EKLD-Greedy) is developed and is presented as Algorithm 7 in Section 6.1. For the scenario involving multiple mobile targets and unconstrained sensor dynamics in Section 6.2, the DPGP-EKLD Rectangle Translation algorithm is developed in Algorithm 8. For the scenario where sensor dynamics are constrained in Section 6.3, the lexicographic algorithm is developed as Algorithm 9. Efficiency of the proposed algorithms are demonstrated by the applications and results in the next section with synthetic and real data sets.

7

Applications, Results, and Extensions

In order to evaluate the efficiency of the novel sensor planning algorithms developed in Chapter 6 to the sensor planning problem, synthetic simulations and physical experiments are considered for three real-world applications that obey the formulation and assumptions in Problem 1. Comparisons are made between the proposed sensor planning algorithms (Algorithms 7, 8, 9) and other applicable algorithms available in the literature on GP and DPGP target kinematics modeling, based on entropy, mutual information algorithm, and tracking. Related topics and extensions of the proposed algorithms are discussed in section 7.4.

7.1 Application of Scenario 1

The effectiveness of the new greedy algorithm (Algorithm 7) proposed for Scenario 1, characterized by time-invariant target kinematics and discrete control space (Section 6.1), a real-world application is considered, in which the sensor measurement sequence is to be decided for the purpose of monitoring a time-invariant velocity field, $\mathbf{f} : \mathcal{W} \rightarrow \mathbb{R}^2$, in a two-dimensional workspace, $\mathcal{W} \subset \mathbb{R}^2$. The VF maps the longitude-latitude coordinates to the velocity of the ocean current, and adopts the

data of the monthly mean ocean surface currents of the Pacific Ocean centered on June 5th, 2016 in the region to the west coast of the North America Continent. The corresponding ocean current data (latitude and longitude resolutions: 1/3 Decimal degree), are provided by [168], and are illustrated in Fig. 7.1. The state of the sensor, $\mathbf{s}$, is two-dimensional and consists of the longitude-latitude coordinates. The accessible set of the sensor state, $\mathcal{A}$, consists of 100 randomly selected longitude-latitude coordinates, which are utilized to simulate the positions of the moored buoys that are deployed to measure the ocean current. The collocation points, $\{\boldsymbol{\xi}_l\}_{l=1}^{L}$, defined in (5.12), are populated on a regular grid in the workspace also shown in Fig. 7.1. At every accessible sensor position, the sensor is able to take a noisy measurement of the ocean current, $\mathbf{z}$, according to the measurement model (6.1). The standard deviation of the measurement noise is assumed to be 10% of the maximum ocean current velocity, such that $\sigma_v = 0.1(\text{m/s})$. The covariance function of the GP is assumed to be the squared-exponential covariance function (2.14), and the mean function is assumed to be zero. The set of hyper-parameters of the GP is $\Theta = \{\sigma_f, \mathbf{\Lambda}, \sigma_n\}$, as defined in (2.14), and are optimized by the technique in Section 2.1.2.

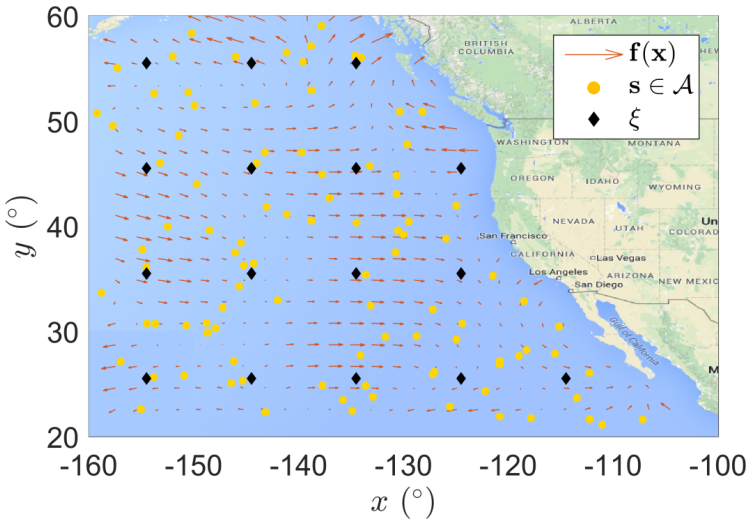

FIGURE 7.1: Example of workspace for Scenario 1 with experimental data of monthly mean ocean surface currents centered on June 5th, 2016[1].

Two sensor planning algorithms are studied for the comparison of performance. The first algorithm is a random algorithm that selects the sensor positions uniformly at random for every time step. The second algorithm is a greedy algorithm that selects the sensor position at the highest entropy,

$$\mathbf{s}^*(k+1) = \arg\max_{\mathbf{s}\in\mathcal{A}} H\big(\mathbf{f}(\mathbf{s}) \mid \mathcal{M}(k)\big) \tag{7.1}$$

where $H(\cdot)$ is the differential entropy defined in (5.1). Due to the optimization of entropy, the algorithm (7.1) is referred to as the 'entropy' algorithm.

To evaluate the performance of the aforementioned sensor planning algorithms, the prediction error, $\epsilon(k)$, is adopted, and is defined as the root mean square error (RMSE) as follows,

$$\epsilon(k) = \sqrt{\frac{1}{L}\sum_{l=1}^{L} \|\mathbf{v}_l - \hat{\mathbf{v}}_l(k)\|_2^2} \tag{7.2}$$

where $\mathbf{v}_l \triangleq \mathbf{f}(\boldsymbol{\xi}_l)$, and $\hat{\mathbf{v}}_l(k)$ is the prediction of $\mathbf{v}_l$ by the GP regression technique (Section 2.1.1) at the kth time step. The prediction errors, $\epsilon(k)$, obtained by the GP-EKLD algorithm (Algorithm 7) and the two comparing algorithms are plotted in Fig. 7.2. As seen in the plot, at the beginning of the simulation, the initial estimations of the ocean currents vary greatly with the actual temperatures. Both the GP-EKLD algorithm and the comparing methods result in decreasing $\epsilon(k)$. However, the GP-EKLD algorithm outperforms the comparing algorithms in that it leads to the fastest and, overall, greatest decrease in $\epsilon(k)$. The simulations for this scenario, although simple, exhibit the effectiveness of the information function based on KL divergence in estimating a velocity field, such as the ocean currents over an area. To illustrate the performance of the GP-EKLD visually, the prediction of the ocean currents and the prediction errors are plotted in Fig. 7.3 and Fig. 7.4, respectively.

[1] ESR. 2009. OSCAR third degree resolution ocean surface currents. Ver. 1. PO.DAAC, CA, USA. Dataset accessed [2016-06-01].

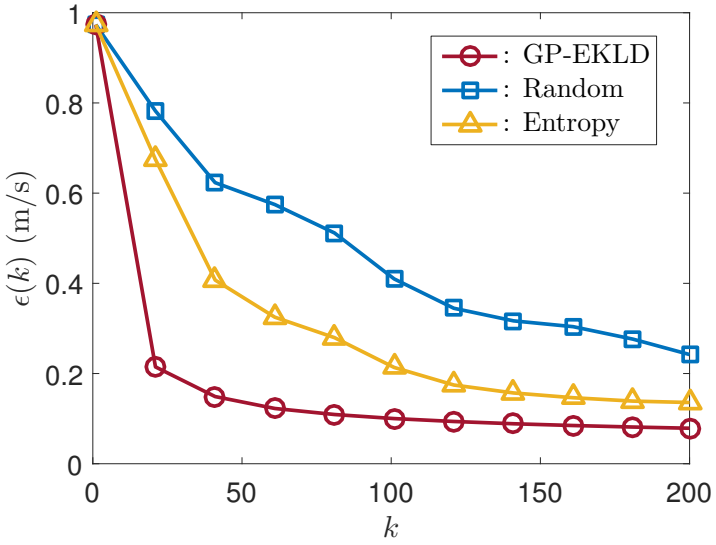

FIGURE 7.2: Prediction error, $\epsilon(k)$, by the GP-EKLD algorithm (Algorithm 7), by the random algorithm, and by the entropy algorithm in (7.1) for the scenario in Fig. 7.1.

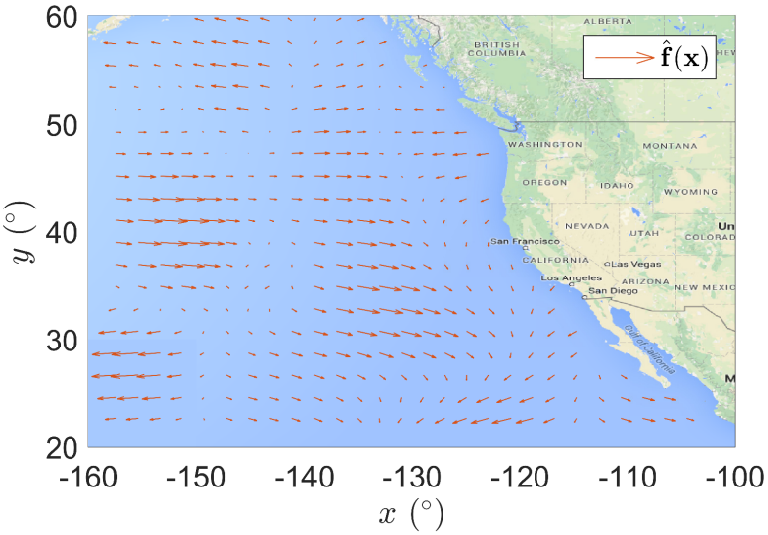

FIGURE 7.3: Prediction of the ocean currents by the GP-EKLD algorithm (Algorithm 7).

7.2 Application of Scenario 2

This section presents the application results obtained for the problem of controlling an indoor surveillance camera with a bounded field-of-view (FOV) under the

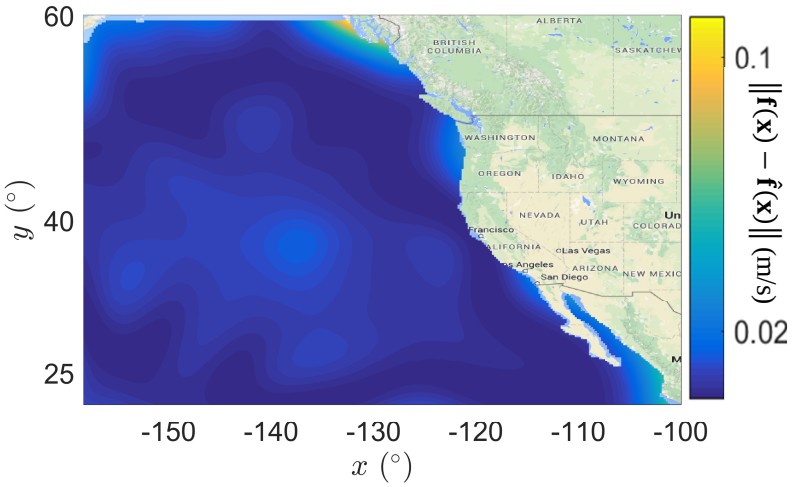

FIGURE 7.4: Prediction error of the ocean currents by the GP-EKLD algorithm (Algorithm 7).

assumptions described in Section 6.2, and as illustrated in Fig. 7.5. The camera (sensor) is utilized to learn the kinematics equations of N unknown and independent mobile targets in a two-dimensional workspace $\mathcal{W}$, described by the set of ordinary differential equations (3.3).

The sensor has two possible FOV zoom levels, $\mathcal{L} = \{1, 2\}$, where the first zoom level enables measurements from a smaller FOV, but with better resolution (less noise), and the second zoom level enables measurements from a larger FOV but lower resolution (more noise), that is $\sigma_{x_1} < \sigma_{x_2}$ and $\sigma_{v_1} < \sigma_{v_2}$. The sensor FOV is assumed to translate in $\mathcal{W}$ without rotation or constraints, i.e. as a free-flying rectangle. Therefore, the sensor state consists of the xy-coordinates of the center of the FOV and the zoom level, $\iota \in \mathcal{L}$, such that,

$$\mathbf{s}(k) \triangleq [x(k) \quad y(k) \quad \iota(k)]^T \tag{7.3}$$

Three algorithms are considered for comparison to demonstrate the efficiency of the DPGP-EKLD-Greedy algorithm (Algorithm 8) presented in Section 6.2. The first algorithm is a tracking algorithm that maximizes the expected entropy reduction of the target position distribution, which is equivalent to the mutual information (MI)

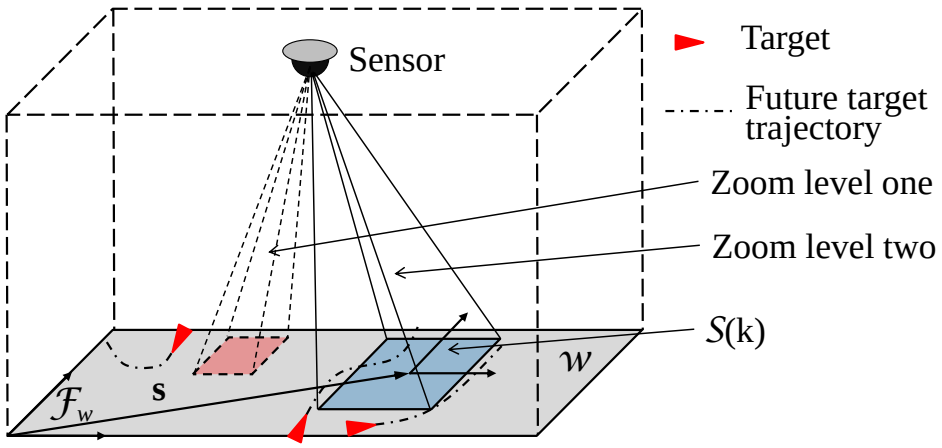

FIGURE 7.5: Illustration of the sensor planning problem in Scenario 2 with a camera and two zoom levels.

of target position estimation and the future measurement. Therefore, the second algorithm is labelled by 'MI', and is described in detail in Appendix A.5. The second algorithm heuristically determines the position of the sensor FOV at the next time step by tracking the nearest target that is not observed at the current time step, and its result is labeled as 'Heuristic' [77]. The last algorithm randomly chooses the FOV position and its result is referred to as 'Random' [83]. The DPGP-EKLD-Greedy algorithm is labeled by 'DPGP-EKLD'.

Algorithm performance is evaluated by the RMSE of velocity between the learned DPGP-MM and the real underlying velocity fields, denoted by $\epsilon(k)$. To obtain the RMS error of velocity, $N_A = 500$ test trajectories (independent from those observed by the camera), $\{\mathcal{T}_j\}_{j=1}^{N_A}$, are generated according to the motion patterns, $\mathcal{F}$ and $\boldsymbol{\pi}$. $\mathcal{T}_j = \{\mathbf{x}_j(k), \mathbf{v}_j(k)\}_{k=1}^{T_j}$, represents the jth new trajectory and T_j is the length of the jth trajectory. These trajectories are compared to the evolving DPGP-MM. If $\hat{\mathbf{v}}_j(k)$ is utilized to denote the predicted velocity at $\mathbf{x}_j(k)$ by the ith Gaussian process component in the DPGP-MM, $\epsilon(k)$ can be obtained as follows,

$$\epsilon(k) = \frac{1}{N_A} \sum_{j=1}^{N_A} \sum_{i=1}^{M} w_{ij} \sqrt{\frac{1}{T_j} \sum_{k=1}^{T_j} \|\mathbf{v}_j(k) - \hat{\mathbf{v}}_j(k)\|_2^2} \tag{7.4}$$

where M is the estimated number of Gaussian process components in the DPGP-MM and w_{ij} is calculated according to (5.30). The performance is evaluated once the DPGP-MM is updated, in order to generate a trend of algorithm performance against time. Fifty runs of the simulation are conducted in order to obtain statistics of the results. Two types of data are studied: the synthetic simulations are first examined in Section 7.2.1 to demonstrate the properties of the DPGP-EKLD-Greedy algorithm, and the physical experiments are conducted in Section 7.2.2 to further verify the efficiency of the DPGP-EKLD-Greedy algorithm.

7.2.1 Synthetic Simulations

In the synthetical simulations, the workspace is assumed to be a square, such that $\mathcal{W} = [0, 10] \times [0, 10]$ (m^2). In zoom level $\iota = 1$, the sensor has a FOV of size 0.5×0.5 (m^2), and the standard derivation of the measurements are assumed to be $\sigma_x = 0.1$ (m), $\sigma_v = 0.1$ (m/s). In zoom level $\iota = 2$, the sensor FOV is of size 1×1 (m^2), and the standard derivation of the measurements are $\sigma_x = 0.5$ (m) and $\sigma_v = 0.5$ (m/s). The sensor takes measurements at every $\Delta t = 0.3$ second in order to learn the target dynamics. The targets dynamics are described by a set of four velocity fields, $\mathcal{F} = \{\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \mathbf{f}_4\}$, as follows,

$$\begin{aligned}
\mathbf{f}_1(\mathbf{x}_j) &= \big[-\sin\left([0.1 \quad \pi/20]\mathbf{x}_j\right) \quad \sin\left([\pi/12 \quad 1]\mathbf{x}_j + \pi/4\right) \big]^T \\
\mathbf{f}_2(\mathbf{x}_j) &= \big[-\cos\left([0 \quad \pi/8]\mathbf{x}_j\right) \quad \sin\left([\pi/4 \quad 0]\mathbf{x}_j\right) \big]^T \\
\mathbf{f}_3(\mathbf{x}_j) &= \left[-0.5 \quad \sin\left([\pi/4 \quad 0.3]\mathbf{x}_j\right)\right]^T \\
\mathbf{f}_4(\mathbf{x}_j) &= \left[-\cos\left([\pi/8 \quad -0.5]\mathbf{x}_j\right) \quad 1\right]^T
\end{aligned} \tag{7.5}$$

It is assumed that every target chooses one velocity field from the set uniformly at random, such that $\boldsymbol{\pi} = [1/4 \quad 1/4 \quad 1/4 \quad 1/4]^T$. Plots of the velocity fields at regular grids in the workspace are shown in Fig. 7.6 for the purpose of visualization, superimposed with sampled trajectories.

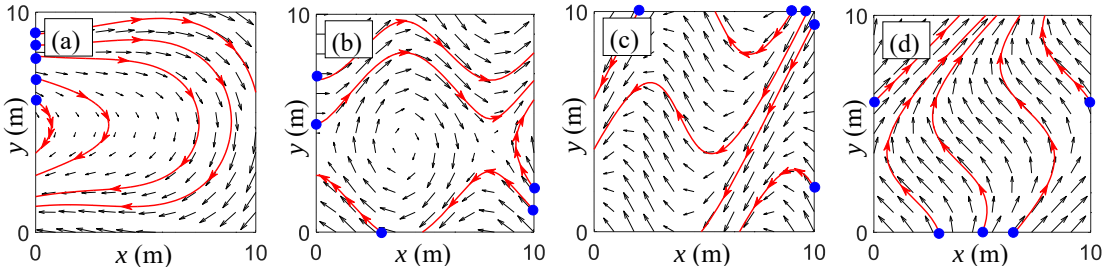

FIGURE 7.6: Visualization of the velocity fields: (a) $\mathbf{f}_1$, (b) $\mathbf{f}_2$, (c) $\mathbf{f}_3$, (d) $\mathbf{f}_4$, by plotting velocity vectors (black arrows) on a regular grid, superimposed with sampled target trajectories (red curves) starting from random initial positions (blue dots).

In order to represent the velocity fields, the parameters for the Gaussian process are selected as follows. Due to the assumption that $\mathbf{f}_i$ is continuously differentiable, the covariance function is chosen to be the squared exponential function (2.14), with hyper-parameters, $\mathbf{\Lambda} = \sqrt{10}\mathbf{I}_2$ (m), and $\sigma_f = 1$ (m/s), based on the size of the workspace. Optimizations of the GP hyper-parameters during the simulations are performed by the gradient-based technique in Section 2.1.2. At the beginning of the simulation, the concentration parameter for the Dirichlet process is assumed to be $\alpha = 1$, and the base distribution is assumed to be the GP with zero mean and covariance function (2.14) with the same hyper-parameters, $\{\mathbf{\Lambda}, \sigma_f\}$. The DPGP-MM is updated once the sensor collects 5 new target trajectories. Since the prior and the likelihood function are conjugate in the DPGP-MM, the DPGP Gibbs Sampler (DPGP-Gibbs in Algorithm 4) is used to sample from the posterior DPGP-MM given measurement histories, where 200 samples are ignored at the beginning, the number of samples is 50, and the sampling interval is set to be every five samples.

Three testing cases with different prior information about the velocity fields are used to examine four strategies. The first case is referred to as 'more informative prior' (MIP), where a large number (15) of training trajectories from the first velocity field and a small number (3 or 4) of training trajectories from the rest velocity fields are utilized for training the prior DPGP-MM. Figure 7.8 shows the training

trajectories in the MIP case and the variance of every Gaussian process at the initial time. It is clear that the prior DPGP-MM provides an estimation of the first velocity field with lower uncertainty. Figure 7.7 shows two snapshots of the simulation in MIP case, which shows that the DPGP-EKLD-Greedy algorithm optimizes the zoom level, ι, automatically. When the groups of samples are close as in Fig. 7.7a, the DPGP-EKLD-Greedy algorithm chooses zoom level $\iota = 2$ to cover all the samples. Although measurement noise at this zoom level is higher, covering most of the samples return higher potential reward. When the distance between the groups of samples is large, the DPGP-EKLD-Greedy algorithm selects zoom level $\iota = 1$ such that the noise is lower.

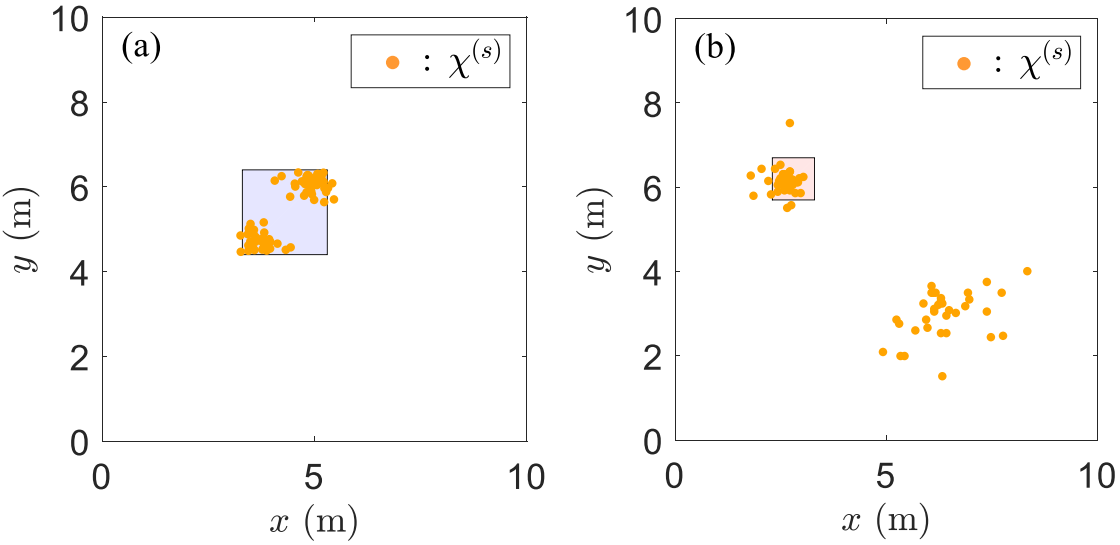

FIGURE 7.7: Simulation snapshots.

The RMS error of velocity obtained by the four algorithms over time is shown in Fig. 7.9. As can be seen, the 'DPGP-EKLD' algorithm outperforms the other algorithms since the error decreases faster and is the lowest at the end of the simulation. In addition, the smaller error bar by the 'DPGP-EKLD' algorithm indicates that its performance is more stable compared to the other methods. The faster decreasing rate of the RMS error by the 'DPGP-EKLD' algorithm can be explained by

the selection of target trajectories. Because the sensor FOV is bounded and there are multiple targets in the workspace, the sensor cannot observe all the targets at the same time. In this case, it is preferable to obtain measurements from targets displaying a motion pattern with higher uncertainty in the current DPGP-MM. Let O_i denote the number of observed target trajectories belonging to the ith velocity field, and $\beta_i = O_i / \sum_{i=1}^{M} O_i$ denote the observed trajectory percentage. Figure 7.10 plots the observed trajectory percentage by the four algorithms, and it is clear that the 'DPGP-EKLD' algorithm is able to obtain more observations from the second to the fourth velocity fields, which are more informative for updating the DPGP-MM in the MIP case. Finally, the results in Fig. 7.11 show that the posterior probabilities of target-VF association defined in (5.30), and obtained from the DPGP-PF (Algorithms 5 and 6), converge to their true values (dash-dotted lines) over time. The snapshots in Fig. 7.11a-c show that, by planning the camera movements via EKLD, the FOV is able to intersect PF clusters with high information value and, thus, rapidly improve their posterior probability distributions.

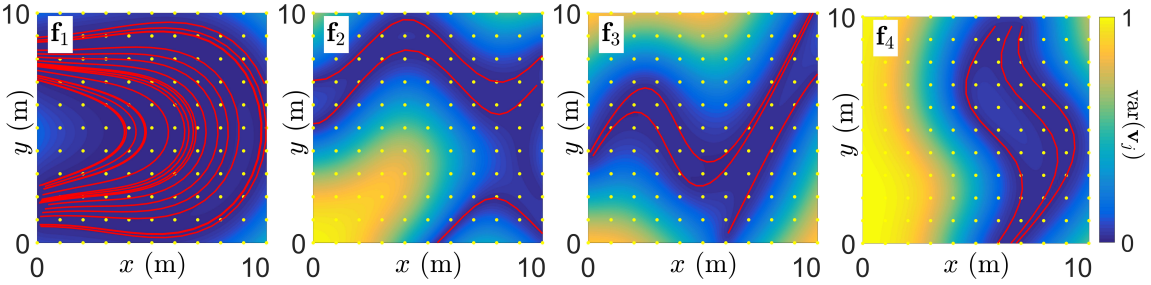

FIGURE 7.8: Training trajectories (red curves) for every velocity field in the MIP case, superimposed with variance of the velocity fields at the initial time and the points of interests (yellow points).

The second case is referred to as 'intermediate informative prior' (IIP), where three trajectories from each velocity field are used to train the prior DPGP-MM. The prior DPGP-MM has an estimation of all the velocity fields with high uncertainty, as shown in Fig. 7.12. Figure 7.14a shows the trend of the RMS errors in

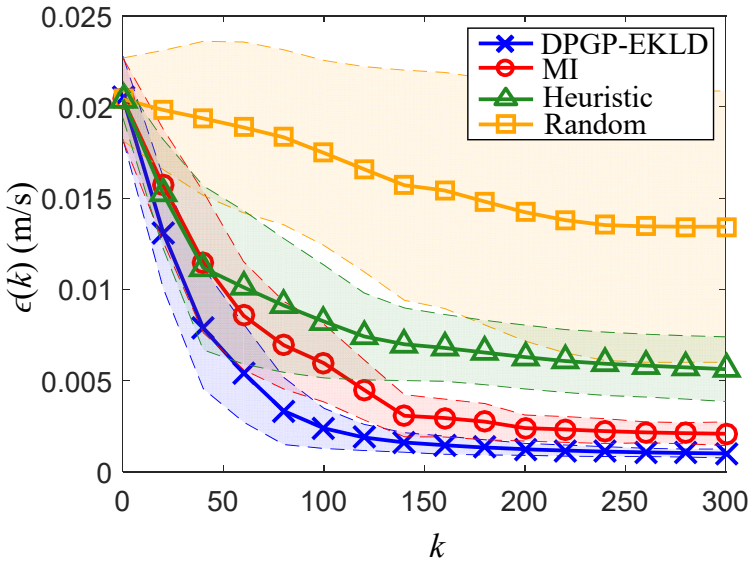

FIGURE 7.9: The mean and variance of RMS error of velocity, ε, obtained by 'DPGP-EKLD' (blue, cross line), by 'MI' (red, circle line), by 'Heuristic' (green, triangle line), and by 'Random' (yellow, square line) algorithms, in the MIP case.

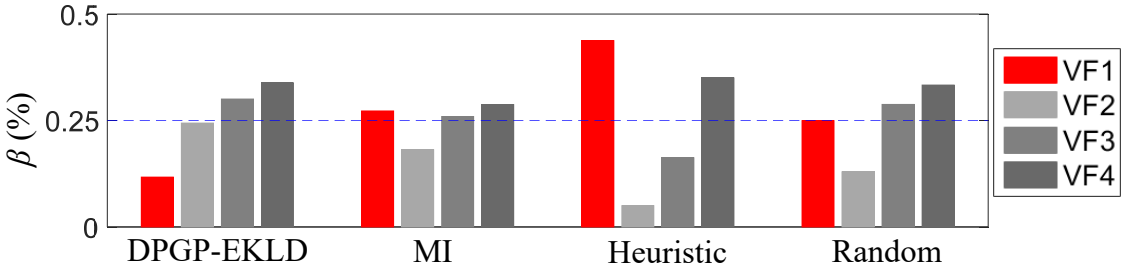

FIGURE 7.10: The distribution of observed trajectory percentage, β, averaged on the 50 runs of simulations, in the MIP case.

the IIP case. Note that the performance of the 'DPGP-EKLD' algorithm and the 'MI' algorithm are close at the beginning of the simulations. This is because IIP contains approximately the same amount of prior knowledge about the four velocity fields, therefore, observing any velocity field is not apparently superior than observing the other velocity fields at the beginning of the simulations. However, when the sensor has collected some information about the target motion patterns, the 'DPGP-EKLD' has the advantage of observing target trajectories from velocity fields with

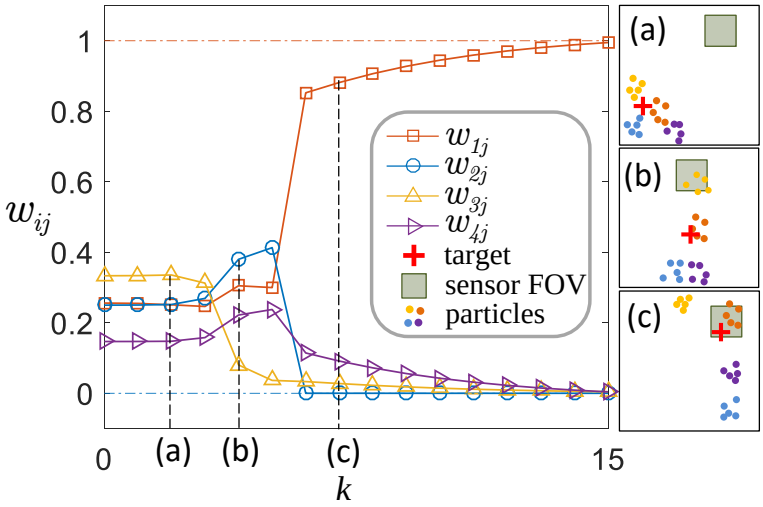

FIGURE 7.11: Time history of target-VF posterior probabilities updated over time by the DPGP-PF (Algorithms 5 and 6).

high uncertainty, which decreases the error at a faster rate. Figure 7.15a shows that 'DPGP-EKLD' algorithm observes approximately the same amount of new trajectories from each velocity field on average, which is preferable since the prior DPGP-MM contains approximately the same amount of information about every velocity field.

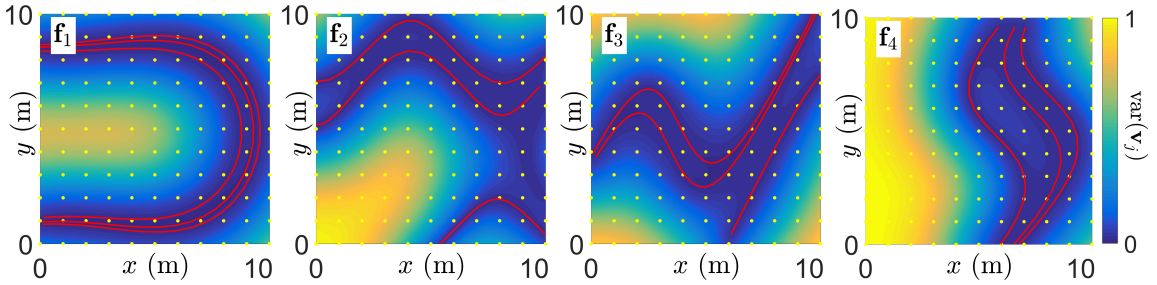

FIGURE 7.12: Training trajectories (red curves) for every velocity field in the IIP case, superimposed with variance of the velocity fields at the initial time and the points of interests (yellow points).

The third case is referred to as 'less informative prior' (LIP), where no training trajectory from the first velocity field is utilized to obtain the prior DPGP-MM. As a result, the trained DPGP-MM has no knowledge of the first velocity and only has an estimation of other three velocity fields with high uncertainty. The training trajec-

tories and variances of velocity fields at the initial time are show in Fig. 7.13. Figure 7.14b shows the trend of the RMS error obtained by the four algorithms in this case. The 'DPGP-EKLD' outperforms other three algorithms since it enables the sensor to observe the target trajectories from the velocity field with higher uncertainty. Figure 7.15b shows that the 'DPGP-EKLD' algorithm is able to obtain more observations of the targets following the first type of velocity field, of which the information is missing in LIP, resulting in a better performance.

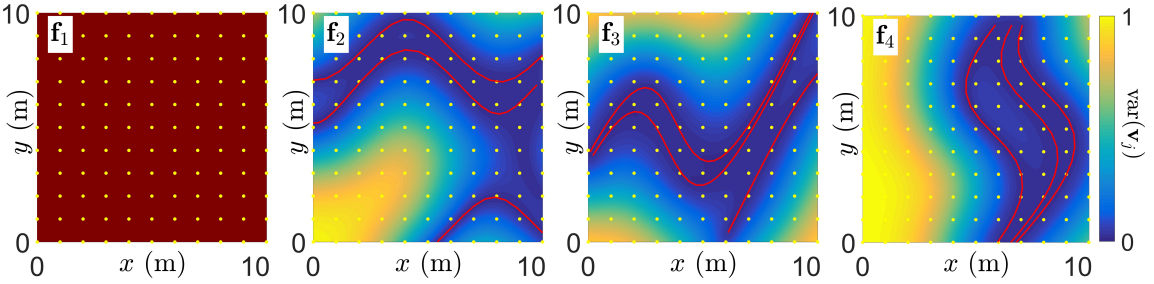

FIGURE 7.13: Training trajectories (red curves) for every velocity field in the LIP case, superimposed with variance of the velocity fields at the initial time and the points of interests (yellow points).

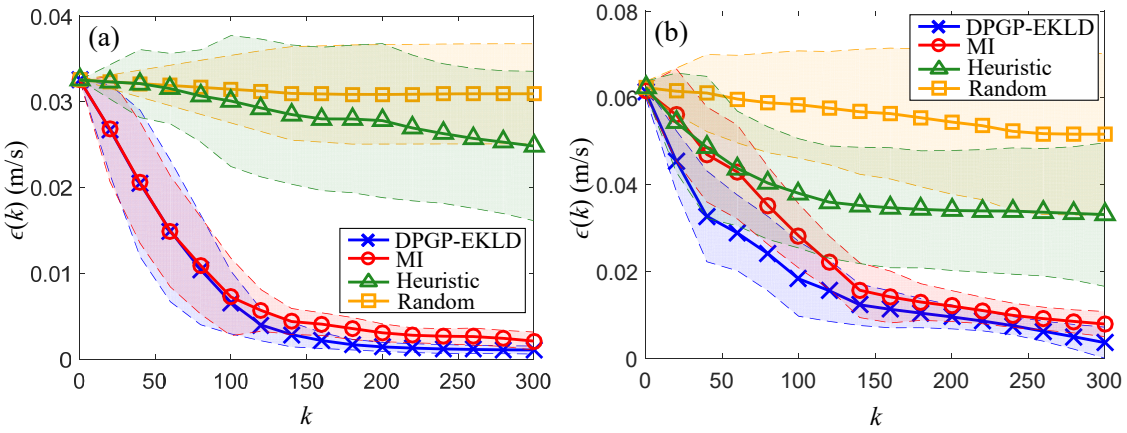

FIGURE 7.14: The mean and variance of RMS error of velocity, ε, obtained by 'DPGP-EKLD' (blue, cross line), by 'MI' (red, circle line), by 'Heuristic' (green, triangle line), and by 'Random' (yellow, square line) algorithms, in the IIP case (left) and the LIP case (right).

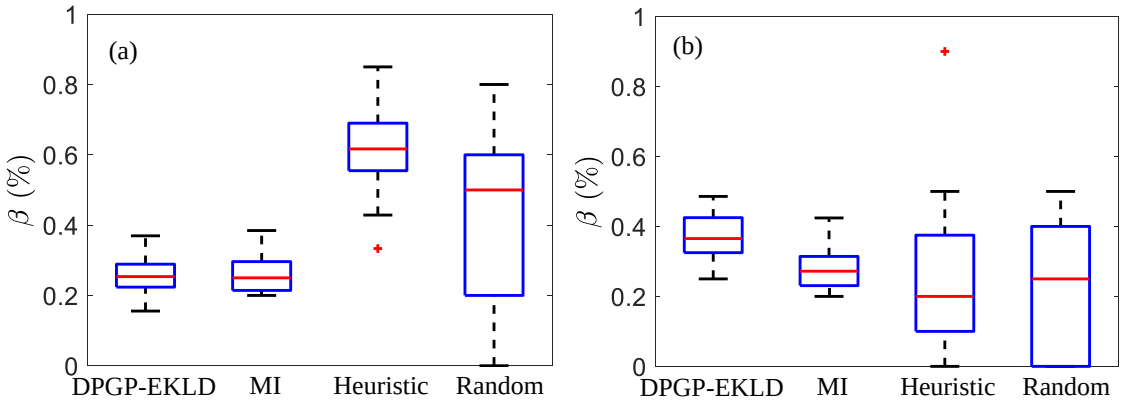

FIGURE 7.15: The percentage of trajectories belonging to the first velocity type observed by the sensor during the simulation in the IIP case (left) and the LIP case (right).

By examining all results from three scenarios, it is clear that for all different priors, the 'DPGP-EKLD' algorithm is more effective at evaluating the expected utility of a future measurement, and thus leads to more informative measurements and more accuracy of target model estimation than 'MI', 'Heuristic', and 'Random' algorithms.

7.2.2 Physical Experiments

The proposed approach was also implemented in hardware using the Real-time indoor Autonomous Vehicle test Environment (RAVEN) at MIT. The domain was constrained to a 16m^2 square region, with two AXIS P5512 PTZ cameras performing target-tracking. Camera intrinsics were utilized to obtain desired square FoVs with correct zoom levels (0.16m^2 and 0.36m^2, respectively) across the domain.

Three iRobot Create ground robots were used as targets, each assigned to one of three underlying velocity fields. A given velocity field may be assigned to multiple targets, and re-assignment was performed upon completion of each vehicle's trajectory (marked by the vehicle departing the domain). Figure 7.16 shows a snapshot of the camera FOV and the corresponding position estimates for the above hardware

setup.

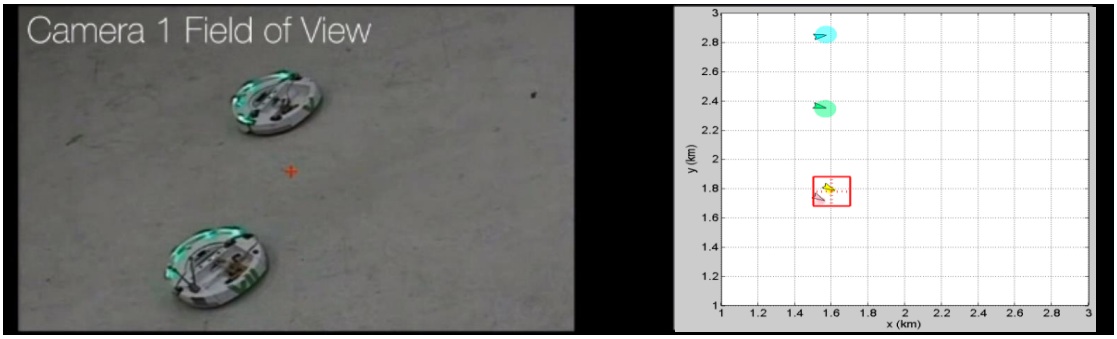

FIGURE 7.16: A snapshot of the moving targets (ground robots) and the optimal camera FOVs (blue squares).

Figure 7.17 illustrates the prediction error $\epsilon(k)$, of the DPGP mixture model in predicting vehicle trajectories using each of the four algorithms described above. As in results from the simulated experiments, this plot illustrates the increasing predictive accuracy of the DPGP using the DPGP-EKLD-Greedy algorithm. Specifically, optimizing the DPGP-EKLD results in fast and significant reduction in model error, owing to more sufficient surveillance of targets exhibiting behaviors with little prior information.

7.3 Application of Scenario 3

The application of Scenario 3 in Section 6.3 considers the problem of developing the optimal control strategy of a pan-tilt (PT) camera for learning the kinematics of N mobile targets traversing a workspace, $\mathcal{W} \subset \mathbb{R}^2$, as illustrated in Fig. 7.18. Recall that the targets' kinematics are modeled by $\mathcal{F}$ and $\boldsymbol{\pi}$ defined in (3.1) and (3.2), respectively, according to the sensor planning problem formulation (Problem 1). In this application, the unknown VFs in $\mathcal{F}$ are assumed to map the target position $\mathbf{x}_j \in \mathcal{W}$ to the target velocity $\mathbf{v}_j$, for $j = 1, \ldots, N$.

The control strategy also needs to account for the camera dynamics, which are

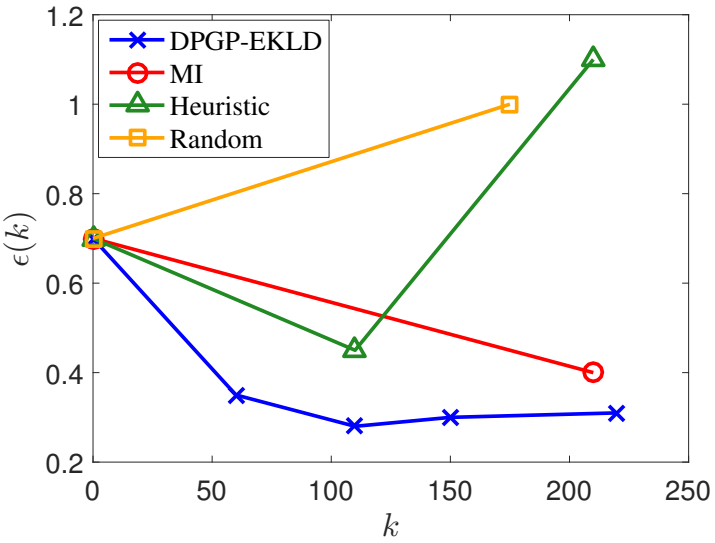

FIGURE 7.17: Prediction error, $\epsilon(k)$, of DPGP mixture models in hardware experiments.

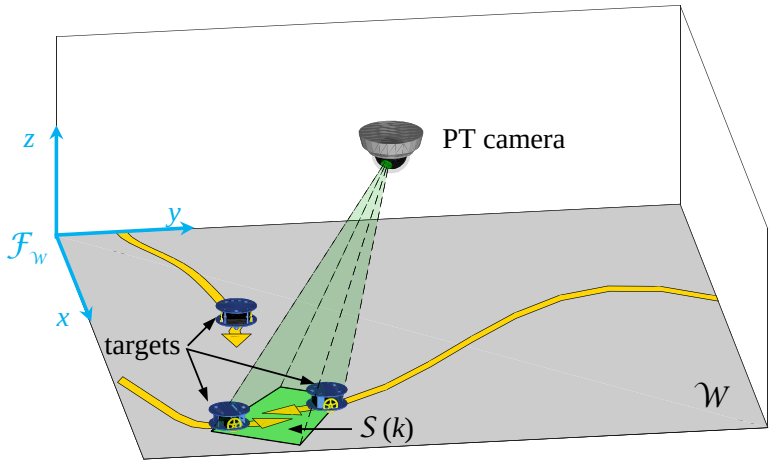

FIGURE 7.18: Illustration of the camera control problem.

modeled by linear time-invariant difference equations, and are taken from [169]. The state of the PT camera, $\mathbf{s}$, consists of the pan angle, $\psi \in [0,\ 2\pi)$, and the tilt angle, $\phi \in [\pi/2,\ \pi]$. The pan and tilt angles represent the ordered set of sequential rotations from the inertial coordinate frame, $\mathcal{F}_{\mathcal{W}}$, to the body fixed coordinate frame, $\mathcal{F}_b$, as illustrated in Fig. 7.19 [170, 149]. Taking also the pan and tilt angular velocities of the camera into consideration, the state of the PT camera can be expressed as $\mathbf{s} = [\psi \quad \phi \quad \dot{\psi} \quad \dot{\phi}]^T$. It is assumed that the control vector, $\mathbf{u} = [u_1 \quad u_2]^T$, consists

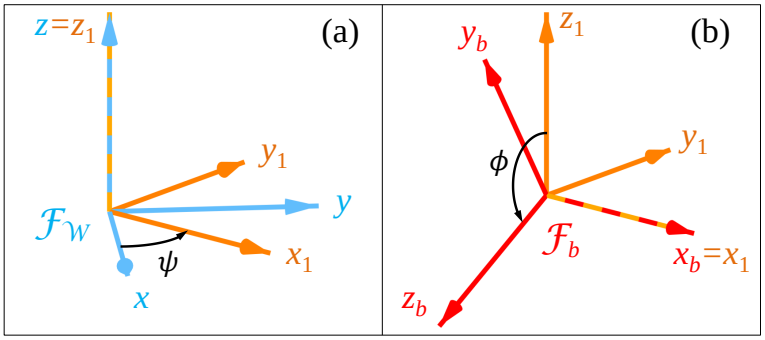

FIGURE 7.19: Sequence of pan-tilt angle rotations. (a) Pan rotation ψ from inertial frame to Intermediate Frame 1. (b) Tilt rotation ϕ from Intermediate Frame 1 to body axes.

of the voltage levels applied to the two motors that independently adjust ψ and ϕ, respectively [171]. Then, the state-space representation of the camera dynamics is,

$$\mathbf{s}(k+1) = \mathbf{A}\mathbf{s}(k) + \mathbf{B}\mathbf{u}(k) \tag{7.6}$$

where,

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{ and } \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ b_1 & 0 \\ 0 & b_2 \end{bmatrix} \tag{7.7}$$

In (7.7), Δt is the interval between discrete times, and b_1, b_2 are the physical parameters of the two motors [8]. Assuming that $\dot{\psi}_m$ and $\dot{\phi}_m$ are the maximum pan and tilt angular velocities of the camera, respectively, the linear constraints on the camera dynamics can be expressed as,

$$\begin{cases} \mathbf{b}_1 \leqslant \mathbf{s} \leqslant \mathbf{b}_2 \\ |\mathbf{u}| \leqslant \mathbf{1}_2 \end{cases}, \quad \text{where} \quad \begin{cases} \mathbf{b}_1 \triangleq \begin{bmatrix} 0 & \pi/2 & -\dot{\psi}_m & -\dot{\phi}_m \end{bmatrix}^T \\ \mathbf{b}_2 \triangleq \begin{bmatrix} 2\pi & \pi & \dot{\psi}_m & \dot{\phi}_m \end{bmatrix}^T \end{cases} \tag{7.8}$$

and $\mathbf{1}_n$ denotes a $n \times 1$ vector of ones. The physical scaling parameters of $\mathbf{u}$ are absorbed into $\mathbf{B}$.

By changing $\mathbf{s}(k)$ according to (7.6)-(7.8), the camera can adjust its FOV, and obtain noisy measurements of the target position and velocity if $\mathbf{x}_j(k) \in \mathcal{S}(k)$. The

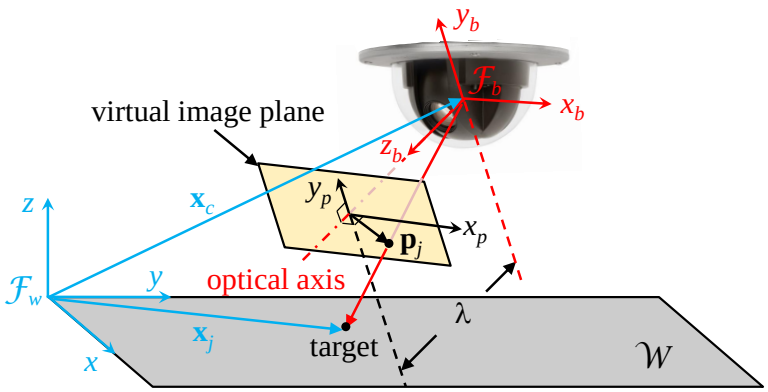

FIGURE 7.20: Pinhole camera model.

measurements are assumed to be characterized by the pinhole camera model, as illustrated in Fig. 7.20 [172]. In this model, the *optical axis* is defined as the line that the camera lens is symmetric about. The *virtual image plane* is the 2D plane perpendicular to the optical axis [170]. The distance between the virtual image plane and the origin of $\mathcal{F}_b$ is the focal length λ. If $\mathbf{p}_j$ is the projection of $\mathbf{x}_j$ in the virtual image plane, the camera measurement model can be expressed as,

$$\mathbf{m}_j(k) = \begin{cases} [\mathbf{p}_j^T(k)\ \dot{\mathbf{p}}_j^T(k)]^T + \mathbf{n}(k), & \mathbf{x}_j(k) \in \mathcal{S}(k) \\ \varnothing, & \mathbf{x}_j(k) \notin \mathcal{S}(k) \end{cases} \tag{7.9}$$

for $j = 1, \ldots, N$, where $\mathbf{n} \in \mathbb{R}^4$ is an additive Gaussian noise vector, with zero mean and known covariance matrix $\text{diag}\left(\sigma_x^2 \mathbf{I}_2, \sigma_v^2 \mathbf{I}_2\right)$. The analytical expressions of $\mathbf{p}_j$ and $\dot{\mathbf{p}}_j$ are given in Appendix A.6.

7.3.1 Lexicographic Approach Revisit

The lexicographic approach (Algorithm 9) presented in Section 6.3 provides a general framework for solving the sensor planning problem with dynamic and FOV constraints. However, Algorithm 9 does not provide details of the algorithm implementation. To this end, the lexicographic approach is revisited, and applied to the PT camera control application subject to the dynamic constraints in (9)-(7.8) and the measurement model in (7.9).

First, matrix $\mathbf{G}$ and the projection function $\mathbf{g}(\cdot)$ in the potential function (6.9) for the PT camera control application are defined as follows. Matrix $\mathbf{G}$ extracts the pan and tilt angles of the sensor state, such that,

$$\mathbf{G} \triangleq \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{7.10}$$

The projection function, $\mathbf{g}(\cdot) : \mathbb{R}^2 \to [0, 2\pi) \times [\pi/2, \pi]$ maps the target position, $\mathbf{x}_j$, to the corresponding pan-tilt angles, denoted by $[\psi_j \quad \phi_j]^T$, such that,

$$\mathbf{g}(\mathbf{x}_j) \triangleq \begin{bmatrix} \psi_j \\ \phi_j \end{bmatrix} = \begin{bmatrix} \tan^{-1}\left[(y_j - y_c)/(x_j - x_c)\right] \\ -\tan^{-1}\left[\sqrt{(y_j - y_c)^2 + (x_j - y_c)^2}/z_c\right] + \pi \end{bmatrix} \tag{7.11}$$

where $\mathbf{x}_c = [x_c \quad y_c \quad z_c]^T$ is the position of the origin of $\mathcal{F}_b$ in $\mathcal{F}_\mathcal{W}$, as illustrated in Fig. 7.20.

From the matrix $\mathbf{G}$ and the projection function $\mathbf{g}(\cdot)$, the analytical form of the geometric constraint in (6.29) can be expressed as follows. Consider the ℓth geometric constraint in (6.29), that is $\mathbf{x}_j(\ell) \in \mathcal{S}(\ell)$, as an example. Consider the case where $\mathbf{x}_j(\ell) \in \mathcal{S}^*(\ell)$. Since the shape of $\mathcal{S}(\ell)$ changes with respect to the camera state, it is easier to derive the analytical form of the geometric constraints in (6.29) by projecting $\mathbf{x}_j(\ell)$ and $\mathcal{S}(\ell)$ into the virtual image plane, as illustrated in Fig. 7.20. Recalling (A.12)-(A.13), the projection of $\mathbf{x}_j(\ell)$ in the virtual image plane, denoted by $\mathbf{p}_j = [p_x \quad p_y]^T$, is obtained by the pinhole camera model. In addition, the projection of $\mathcal{S}(\ell)$ in the virtual image plane is a rectangle with the same size as the image sensor. Let a and b denote the width and height of the image sensor, respectively. It follows that the target lies in the camera FOV in the workspace if and only if $\mathbf{p}_j$ is in the image sensor, that is,

$$\mathbf{x}_j(\ell) \in \mathcal{S}(\ell) \Leftrightarrow \mathbf{p}_j \in [-a/2,\ a/2] \times [-b/2,\ b/2] \tag{7.12}$$

Recall that the pan-tilt coordinates of the target, $[\psi_j \quad \phi_j]^T$, have been obtained by (7.11). Substituting (A.12)-(A.13) in (7.11) yields the relationship between $[\psi \quad \phi]^T$

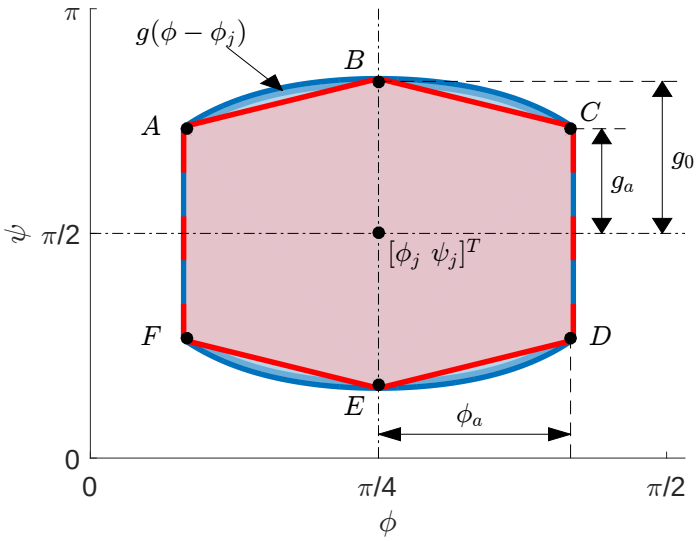

FIGURE 7.21: Example of the nonlinear constraint (7.14) (blue region bounded by arcs $\widehat{ABC}$, $\widehat{DEF}$, and segments $\overline{CD}$, $\overline{FA}$) and the linear approximation (7.17) (ploygon $ABCDEF$).

and $[\psi_j \quad \phi_j]^T$,

$$\begin{cases} \phi - \phi_j = -\tan^{-1}(p_y/\lambda) \\ \psi - \psi_j = -\tan^{-1}\left[p_x \sec(\phi_j)\cos(\phi-\phi_j)/\lambda\right] \end{cases} \tag{7.13}$$

Then, the analytical form of $\mathbf{x}_j(\ell) \in \mathcal{S}(\ell)$ in terms of the pan-tilt coordinates of the camera can be obtained by substituting (7.12) in (7.13),

$$\begin{cases} |\phi - \phi_j| \leqslant \tan^{-1}[b/(2\lambda)] \triangleq \phi_a \\ |\psi - \psi_j| \leqslant \tan^{-1}\left[\dfrac{a}{2\lambda}|\sec(\phi_j)|\cos(\phi-\phi_j)\right] \triangleq g(\phi-\phi_j) \end{cases} \tag{7.14}$$

An example of the constraint (7.14), where $[\psi_j \quad \phi_j]^T = [\pi/2 \quad \pi/4]^T$, is illustrated in Fig. 7.21.

The constraint (7.14) is nonlinear, however, it can be proven to be convex by the following theorem:

Theorem 13. *The constraint* (7.14) *defines a convex set of the camera pan-tilt coordinates,* $[\psi \quad \phi]^T$*, given any camera parameters,* $a, b, \lambda > 0$*, and target pan-tilt coordinates,* $[\psi_j \quad \phi_j]^T$*.*

Proof. As a first step, let us adopt the coordinate transformations $\phi' \triangleq \phi - \phi_j$ and $\psi' \triangleq \psi - \psi_j$, such that (7.14) can be simplified to,

$$|\phi'| \leqslant \phi_a \quad \text{and} \quad |\psi'| \leqslant g(\phi') \tag{7.15}$$

In convex analysis, the epigraph of a function is defined as the set of points lying on or above its graph [173]. Similarly, the hypograph of a function refers to the set of points lying on or below its graph. Since ϕ_a is independent of the variables ψ' and ϕ', (7.15) is equivalent to the intersection of the epigraph of $-g(\phi')$ and the hypograph of $g(\phi')$, for $\phi' \in [-\phi_a,\ \phi_a]$, as illustrated in Fig. 7.21. Let $g_0 \triangleq g(0)$, the second derivative of $g(\phi')$ is,

$$\frac{d^2 g(\phi')}{d\phi'^2} = -\frac{g_0 \cos(\phi')[1 + g_0^2 + g_0^2 \sin^2(\phi')]}{[1 + g_0^2 \cos^2(\phi')]^2} \tag{7.16}$$

Since g_0 is a positive constant, (7.16) is less than or equal to zero for $\phi' \in [-\pi/2,\ \pi/2]$. In addition, since $b, \lambda > 0$, it follows that $0 < \phi_a < \pi/2$. Therefore, $g(\phi')$ and $-g(\phi')$ are concave and convex functions for $\phi' \in [-\phi_a,\ \phi_a]$, respectively. The epigraph (hypograph) of a function is a convex set, if the function is a convex (concave) function [173]. Therefore, the epigraph of $-g(\phi')$ and the hypograph of $g(\phi')$ are both convex sets. Since the intersection of two convex sets is a convex set, the intersection of the epigraph of $-g(\phi')$ and the hypograph of $g(\phi')$ is a convex set, which completes the proof. □

Theorem 13 shows that (7.14) is convex, which retains the polynomial time complexity of the $\imath$th iteration in (6.17), for $\imath = 2, \ldots, MN$. In order to further reduce the computational complexity, a set of linear constraints that are only slightly more

restricted than (7.14) are proposed, as illustrated in Fig. 7.21. Recall that $g_0 \triangleq g(0)$, and let $g_a \triangleq g(\phi_a)$. The linear constraints can be expressed as,

$$\begin{cases} |\phi - \phi_j| \leqslant \phi_a \\ |\psi - \psi_j| \leqslant g_0 \pm \dfrac{g_0 - g_a}{\phi_a}(\phi - \phi_j) \end{cases} \tag{7.17}$$

where ϕ_a and $g(\cdot)$ are defined in (7.14). Notice that ϕ_a is half of the vertical angle of view with respect to focal length λ. Assuming that $\lambda_{\min} > 0$ denotes the minimum focal length, the upper bound of ϕ_a is,

$$\phi_a \leqslant \tan^{-1}[b/(2\lambda_{\min})] \triangleq \phi_u \tag{7.18}$$

Then, the following theorem states that (7.17) is a close approximation of (7.14):

Theorem 14. *The ratio of the area of the convex set defined by the linear constraints* (7.17) *over the area of the convex set defined by the nonlinear constraints* (7.14) *is lower bounded by,*

$$r(\phi_u) \triangleq 1 + \frac{1 - \cos\phi_u}{\cos\phi_u}\frac{\pi - 2\phi_u}{4\phi_u} - \frac{1 - \cos\phi_u}{4\phi_u}\pi \tag{7.19}$$

Proof. In order to keep the proof concise, the coordinate transformations in (7.15) are also adopted here. Since (7.15) and (7.17) are symmetric about both the ϕ' and ψ' axes, the ratio of the area of (7.17) over the area of (7.15) remains the same in the first quadrant, where $\phi' \geqslant 0$ and $\psi' \geqslant 0$, as illustrated in Fig. 7.22.

Let S_1 denote the area of the convex set specified by the nonlinear constraint (7.15) in the first quadrant. Let line l_0 denote the horizontal line that passes through point A, as illustrated in Fig. 7.22. The first derivative of $g(\phi')$ is,

$$dg(\phi')/d\phi' = -g_0 \sin\phi'/(g_0^2 \cos^2\phi' + 1) \leqslant 0, \quad \text{for } \phi' \in [0,\ \pi/2] \tag{7.20}$$

Therefore, $g(\phi')$ is monotonically decreasing for $\phi' \in [0,\ \pi/2]$, and is upper bounded by line l_0. Let line l_1 denote the line that passes through point E and point F in

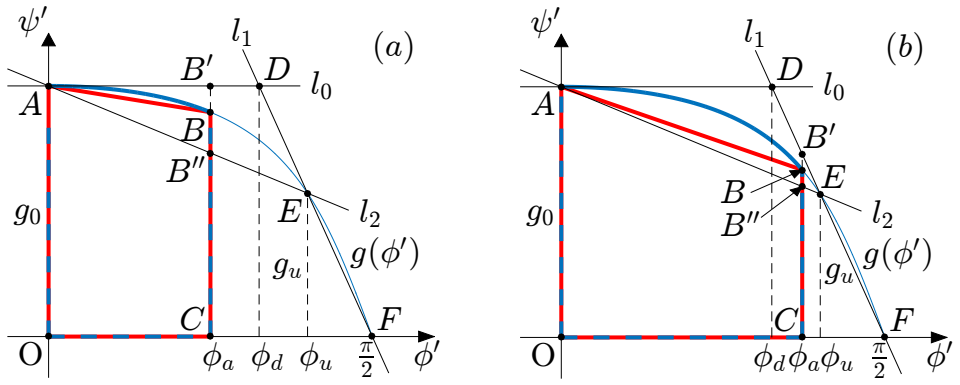

FIGURE 7.22: Linear approximation (7.17) of the nonlinear constraint (7.14) in the first quadrant when (a) $\phi_a \leqslant \phi_d$ and (b) $\phi_a > \phi_d$.

Fig. 7.22. Since $g(\phi')$ is a concave function for $\phi' \in [0,\ \pi/2]$ (see proof of Theorem 13), $g(\phi')$ is also upper bounded by line l_1 for $\phi' \in [0,\ \phi_u]$. Let point D denote the intersection of line l_0 and line l_1, and let ϕ_d denote the ϕ'-axis coordinate of point D. It follows that, when $\phi_a \leqslant \phi_d$, S_1 is upper bounded by the area of polygon $OAB'C$, as illustrated in Fig. 7.22a. When $\phi_a > \phi_d$, S_1 is upper bounded by the area of polygon $OADB'C$. In other words,

$$S_1 \leqslant \begin{cases} g_0\phi_a, & \text{if } \phi_a \leqslant \phi_d \\ g_0\phi_d + \dfrac{1}{2}(g_a + g_0)(\phi_a - \phi_d), & \text{if } \phi_a > \phi_d \end{cases} \tag{7.21}$$

where $g_0 \triangleq g(0)$, and $g_a \triangleq g(\phi_a)$.

Let S_2 denote the area of the convex set specified by the linear constraints (7.17) in the first quadrant. Let line l_2 denote the line passing through point A and point E, as illustrated in Fig. 7.22. Since $g(\phi')$ is a concave function in $[0,\ \pi/2]$, it is lower bounded by line l_2 for $\phi' \in [0,\ \phi_u]$. Therefore, S_2 is greater than the area of polygon $OAB''C$,

$$S_2 \geqslant \frac{(g_a + g_0)}{2}\phi_a = \left(\frac{g_u - g_0}{2\phi_u}\phi_a + g_0\right)\phi_a \tag{7.22}$$

where $g_u \triangleq g(\phi_u)$.

Substituting (7.22) in (7.21) yields that, when $\phi_a \leqslant \phi_d$,

$$\begin{aligned}\frac{S_2}{S_1} &\geqslant 1 - \frac{g_0 - g_u}{2g_0\phi_u}\phi_a \geqslant 1 - \frac{g_0 - g_u}{2g_0\phi_u}\phi_d \\ &= 1 + \frac{1 - \frac{g_u}{g_0}}{\frac{g_u}{g_0}}\frac{\pi - 2\phi_u}{4\phi_u} - \frac{1 - \frac{g_u}{g_0}}{4\phi_u}\pi \\ &\geqslant 1 + \frac{1 - \cos\phi_u}{\cos\phi_u}\frac{\pi - 2\phi_u}{4\phi_u} - \frac{1 - \cos\phi_u}{4\phi_u}\pi\end{aligned} \tag{7.23}$$

where the last inequality comes from that $g_u \geqslant g_0 \cos\phi_u$. In addition, when $\phi_a > \phi_d$, it follows that,

$$\begin{aligned}\frac{S_2}{S_1} &\geqslant \frac{1}{\frac{2g_0\phi_d}{(g_a+g_0)\phi_a} + (1 - \frac{\phi_d}{\phi_a})} \geqslant 1 - \frac{g_0 - g_u}{2g_0\phi_u}\phi_d \\ &\geqslant 1 + \frac{1 - \cos\phi_u}{\cos\phi_u}\frac{\pi - 2\phi_u}{4\phi_u} - \frac{1 - \cos\phi_u}{4\phi_u}\pi\end{aligned} \tag{7.24}$$

where the second inequality is true since $g(\phi')$ is monotonically decreasing for $\phi' \in [0,\ \pi/2]$. The equalities in both (7.23) and (7.24) are achieved when $\phi_a = \phi_d$, and $\lambda \to \infty$. □

Remark 15. *$r(\phi_u)$ is a monotonically decreasing function for $\phi_u \in [0,\ \pi/2]$, as illustrated in Fig. 7.23.*

Remark 16. *For most surveillance cameras, the upper bound of the vertical angle of view is less than 90° [174], which means that the lower bound in Theorem 14 is greater than $r(\pi/4) = 1 - \frac{(\sqrt{2}-1)^2}{2} \approx 91.4\%$.*

From Theorem 14 and Remark 16, it can be seen that the optimization problems in the remaining iterations of the lexicographic method (6.17) can be formulated as quadratic programming problems with objective function (6.25) and linear constraints (7.17). Therefore, they can also be solved by convex quadratic programming algorithms in polynomial time [167]. The performance of the lexicographic approach

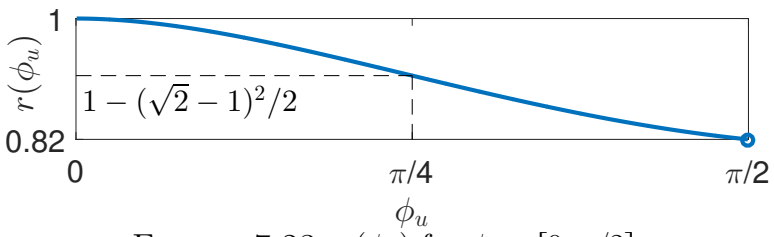

FIGURE 7.23: $r(\phi_u)$ for $\phi_u \in [0, \pi/2]$.

using the simplified linear constraints (7.17) is demonstrated by the simulation and results in the next section.

7.3.2 Simulation and Results

In this section, the cumulative lower bound of the DPGP-EKLD, $\hat{D}_L$, derived in closed form in (5.65), and used to obtain the objective functions (6.10), is first demonstrated for a variety of target kinematics. Then, the efficiency of the lexicographic method (Algorithm 9) is demonstrated by comparing to the optimal solution [175], entropy reduction [176], greedy [102], potential field [145], patrol [177] and random [83] algorithms. Finally, the computation complexities of the lexicographic method and the six comparing algorithms are analysed theoretically and tested by experiments.

The simulations are performed using two experimental datasets collected from pedestrian movements, as shown in Fig. 7.24 [93]. For both of the experimental datasets, 75% of the pedestrians are selected at random as the targets in the simulations, and the remaining pedestrian measurements are used to evaluate the performance of the camera control algorithms. One PT camera is assumed to be located at the center of the workspace. Parameters of the camera are adopted from commercial pan-tilt cameras, such as the AXIS® M5013 Dome Network camera [174]. The parameters of the PT camera are summarized in Table I, and are representative of all simulations conducted with other parameters.

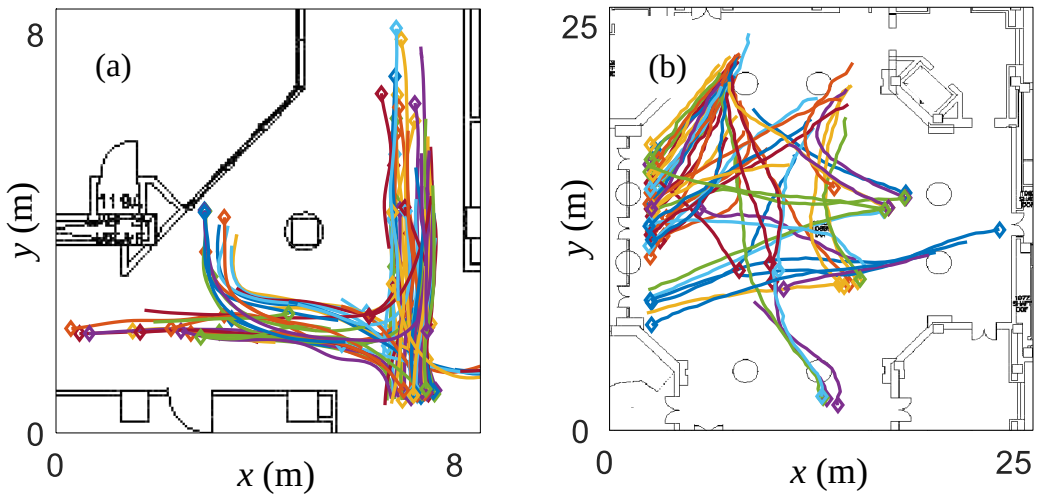

FIGURE 7.24: Two experimental datasets consisting of time-stamped sequences of position and velocity measurements of pedestrians at a frequency of 2Hz. Initial positions are denoted by diamonds. (a) Dataset I: 88 pedestrians measured at the intersection of two corridors. (b) Dataset II: 61 pedestrians measured in a lobby area.

Table 7.1: Parameters of PT Camera

Description	Variable	Value
Horizontal angle of view	g_0	45°
Vertical angle of view	ϕ_u	32°
Maximum pan angular velocity	$\dot{\psi}_m$	100°/s
Maximum tilt angular velocity	$\dot{\phi}_m$	100°/s
Motor coefficients	b_1, b_2	$100°/(\text{Vs}^2)$
Standard deviation of position measurement noise	σ_x	0.1 (m)
Standard deviation of velocity measurement noise	σ_v	0.1 (m/s)

A. Cumulative Lower Bound Simulation Results

The distance between the cumulative lower bound (5.65), and the DPGP-EKLD (5.23) is demonstrated by considering four examples of targets as shown in Fig. 7.25. Both the cumulative lower bound and the DPGP-EKLD are calculated using the squared-exponential covariance function (2.14) with $\mathbf{\Lambda} = \mathbf{I}_2$ and $\sigma_f = 1$. The

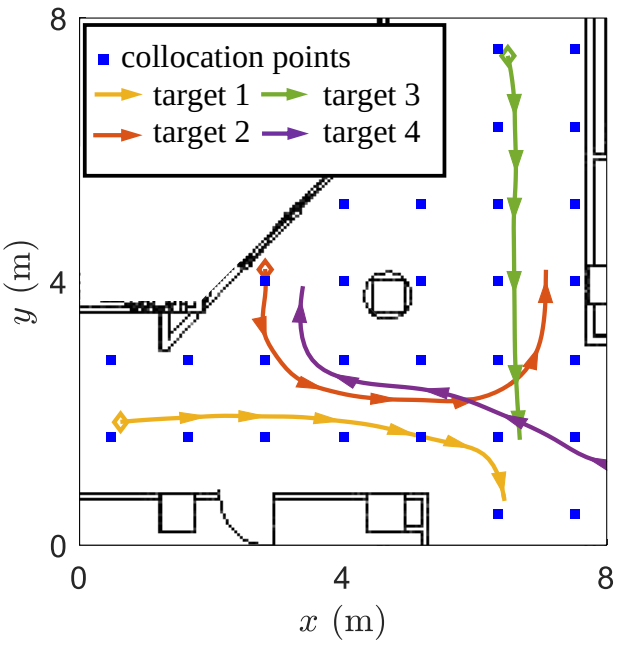

FIGURE 7.25: Four examples of targets from Dataset I in Fig. 7.24a superimposed with the collocation points.

collocation points are selected as the evenly distributed grid points in the workspace. The discount factor in (5.65) is chosen to be $\gamma = 0.9$ for all the simulations. It can be seen from Fig. 7.26 that the cumulative lower bound (5.65) is approximately lower than the DPGP-EKLD (5.23) by a constant distance in the logarithm-scale plots for all the targets in Fig. 7.25. Therefore, the cumulative lower bound approximately equals to the DPGP-EKLD multiplied by a constant factor (close to $1 - \gamma$), which validates the claim that the cumulative lower bound (5.65) can be used in lieu of the DPGP-EKLD (5.23) for deriving the objective functions J_{ij} in (6.10).

B. Camera Control Optimization Results

The DPGP KL divergence, $D\big(\boldsymbol{v}; M(1,k)\big)$, derived in closed form in (5.23), represents the information value of obtained measurements for improving the DPGP-MM [175]. Therefore, the DPGP KL divergence can be used to evaluate the performance of camera control algorithms. The effectiveness of the lexicographic method

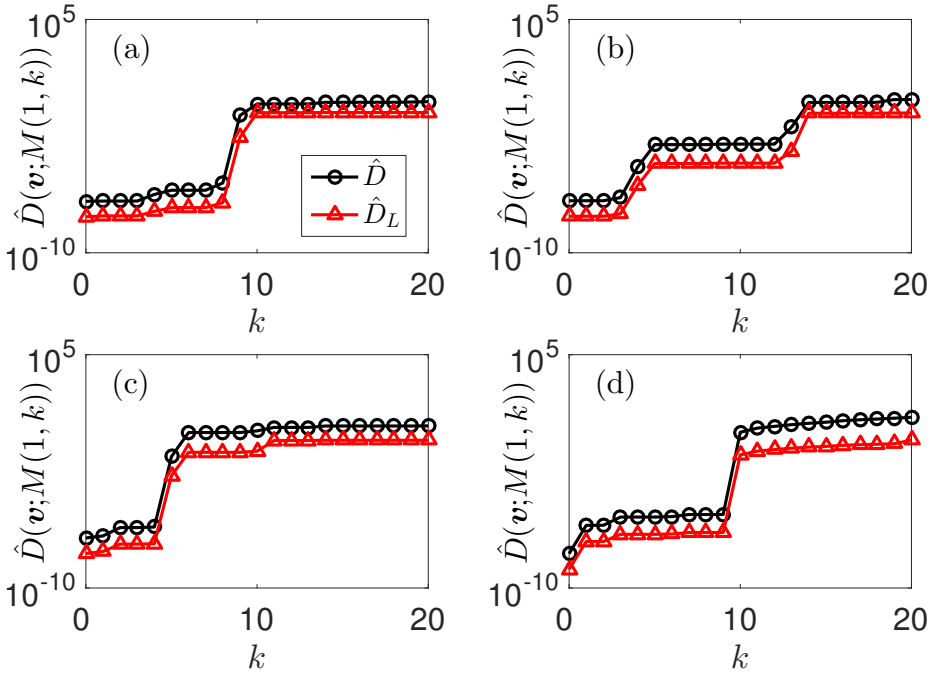

FIGURE 7.26: Comparison between the cumulative lower bound (5.65) (red line with triangles), and the DPGP-EKLD (5.23) (black line with circles). Panels (a)-(d) correspond to targets 1-4 in Fig. 7.25, respectively.

is compared to that of six existing camera control algorithms known as the optimal solution [175], entropy reduction algorithm [176], greedy algorithm [102], potential field [145], patrol algorithm [177] and random algorithm [83]. The optimal solution obtains the control inputs by solving a nonlinear programming problem with the DPGP-EKLD (5.23) as the objective function. Results are obtained using an SQP algorithm implemented by the MATLAB® Optimization Toolbox *fmincon* function [178]. The entropy reduction algorithm decides the next camera state by maximizing the entropy reduction in the DPGP-MM. The greedy algorithm maximizes the DPGP-EKLD (5.23) for one time step by a computationally efficient particle filter based search method. The potential field algorithm controls the camera movement by building attracting fields centered at predicted target states in the pan-tilt space. The patrol algorithm adopts a sliding mode based method to predefine a fixed route for the camera. The random algorithm generates multiple number of random con-

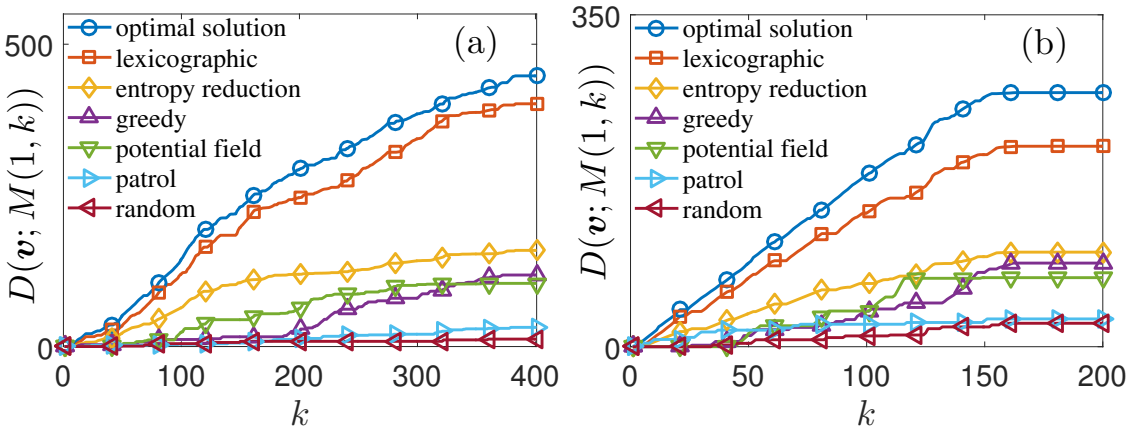

FIGURE 7.27: DPGP KL divergence obtained by the seven camera control algorithms for (a) pedestrian dataset I as shown in Fig. 7.24a, and (b) pedestrian dataset II as shown in Fig. 7.24b.

trol trajectories based on an extension of the Rapidly-exploring Random Tree, and chooses the control trajectory with the highest DPGP-EKLD.

The performance of the lexicographic method and the six comparing algorithms in terms of maximizing the information value are plotted in Fig. 7.27, using the datasets in Fig. 7.24, and the camera parameters in Table 7.1. It can be seen from Fig. 7.27 that the lexicographic method is superior to all the other algorithms except the optimal solution. In addition, the performance of the lexicographic method is close to that of the optimal solution. It is worth noticing that the computational complexity of the optimal solution approach is too high to be used in real time applications, as shown in Section 7.3.2.

In order to evaluate the accuracy of the target kinematics learned by the camera control algorithms at the final time of the simulations, the relative root-mean-square error (RMSE) of the DPGP-MM, denoted by $\varepsilon(k)$, is adopted. Recall that T_j denotes the number of measurements of the jth pedestrian in the test datasets, and that $\hat{\mathbf{v}}_j$ represents the predicted target velocity by the DPGP-MM. Then, the relative RMSE

is defined as follows,

$$\varepsilon(k) \triangleq \frac{1}{N_T} \sum_{j=1}^{N_T} \sum_{i=1}^{M} w_{ij} \sqrt{\frac{1}{T_j} \sum_{k=1}^{T_j} \frac{\|\mathbf{v}_j(k) - \hat{\mathbf{v}}_j(k)\|^2}{\|\mathbf{v}_j(k)\|^2}} \tag{7.25}$$

where N_T is the number of targets in the test datasets. The relative RMSEs for all the camera control algorithms using the pedestrian datasets I and II shown in Fig. 7.24 are summarized in Table 7.2. The 'all data' cases use all the measurements of the pedestrian movements in the training datasets for learning the DPGP-MMs, which correspond to the minimum errors that can be achieved by any algorithm. They are included in Table 7.2 to help demonstrate the performance of the camera control algorithms. The relative RMSEs in Table 7.2 show that the DPGP-MMs learned from the measurements obtained by the optimal solution and the lexicographic algorithm are the most accurate. In addition, the VFs learned by the lexicographic method at the final time of the simulation for dataset I are plotted in Fig. 7.28, superimposed with the movements of the test targets. Therefore, Fig. 7.28 is a visual demonstration that the target kinematics learned from the measurements by the lexicographic method are close to the ground truth.

Table 7.2: Relative RMSEs of DPGP-MMs

Algorithms	Dataset I	Dataset II
All data	8.97%	9.03%
Optimal solution	9.12%	9.58%
Lexicographic	9.15%	10.88%
Entropy reduction	16.25%	18.52%
Greedy	15.68%	17.89%
Potential field	29.72%	30.21%
Patrol	27.47%	40.17%
Random	92.81%	93.51%

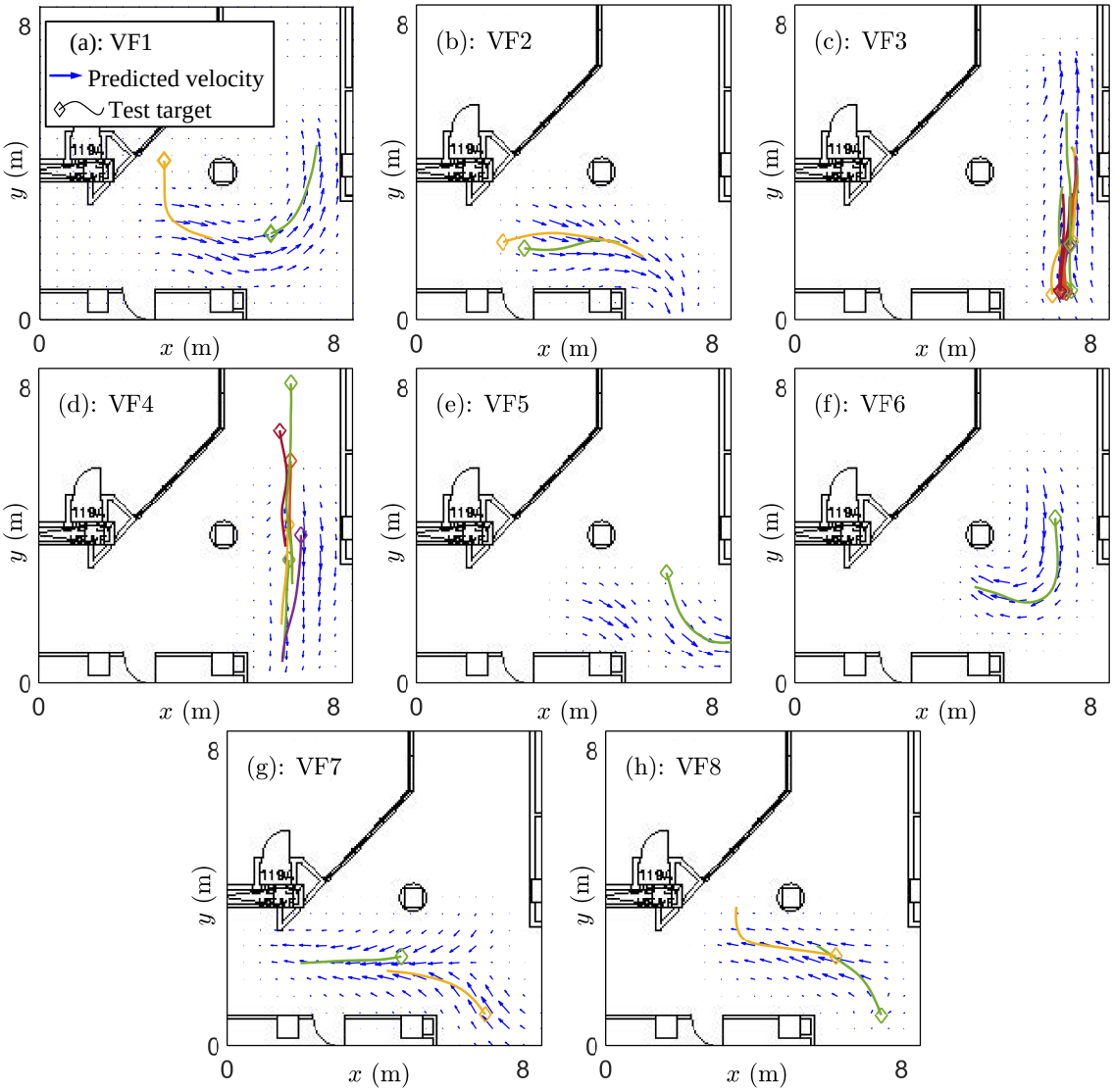

FIGURE 7.28: Target kinematic models learned by the lexicographic algorithm at the final time of the simulation superimposed with the movements of the test targets from the pedestrian dataset I in Fig. 7.24a. Targets are superimposed with the VF with the highest likelihood.

C. Computational Complexity

Because the camera control algorithms are designed for real-time applications, it is essential to analyse their computational complexity. The optimal solution algorithm was proven NP-hard in Section 5.3.5. The complexity of the lexicographic method is dominated by the calculations of the cumulative lower bounds (5.65), which take

$O(L^2K)$ time for every target and every VF, where L is the number of collocation points, and K is the control horizon. In addition, the single-objective quadratic optimization (6.17) takes $O(K^3)$ time. Therefore, the computational complexity of the lexicographic method is $O((L^2+K^2)MNK)$, where M is the number of VFs and the N is the number of targets in the workspace.

The theoretical computational complexities of all the algorithms are summarized in Table 7.3, which shows that the computational complexity of the lexicographic method is much lower than that of the optimal solution. The experimental computation time results in Table 7.3 are obtained on the same Dell Precision T7400 workstation, with a 3.20 GHz Intel(R) Xeon(R) CPU, and 16.0 GB installed memory. The experimental results show that the lexicographic algorithm is fast enough to be utilized in real time applications.

Table 7.3: Computational Complexity

Algorithms	Theoretical complexity	Experimental complexity (s)	
		Dataset I	Dataset II
Optimal solution	NP-hard	16.014	15.092
Lexicographic	$O\big((L^2+K^2)MNK\big)$	0.081	0.073
Entropy reduction	$O\big((L^2+K^2)MNK\big)$	0.077	0.072
Greedy	$O\big([L^2+\log(MN)]MN\big)$	0.044	0.044
Potential field	$O\big(L^2MN\big)$	0.003	0.003
Patrol	$O(1)$	< 0.001	< 0.001
Random	$O\big(L^2MNK\big)$	0.002	0.002

7.4 Related Topics and Extensions

Three real-life applications where the DPGP information value can be applied for controlling a single sensor have been discussed in Section 7.1-Section 7.3. The DPGP information value function can also be used to represent the utilities of future measurements in decentralized sensor planning problems, where multiple sensors are con-

trolled to collaboratively collect the most informative measurements. A key problem in decentralized sensor planning is to determine the communication times, such that necessary information obtained by individual sensors can be shared by the minimal number of communications for stealth requirements or saving energy. In Section 7.4.1, an intermittent communication control policy is derived based on the expected average information value of the Gaussian process. In addition, decentralized optimization of the DPGP information value function is discussed in Section 7.4.2, and three decentralized sensor planning algorithms and their applications are presented.

7.4.1 Extension of Scenario 1 to Intermittent Communication

The application Scenario 1 considers a single sensor. However, it can be extended to decentralized Gaussian process learning with intermittent communications. Consider M sensors deployed in the workspace to collaboratively observe the ocean currents as shown in Fig. 7.29. The learning process of the ocean current is *decentralized*, with only *intermittent* communications among the sensors. During the decentralized learning phase, the sensors are not allowed to communicate any information. In addition, the sensors can only observe their own states and get access to their own measurements of $\mathbf{f}(\cdot)$. In other words, the *local* information available to the ith sensor consists of only the history of its own positions and measurements, $\mathcal{M}_i(k) \triangleq \{\mathbf{s}_i(\ell), \mathbf{z}_i(\ell) \mid \ell = 1, \ldots, k\}$. Therefore, without communication, every sensor in the network can only update its estimation of the spatial phenomenon by the new data in $\mathcal{M}_i(k)$. In addition, the sensor planning policy of every sensor is assumed to be the GP-EKLD algorithm (Algorithm 7).

However, every sensor is able to initiate the communication among all the sensors in the network. Let $u_i(k) \in \{0, 1\}$ denote the binary communication control signal of the ith sensor, such that the event $\{u_i(k) = 1\}$ represents the communication is required by the ith sensor at the kth time step. For simplicity, it is assumed that

the sensors communicate when if any individual sensor initiates the communication, such that $\bigvee_{i=1}^{M} u_i(k) = 1$, where $\bigvee$ denotes the logical disjunction. At the time of communication, the sensors can construct a connected graph, where the nodes are the sensors and the edges are the communication channels between the sensors, to share obtained measurements by all the sensors, $\mathcal{M}(k) \triangleq \bigcup_{i=1}^{M} \mathcal{M}_i(k)$. Since the communication consumes energy, it should only be performed when necessary. To this end, the *communication control* is to design a communication control algorithm that determines $u_i(k)$ based on the locally available sensor measurements and the globally shared Gaussian process model.

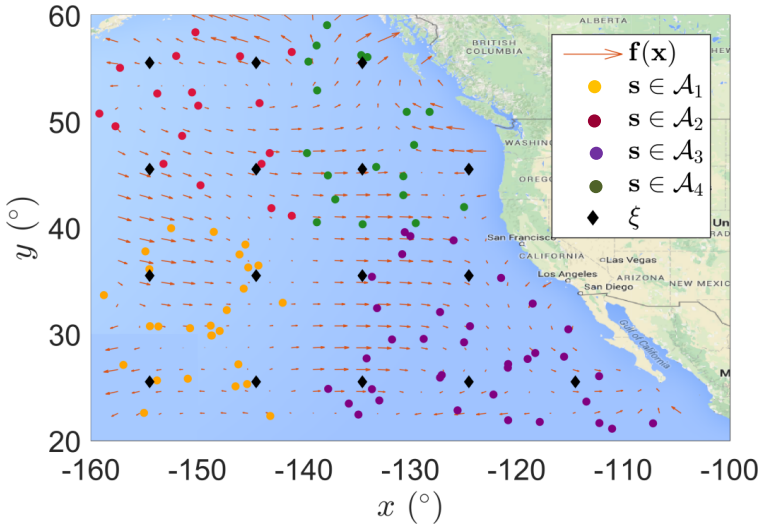

FIGURE 7.29: Example of workspace for extension of Scenario 1 with decentralized accessible sensor positions.

The prediction error, $\epsilon(k)$, defined in (7.2) provides a natural way of designing the communication criterion. Although (7.2) can not be applied to the intermittent communication control problem directly before measurements are taken, the expectation of $\epsilon(k)$ can be utilized,

$$\mathbb{E}[\epsilon(k)] \triangleq \frac{1}{L}\sum_{l=1}^{L} \mathbb{E}_{\mathbf{v}_l} \|\mathbf{v}_l - \hat{\mathbf{v}}_l(k)\|_2^2 = \frac{1}{L}\mathrm{tr}[\mathbf{\Sigma}_i(k)] \tag{7.26}$$

where $\boldsymbol{\Sigma}_i(k)$ is defined in (5.17) [179]. The GP prediction performance over the entire workspace is then evaluated by the expected average generalization error (EAGE),

$$\bar{\epsilon}(k) \triangleq \mathbb{E}_{\mathcal{M}(k)} \{\mathbb{E}[\epsilon(k)]\} \tag{7.27}$$

Since $\bar{\epsilon}(k)$ only depends on the time step k, it is also referred to as the *learning curve* in the literature [180, 181, 182].

The exact calculation of $\bar{\epsilon}(k)$ is computationally intractable in most cases, since it requires the marginalization over the joint distribution of the sensor positions and the collocation points [183]. To this end, an efficient recursive algorithm to approximating $\bar{\epsilon}(k)$ is proposed by exploiting the property that $\mathcal{A}_i$ are discrete and time-invariant sets. Let $\boldsymbol{\phi}_k(\cdot,\cdot)$ denote the GP posterior covariance matrix conditioned on $\mathcal{M}(k)$. Given a new sensor position of the ith sensor, $\mathbf{s}_i(k+1)$, the posterior covariance function can be updated as follows,

$$\boldsymbol{\phi}_{k+1}(\boldsymbol{\xi}_\imath, \boldsymbol{\xi}_\jmath) = \boldsymbol{\phi}_k(\boldsymbol{\xi}_\imath, \boldsymbol{\xi}_\jmath) - \frac{\boldsymbol{\phi}_k\big[\boldsymbol{\xi}_\imath, \mathbf{s}_i(k+1)\big]\boldsymbol{\phi}_k\big[\mathbf{s}_i(k+1), \boldsymbol{\xi}_\jmath\big]}{\boldsymbol{\phi}_k\big[\mathbf{s}_i(k+1), \mathbf{s}_i(k+1)\big] + \sigma_n^2 \mathbf{I}_2} \tag{7.28}$$

for $\imath, \jmath = 1, \ldots, L$, and $i = 1, \ldots, M$. Therefore, the expected GP posterior covariance function can be obtained by taking expectation of (7.28) with respect to the sensor positions. Assuming that the sensor measurements are independent, the expected GP posterior covariance can be approximated as follows,

$$\hat{\boldsymbol{\phi}}_{k+1}(\boldsymbol{\xi}_\imath, \boldsymbol{\xi}_\jmath) = \hat{\boldsymbol{\phi}}_k(\boldsymbol{\xi}_\imath, \boldsymbol{\xi}_\jmath) - \sum_{i=1}^{M} \sum_{\mathbf{s}_l \in \mathcal{A}_i} \frac{\hat{\boldsymbol{\phi}}_k(\boldsymbol{\xi}_\imath, \mathbf{s}_l)\hat{\boldsymbol{\phi}}_k(\mathbf{s}_l, \boldsymbol{\xi}_\jmath)}{\hat{\boldsymbol{\phi}}_k(\mathbf{s}_l, \mathbf{s}_l) + \sigma_n^2 \mathbf{I}_2} p(\mathbf{s}_l) \tag{7.29}$$

for $\imath, \jmath = 1, \ldots, L$, where $p(\mathbf{s}_l)$ denotes the probability that the sensor takes measurement at $\mathbf{s}_l \in \mathcal{A}_i$, which is determined by the sensor planning algorithm. Then, the EAGE can be approximated as follows,

$$\bar{\epsilon}(k) \approx \hat{\epsilon}(k) = \sum_{\imath=1}^{L} \frac{1}{L} \hat{\boldsymbol{\phi}}_k(\boldsymbol{\xi}_\imath, \boldsymbol{\xi}_\imath) \tag{7.30}$$

The *nominal* GP prediction performance, denoted by $\bar{\epsilon}_u(k)$, is defined with respect to the uniform planning algorithm that draws the sensor position from $\mathcal{A}_i$ uniformly at random, such that,

$$p(\mathbf{s}_l) = \begin{cases} \dfrac{1}{\text{Card}(\mathcal{A}_i)}, & \mathbf{s}_l \in \mathcal{A}_i \\ 0, & \mathbf{s}_l \notin \mathcal{A}_i \end{cases} \qquad i = 1, \ldots, M \tag{7.31}$$

where $\text{Card}(\cdot)$ denotes the cardinality of a set. In order to apply the approximation technique in (7.30), it is necessary to verify that the approximation, $\hat{\epsilon}_u(k)$, truthfully represents the nominal EAGE, $\bar{\epsilon}_u(k)$. Monte Carlo simulations are performed to obtain the nominal EAGE, $\bar{\epsilon}_u(k)$, for the spatial phenomenon in Fig. 7.29. The collocation points and the accessible sensor positions are also the same as shown in Fig. 7.29. Comparisons between $\hat{\epsilon}_u(k)$ and $\bar{\epsilon}_u(k)$ are conducted for various length-scales, λ_i, of the covariance function (2.14), since the length-scales are believed to be the most importance parameters. Figure 7.30 shows that $\hat{\epsilon}_u(k)$ is close to $\bar{\epsilon}_u(k)$ for both small and large length-scales. Therefore, $\hat{\epsilon}_u(k)$ can be used in stead of $\bar{\epsilon}_u(k)$ for developing the communication control algorithm to bypass the need of time-consuming Monte Carlo simulations.

Based on the nominal GP prediction performance, $\hat{\epsilon}_u(k)$, a heuristic policy determines the communication control signals, $u_i(k)$, can be developed. The idea of the heuristic communication control policy is that an individual sensor requests the measurements from other sensors in the network when the sensor believes the inquired measurements can help to decrease the generalization error of its model over a predefined threshold, $\gamma > 0$. The mathematical presentation of the heuristic communication control policy is as follows. Assuming k' denotes the last communication

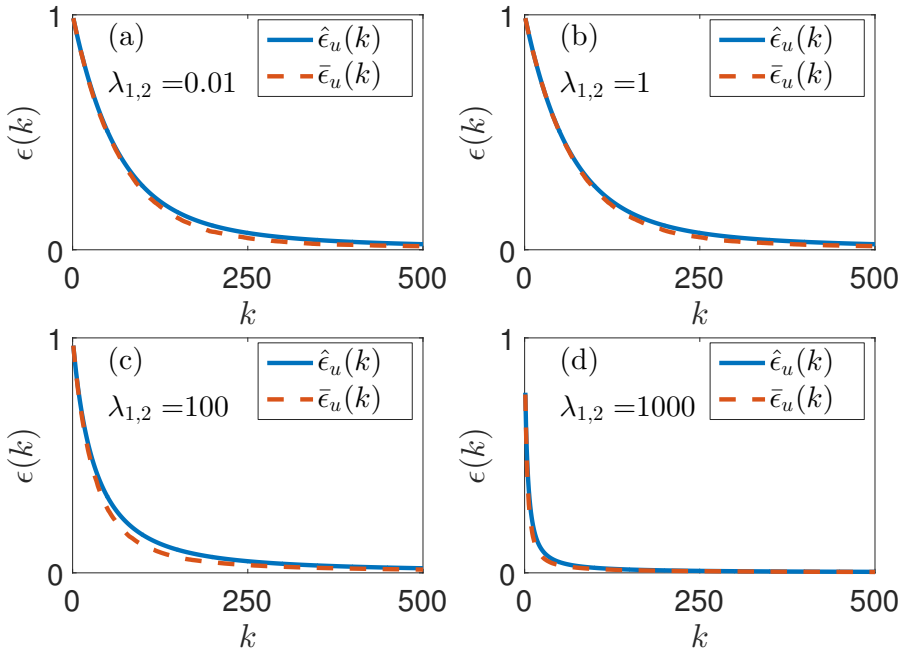

FIGURE 7.30: Example of nominal expected average generalization error, $\bar{\epsilon}_u(k)$, obtained by Monte Carlo experiments (red dashed line), compared with the approximation, $\hat{\epsilon}_u(k)$, (blue solid line) for various GP length-scales, λ_i.

time, the sensors communicate at time step k if $\bigvee_{i=1}^{N} u_i(k) = 1$, where,

$$u_i(k) = \begin{cases} 1, & \text{if } \sum_{\ell=k'}^{k} |\epsilon(k) - \hat{\epsilon}_u(\ell)| \geqslant \gamma \\ 0, & \text{otherwise} \end{cases}, \quad i = 1, \ldots, M \tag{7.32}$$

A schematic plot of the communication control algorithm is illustrated by Fig. 7.31. The integrated sensor planning and communication control algorithm is then presented as Algorithm 10.

The effect of the communication control policy in (7.32) on the generalization error of an individual sensor is shown in Fig. 7.32. The simulation is conducted by assuming $\lambda_1 = \lambda_2 = 10$ in the covariance function (2.14). The threshold in the communication policy (7.32) is chosen to be $\gamma = 0.4$. It can be seen that at the time of communications, the AGE calculated from the measurements taken by a single sensor, $\epsilon_1(k)$, can be reduced to the AGE calculated from the measurements taken

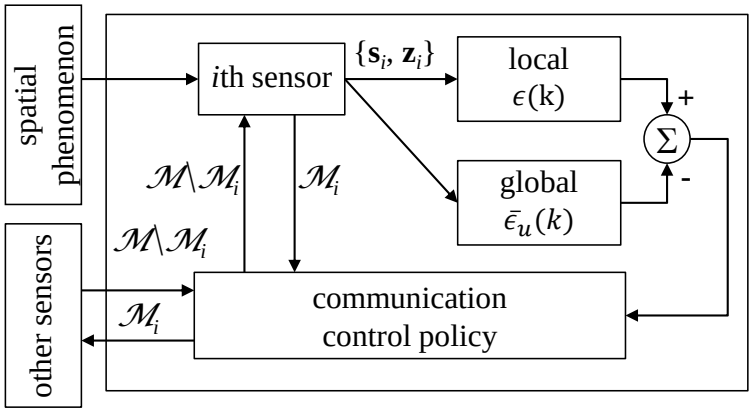

FIGURE 7.31: Schematic plot of the communication control policy.

Algorithm 10 Communication Control Algorithm for GP-EKLD Sensor Planning

Input: Accessible sensor positions $\{\mathcal{A}_i\}_{i=1}^{M}$; Collocation points $\boldsymbol{\xi}$; Communication threshold γ

Output: Communication control signal $u_i(k+1) \in \{0, 1\}$

1: Obtain $\mathbf{s}_i^*(k+1)$ by the GP-EKLD algorithm (Algorithm 7).
2: Update GP covariance matrix, $\boldsymbol{\phi}_{k+1}(\cdot, \cdot)$, with $\mathbf{s}_i^*(k+1)$ by (7.28).
3: Update expected GP covariance matrix $\hat{\boldsymbol{\phi}}_{k+1}(\cdot, \cdot)$ by (7.29)
4: Obtain the nominal EAGE of the sensor network, $\hat{\epsilon}_u(k+1)$, by (7.30)
5: Determine the communication control signal, $u_i(k+1)$, by (7.32).

by all the sensors, $\epsilon(k)$. In addition, the communication control policy (7.32) is able to automatically adjust the communication frequency based on the generalization errors. At the beginning of the simulation when the generalization error decreases fast, more frequent communications are performed. This is preferable, since communicating at this time can help reduce the uncertainty of the spatial phenomenon more efficiently. In contrast, towards the end of the simulation, when the new measurements become less informative, the communication control policy decreases the communication frequency and consumes less energy.

The spatial phenomenon under observation can also affect the communication control policy. Figure 7.33 shows that the frequency of the communication decreases as the length-scales of the GP increase. This behavior is preferable since smaller GP length-scale means the underlying spatial phenomenon is more complex and mea-

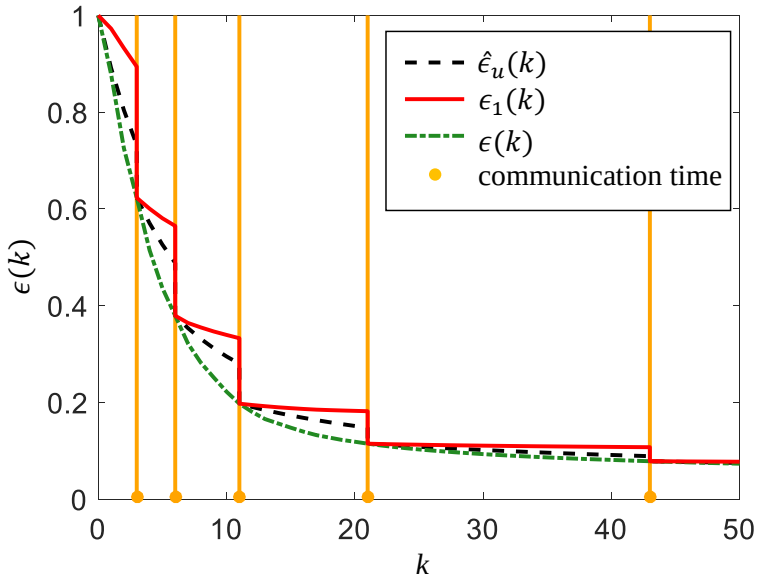

FIGURE 7.32: Prediction errors obtained by the GP-EKLD algorithm (Algorithm 7) for a single sensor, $\epsilon_1(k)$ (red solid line), and all the sensors, $\epsilon(k)$ (green dash-dotted line), compared with the approximated EAGE, $\hat{\epsilon}_u(k)$ (black dashed line).

surements of the spatial phenomenon by different sensors are less correlated. In other words, when GP length-scales are small, individual sensors can not predict the spatial phenomenon in the entire workspace by their own measurements, therefore, more frequent communications are needed as determined by the communication control algorithm (7.32).

Finally, the effect of the sensor planning policy on the intermittent communication control policy is studied. Figure 7.34 shows the communication control signals for the GP-EKLD algorithm (Algorithm 7) and the entropy algorithm (7.1). Recall that the GP-EKLD outperforms the entropy algorithm as shown in Section 7.1. By comparing the communication control signals, it can be concluded that less efficient sensor planning algorithms result in more frequent communications, which is preferable in practice.

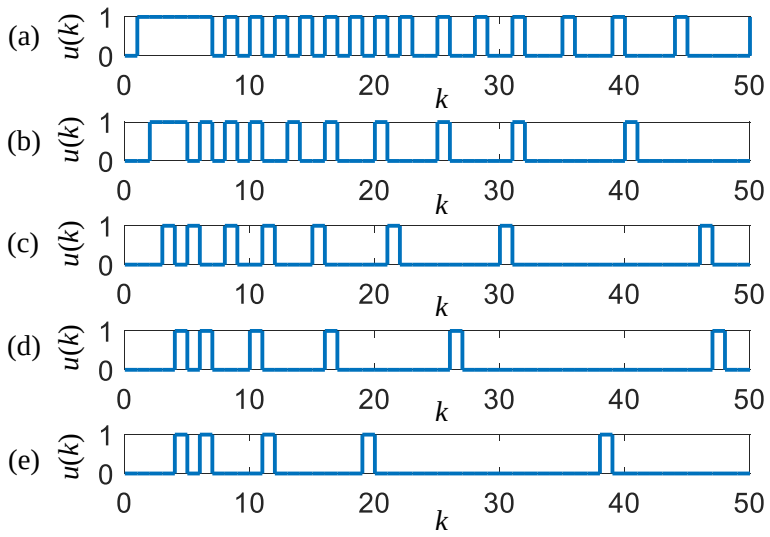

FIGURE 7.33: Communication control policy history for various GP length scales: (a) $\lambda_1 = \lambda_2 = 0.1$, (b) $\lambda_1 = \lambda_2 = 1$, (c) $\lambda_1 = \lambda_2 = 10$, (d) $\lambda_1 = \lambda_2 = 20$, (e) $\lambda_1 = \lambda_2 = 50$.

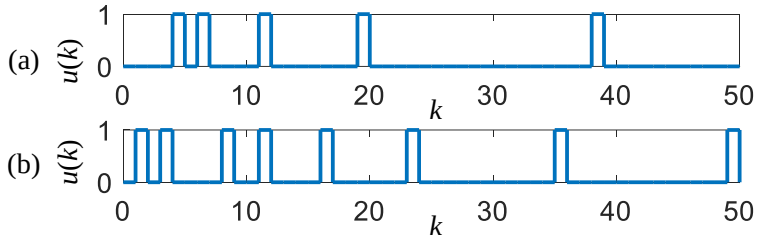

FIGURE 7.34: Communication control policy history for (a) GP-EKLD (Algorithm 7) and (b) the entropy algorithm (7.1).

7.4.2 Extension of Scenario 2 in Decentralized Sensor Planning

When multiple sensors are involved to surveil a common workspace, a decentralized controller for each sensor to cooperatively track moving targets is required to scale up the solution. To this end, the DPGP-EKLD-Greedy algorithm (Algorithm 7) discussed in Scenario 2 is extended to decentralized sensor planning, and three strategies are developed. The first two strategies transform the multiple sensor planning problem into a number of independent single sensor control problems. The first strategy groups targets based on the estimation of their positions by k-means

algorithm [184]. The number of target groups is equal to the number of sensors. Then each target group is assigned to one sensor. After the assignment, each sensor is controlled by the DPGP-EKLD-Greedy algorithm (Algorithm 8) to observe the targets assigned to it. Sensors communicate periodically and re-assign targets based on new information. In this algorithm the re-assigning period is pre-fixed. Figure 7.35a shows a snap shot of one simulation in which sensors are controlled by the first strategy to monitor up to six targets. Samples of different targets are marked by colors in Fig. 7.35a, and groups of targets are indicated by the dashed ellipse. The performance of the decentralized sensor planning algorithm is summarized in Fig. 7.35b and it can be seen that the performance of two decentralized sensors is close to the performance of a single sensor with half the number of targets. Therefore, the decentralized DPGP-EKLD algorithm is able scale up the solution without impairing the performance when the number of targets is increased.

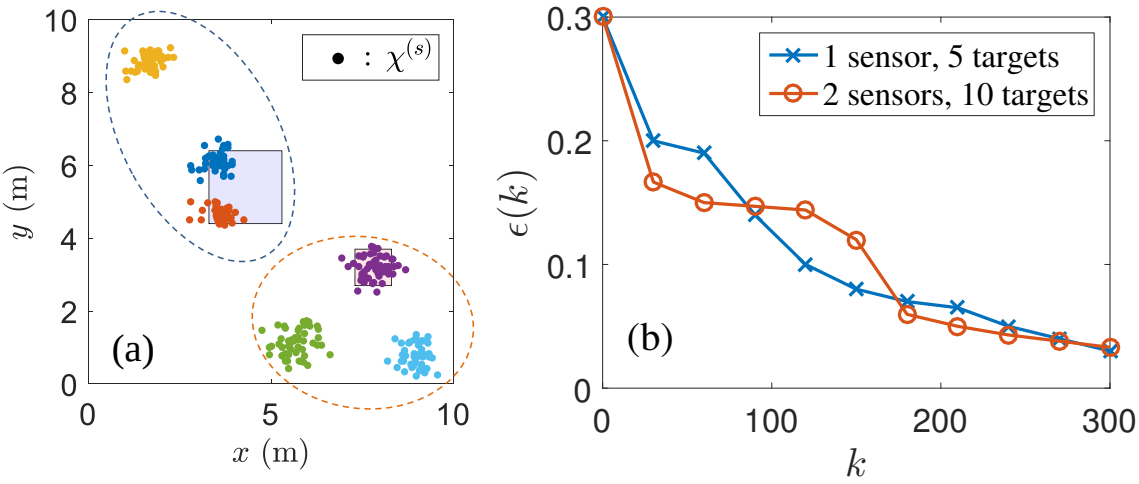

FIGURE 7.35: Snapshot (a) and performance comparisons (b) of decentralized DPGP-EKLD sensor planning algorithm by target assignment.

The second strategy is developed for the case when the movement of the N_S sensors' FOVs are constrained in sub-workspaces, $\mathcal{W}_i \subset \mathcal{W}$, for $i = 1, \ldots, N_S$, as shown by the green areas in Fig. 7.36. It is assumed that the union of the sub-workspaces

covers the entire workspace, such that $\mathcal{W} \subset \bigcup_{i=1}^{N_S} \mathcal{W}_i$. Each sensor only considers the targets present in its corresponding sub-workspace, and the sensor planning is determined by the DPGP-EKLD-Greedy algorithm (Algorithm 8). Figure 7.36a shows a snapshot of the decentralized sensor planning strategy with sub-workspace constraints. The performance of the decentralized sensor planning algorithm with sub-workspace constraints is summarized in Fig. 7.36b, where the 'centralized' result is obtained by a single sensor with two targets, and the 'decentralized' result is obtained by four sensors with eight targets. It can be seen from Fig. 7.36b that the DPGP-EKLD algorithm can be applied to decentralized sensor planning without impairing the performance when the sensors are deployed in sub-workspaces.

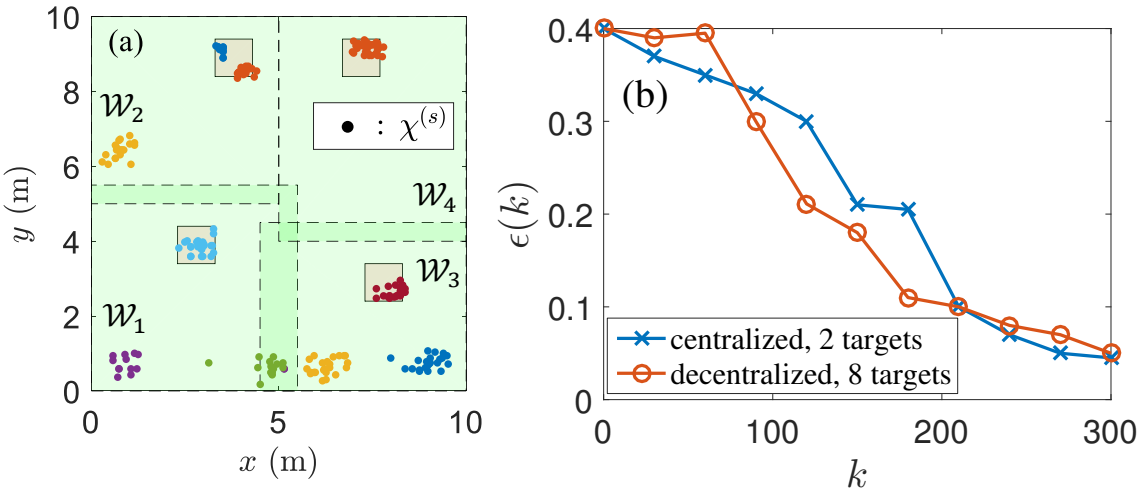

FIGURE 7.36: Snapshot (a) and performance comparisons (b) of decentralized DPGP-EKLD sensor planning algorithm with workspace constraints.

The last decentralized (or distributed) algorithm computes the local optimal control inputs for all sensors by decomposing the DPGP-EKLD into a set of reward functions, which are simultaneously optimized by the accelerated distributed augmented Lagrangian (ADAL) method [185]. Each reward is a function of one target and the sensors in the vicinity of the target. The decomposition of the DPGP-EKLD is performed by constructing a bipartite graph, where the two disjoint sets of nodes consist

of the sensors and the targets, respectively. The edges of the bipartite graph represents the decomposition of the DPGP-EKLD, and can be determined by algorithms, such as k-means, based on the estimated states of the targets and the states of the sensors. Figure. 7.37a, shows an example consisting of four targets, $\{T_j\}_{j=1}^{4}$, and five sensors, $\{S_i\}_{i=1}^{5}$. The edges of the bipartite are determined by the distance between the target samples and the center of the sensor FOV. The corresponding bipartite graph is show in Fig. 7.37b. To utilize the ADAL method, the DPGP-EKLD is augmented with the constraints that the control inputs for the same sensor obtained by maximizing different reward functions should be the same. By optimizing the reward functions with respect to the sensor control inputs and the Lagrangian multipliers, the DPGP-EKLD is maximized, and thus, local optimal control inputs for multiple sensors are obtained. Pseudocode of the augmented Lagrangian decentralized sensor planning algorithm is summarized in Algorithm 7.4.2.

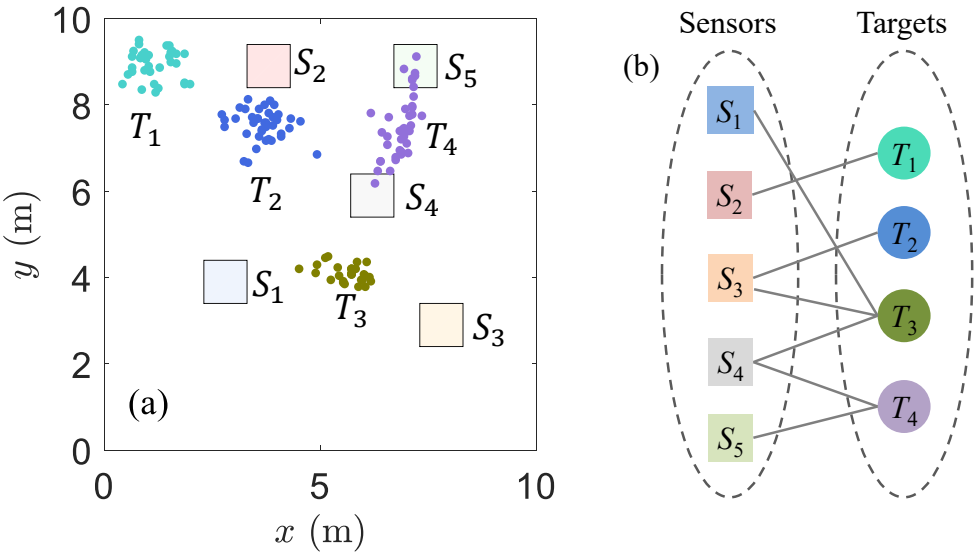

FIGURE 7.37: Snapshot (a) and bipartite graph (b) of decentralized DPGP-EKLD sensor planning algorithm by augmented Lagrangian.

Algorithm 11 Decentralized DPGP-EKLD Augmented Lagrangian

Input: Current sensor states, $\{\mathbf{s}_i(k)\}_{i=1}^{N_S}$; Estimations of target states, $\{\hat{\mathbf{x}}_j(k)\}_{j=1}^{N}$
Output: Sensor control inputs, $\{\mathbf{u}_i^*(k)\}_{i=1}^{N_S}$
1: Initialize Lagrange multipliers, $\{\lambda_i\}_{i=1}^{N_S}$
2: **while** $\{\mathbf{u}_i(k)\}_{i=1}^{N_S}$ does not converge **do**
3: Augment the DPGP-EKLD (5.37) by the Lagrange multipliers
4: Update $\{\mathbf{u}_i(k)\}_{i=1}^{N_S}$ and $\{\lambda_i\}_{i=1}^{N_S}$ by the ADAL algorithm [185]
5: **end while**

7.5 Chapter Conclusion

Three real-world applications of the proposed sensor planning algorithms (Algorithms 7, 8, 9) are discussed. The performance of the GP-EKLD-Greedy algorithm (Algorithm 7) is evaluated by an example of monitoring ocean currents in Section 7.1. The efficiency of the DPGP-EKLD-Greedy algorithm (Algorithm 8) is demonstrated by a pan-tilt camera surveillance task without considering the camera dynamics in Section 7.2. Finally, the lexicographic method (Algorithm 9) is applied when the dynamics of the pan-tilt camera are linear and time-invariant with constrained states and control inputs, as shown in Section 7.3. By comparing to existing algorithms, such as the entropy algorithm, the mutual information algorithm and the heuristic algorithms, it can be seen that the proposed algorithms are superior at obtaining informative measurements for improving the GP or DPGP target kinematics models.

8

Conclusions

Sensor planning problems for Bayesian nonparametric modeling are of interest for applications where no or little prior knowledge of the target or processes of interest is available, and, thus, the sensors must be automatically controlled to obtain the best measurements over time. Bayesian nonparametric models provide a systematic way of adapting the model parameters and dimensionality to data, and have been successfully applied to learn target kinematics in the form of velocity fields. The resulting models are the GP target kinematics model and the DPGP target kinematics model. Inference, prediction and filtering approaches have been developed to utilize the proposed GP and DPGP target kinematics models in the sensor planning problem, as described in Chapter 4. The GP particle filter (Algorithm 1 and 2) is presented for the prediction and filtering for a given GP target kinematics model. Similarly, the DPGP particle filter (Algorithm 5 and 6) is developed for the prediction and filtering for a given DPGP target kinematics model. The DPGP-Gibbs algorithm (Algorithm 4) is proposed for inference using a given DPGP target kinematics model.

Novel information theoretic functions are presented an analyzed to evaluate the utility of future measurements in closed form. For the GP target kinematics model

and single future measurement, the GP-EKLD has been derived in Theorem 4. For the DPGP target kinematics model and single future measurement, the DPGP-EKLD was derived and used to obtain an unbiased estimator of the DPGP-EKLD in Theorem 8. Then, the DPGP-EKLD was extended to multiple future measurements and a cumulative lower bound of the DPGP-EKLD was developed in Theorem 11 in order to develop efficient sensor planning algorithm with low computational complexity.

Three scenarios where the Bayesian nonparametric target kinematics models and the novel information values can be utilized were discussed, and three sensor planning algorithms were developed correspondingly. The GP-EKLD-Greedy algorithm (Algorithm 7) was developed for the least-constrained scenario with always observable target kinematics modeled by a single GP and discrete control space. The performance of the GP-EKLD-Greedy algorithm was evaluated by an example of monitoring ocean currents with data collected from moored buoys. The DPGP-EKLD-Greedy algorithm (Algorithm 8) was developed for the scenario involving multiple mobile targets and unconstrained sensor dynamics, and was applied in a PT camera surveillance task without considering the camera dynamics. The lexicographic algorithm (Algorithm 9) was developed for the scenario where sensor dynamics are linear and time-invariant with constrained sensor state and control inputs.

By comparing to existing algorithms in the literature, such as the entropy algorithm, the mutual information algorithm and the heuristic algorithms, it can be seen that the proposed informative-driven sensor planning algorithms are superior at obtaining informative measurements for improving the Bayesian nonparametric target kinematics models. Therefore, the proposed algorithms are preferable when the target kinematics under observation are complex with little or no prior information, and when learning the target kinematics is key to the success of the sensor planning tasks.

Appendix A

Supplement Materials

A.1 Probability Density Function of Multivariate Gaussian Distribution

The probability density function of a multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, denoted by $f_G(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$, is defined as follows,

$$f_G(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \triangleq \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right) \tag{A.1}$$

where d is the dimension of $\mathbf{x}$.

A.2 Proof of Theorem 4

Proof. From the KL divergence of multivariate Gaussian distributions, the KL divergence in (5.15) can be calculated analytically as follows:

$$\begin{aligned} D = &\frac{1}{2}\left\{\mathrm{tr}[\boldsymbol{\Sigma}_i(k)^{-1}\boldsymbol{\Sigma}_i(k+1)] - \ln\left\{\det[\boldsymbol{\Sigma}_i(k+1)\boldsymbol{\Sigma}_i(k)^{-1}]\right\} - 2L\right\} \\ &+ \frac{1}{2}[\boldsymbol{\mu}_i(k+1) - \boldsymbol{\mu}_i(k)]^T \boldsymbol{\Sigma}_i(k)^{-1}[\boldsymbol{\mu}_i(k+1) - \boldsymbol{\mu}_i(k)] \end{aligned} \tag{A.2}$$

To simplify the integral in (5.15), we use the matrix inversion lemma for $\mathbf{\Sigma}_{i,k+1}$, such that,

$$\begin{aligned}\mathbf{\Sigma}_i(k+1)^{-1} &= \begin{bmatrix} \mathbf{\Phi}[\mathbf{Y}_i(k),\mathbf{Y}_i(k)] + \sigma_v^2\mathbf{I} & \mathbf{\Phi}[\mathbf{Y}_i(k),\mathbf{y}_j(k+1)] \\ \mathbf{\Phi}[\mathbf{y}_j(k+1),\mathbf{Y}_i(k)] & \mathbf{\Phi}[\mathbf{y}_j(k+1),\mathbf{y}_j(k+1)] + \sigma_v^2\mathbf{I} \end{bmatrix}^{-1} \\ &\triangleq \begin{bmatrix} \mathbf{\Sigma} & \mathbf{C} \\ \mathbf{C}^T & \mathbf{D} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{\Sigma}^{-1}(\mathbf{I}+\mathbf{C}\mathbf{Q}^{-1}\mathbf{C}^T\mathbf{\Sigma}^{-1}) & -\mathbf{\Sigma}^{-1}\mathbf{C}\mathbf{Q}^{-1} \\ \mathbf{Q}^{-1}\mathbf{C}^T\mathbf{\Sigma}^{-1} & \mathbf{Q}^{-1} \end{bmatrix}\end{aligned} \tag{A.3}$$

where $\mathbf{Q} = \mathbf{D} - \mathbf{C}^T\mathbf{\Sigma}^{-1}\mathbf{C}$ is defined in (5.20). Substituting (A.3) into (5.16) gives that,

$$\boldsymbol{\mu}_i(k+1) - \boldsymbol{\mu}_i(k) = \mathbf{R}\mathbf{Q}^{-1}\{\mathbf{z}_j(k+1) - \mathbb{E}[\mathbf{z}_j(k+1)]\} \triangleq \mathbf{R}\mathbf{Q}^{-1}\bar{\mathbf{z}} \tag{A.4}$$

where $\bar{\mathbf{z}}$ is Gaussian distributed with zero mean and covariance $\sigma_v^2\mathbf{I}$. Substituting (A.4) into (A.2) yields that

$$\begin{aligned}\hat{D} &= \frac{1}{2}\left[\mathrm{tr}\left(\mathbf{\Sigma}_{i,k}^{-1}\mathbf{\Sigma}_{i,k+1}\right) - \ln\left(\frac{|\mathbf{\Sigma}_{i,k+1}|}{|\mathbf{\Sigma}_{i,k}|}\right) - 2L\right] + \frac{1}{2}\int_{\mathbb{R}} \mathbf{Q}^{-1}\mathbf{R}^T\mathbf{\Sigma}_{i,k}^{-1}\mathbf{R}\mathbf{Q}^{-1}\bar{\mathbf{z}}^2 p(\bar{\mathbf{z}})d\bar{\mathbf{z}} \\ &= \frac{1}{2}\left[\mathrm{tr}\left(\mathbf{\Sigma}_{i,k}^{-1}\mathbf{\Sigma}_{i,k+1}\right) - \ln\left(\frac{|\mathbf{\Sigma}_{i,k+1}|}{|\mathbf{\Sigma}_{i,k}|}\right) - 2L + \mathrm{tr}(\mathbf{Q}^{-1}\mathbf{R}^T\mathbf{\Sigma}_{i,k}^{-1}\mathbf{R}\mathbf{Q}^{-1})\sigma_v^2\right]\end{aligned} \tag{A.5}$$

□

A.3 Proof of Equation (5.26)

Proof. When $G_j(k+1) = i$,

$$p(\boldsymbol{\upsilon}|Q(k+1)) = p(\boldsymbol{\upsilon}_i|Q(k+1)) \prod_{1\leqslant l\leqslant M,\ l\neq i} p(\boldsymbol{\upsilon}_l|Q(k)) \tag{A.6}$$

since $\boldsymbol{\upsilon}_i = \boldsymbol{\upsilon}_i(\boldsymbol{\xi})$ and $\boldsymbol{\upsilon}_j = \boldsymbol{\upsilon}_j(\boldsymbol{\xi})$ are conditional independent given $Q(k+1)$. Substituting (A.6) into (5.23), the conditional KL divergence can be written as

$$\begin{aligned} D\left(\boldsymbol{\upsilon}; \mathbf{m}_j(k+1)|Q(k)\right) &= \int_{\mathbb{R}^{2LM}} \ln \left\{ \frac{p(\boldsymbol{\upsilon}_i|Q(k+1)) \prod_{1\leqslant l\leqslant M,\ l\neq i} p(\boldsymbol{\upsilon}_l(\boldsymbol{\xi})|Q(k))}{\prod_{\ell=1}^{2M} p(\boldsymbol{\upsilon}_\ell(\boldsymbol{\xi})|Q(k))} \right\} \\ &\quad \cdot p\big(\boldsymbol{\upsilon}|Q(k+1)\big)\ d\boldsymbol{\upsilon} \\ &= \int_{\mathbb{R}^{2M}} \ln \left\{ \frac{p\big(\boldsymbol{\upsilon}_i|Q(k+1)\big)}{p\big(\boldsymbol{\upsilon}_i|Q(k)\big)} \right\} p\big(\boldsymbol{\upsilon}_i|Q(k+1)\big) d\boldsymbol{\upsilon}_i \\ &\quad \cdot \prod_{\substack{1\leqslant l\leqslant M \\ l\neq i}} \int_{\mathbb{R}^{2M}} p(\boldsymbol{\upsilon}_l(\boldsymbol{\xi})|Q(k))\ d\boldsymbol{\upsilon}_l(\boldsymbol{\xi}) \\ &= D\left(\boldsymbol{\upsilon}_i; \mathbf{m}_j(k+1)|Q(k)\right) \end{aligned} \tag{A.7}$$

□

A.4 Lemma on Complexity of Optimizing Entropy

For the completeness of the dissertation, Theorem 1 in [153] is introduced, which adopts the notation in this dissertation.

Lemma 17. *Given rational number n and rational covariance matrix $\boldsymbol{\Sigma}$ over a set of Gaussian random variables S, deciding whether there exits a subset $A \subset S$ of cardinality d such that $H(A) \geqslant n$ is NP-complete, where $H(\cdot)$ denotes the entropy function.*

A.5 Mutual Information

This section shows the equivalence between the expected entropy reduction and the mutual information, and presents the details about the MI criterion.

Proof. Recall that $H(\cdot)$ denotes the differential entropy of a continuous random variable. Let the random variable, $\hat{\mathbf{x}}_j(k+1)$, denote the predicted target position at the

next time step given $\mathcal{M}(k)$. Recalling that the random variable, $\mathbf{m}_j(k+1)$, denotes the measurement at the $(k+1)$th time step, from the property of mutual information, it is true that,

$$H[\hat{\mathbf{x}}_j(k+1)] - H[\hat{\mathbf{x}}_j(k+1) \mid \mathbf{m}_j(k+1)] = I[\hat{\mathbf{x}}_j(k+1) \mid \mathbf{m}_j(k+1)] \tag{A.8}$$

where $I(\cdot|\cdot)$ denotes the mutual information between two random variables. □

To evaluate the mutual information, (A.8), we use the assumption that the position measurement noise is zero. Therefore, when the target is in the sensor FOV at the next time step, the expected entropy reduction is $H[\hat{\mathbf{x}}_j(k+1)] \times P_d$, where P_d is the probability of detection:

$$P_d \triangleq \int_{\mathcal{S}(k+1)} p[\hat{\mathbf{x}}_j(k+1)] d\hat{\mathbf{x}}_j(k+1) \tag{A.9}$$

When the sensor fails to observe the target at the next time step, there is still positive expected entropy reduction, since the target position distribution is refined to

$$\frac{p[\hat{\mathbf{x}}_j(k+1)]\mathbf{1}_{\mathcal{W}\setminus\mathcal{S}(k+1)}[\hat{\mathbf{x}}_j(k+1)]}{\int_{\mathcal{W}} p[\hat{\mathbf{x}}_j(k+1)]\mathbf{1}_{\mathcal{W}\setminus\mathcal{S}(k+1)}[\hat{\mathbf{x}}_j(k+1)] d\hat{\mathbf{x}}_j(k+1)} \triangleq q[\mathbf{x}_j(k+1)] \tag{A.10}$$

Therefore, the mutual information considering the two cases is,

$$\begin{aligned} I[\hat{\mathbf{x}}_j(k+1) \mid \mathbf{m}_j(k+1)] = (1-P_d) \int_{\mathcal{W}} q[\mathbf{x}_j(k+1)] \log q[\mathbf{x}_j(k+1)] d\mathbf{x}_j \\ + H[\hat{\mathbf{x}}_j(k+1)] \end{aligned} \tag{A.11}$$

A.6 Pinhole Camera Model

From the pinhole camera model, it follows that,

$$\mathbf{p}_j = \lambda [q_x/q_z \quad q_y/q_z]^T \tag{A.12}$$

where

$$[q_x \quad q_y \quad q_z]^T = \mathbf{R}_\phi^T \mathbf{R}_\psi^T \left([\mathbf{x}_j^T \ 0]^T - \mathbf{x}_c \right) \tag{A.13}$$

$$\mathbf{R}_\phi \triangleq \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}, \quad \mathbf{R}_\psi \triangleq \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and $\mathbf{x}_c = [x_c \quad y_c \quad z_c]^T$ is the position of the origin of $\mathbf{F}_b$ with respect to $\mathcal{F}_\mathcal{W}$.

Taking the time derivatives of both sides of (A.12), it follows that,

$$\dot{\mathbf{p}}_j = \mathbf{F} \begin{bmatrix} \mathbf{R}_\phi^T \mathbf{R}_\psi^T & \mathbf{0} \\ \mathbf{0} & -\mathbf{R}_\phi^T \end{bmatrix} [\dot{x}_j \ \dot{y}_j \ 0 \ \dot{\phi} \ 0 \ \dot{\psi}]^T \tag{A.14}$$

where $\mathbf{F}$ is the image Jacobian matrix [172],

$$\mathbf{F} \triangleq \begin{bmatrix} -\frac{\lambda}{q_z} & 0 & \frac{p_x}{q_z} & \frac{p_x p_y}{\lambda} & -\frac{\lambda^2+p_x^2}{\lambda} & p_y \\ 0 & -\frac{\lambda}{q_z} & \frac{p_y}{q_z} & \frac{\lambda^2+p_x^2}{\lambda} & -\frac{p_x p_y}{\lambda} & -p_x \end{bmatrix} \tag{A.15}$$

Bibliography

[1] H. Wei and S. Ferrari. A geometric transversals approach to analyzing the probability of track detection for maneuvering targets. *Computers, IEEE Transactions on*, 63(11):2633–2646, 2014.

[2] Ian Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks: A survey. *Computer networks*, 38(4):393–422, 2002.

[3] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Computer Networks*, 52(12):2292 – 2330, 2008.

[4] H. Wei, W. Lu, P. Zhu, G. Huang, J. Leonard, and S. Ferrari. Optimized visibility motion planning for target tracking and localization. In *Intelligent Robots and Systems, 2014 IEEE/RSJ International Conference On*, pages 76–82. IEEE, 2014.

[5] X. R. Li and Vesselin P. Jilkov. Survey of maneuvering target tracking. part i. dynamic models. *Aerospace and Electronic Systems, IEEE Transactions on*, 39(4):1333–1364, 2003.

[6] Dazhi Chen and Pramod K. Varshney. Qos support in wireless sensor networks: A survey. In *International Conference on Wireless Networks*, volume 233, 2004.

[7] Rolf Johansson. System modeling & identification. 1993.

[8] Lennart Ljung. System identification: Theory for the user. *PTR Prentice Hall Information and System Sciences Series*, 198, 1987.

[9] Torsten Söderström and Petre Stoica. *System Identification*. Prentice-Hall, Inc., 1988.

[10] Lennart Ljung. Perspectives on system identification. *Annual Reviews in Control*, 34(1):1–12, 2010.

[11] Chi-Tsong Chen. *Linear system theory and design*. Oxford University Press, Inc., 1995.

[12] Wilson J Rugh. *Linear System Theory*, volume 2. prentice hall Upper Saddle River, NJ, 1996.

[13] Peter Orbanz and Yee Whye Teh. Bayesian nonparametric models. pages 81–89, 2010.

[14] W. Lu, G. Zhang, and S. Ferrari. A comparison of information theoretic functions for tracking maneuvering targets. In *Proc. of Statistical Signal Processing Workshop (SSP), 2012 IEEE*, pages 149–152, August 2012.

[15] C. Cai and S. Ferrari. Information-driven sensor path planning by approximate cell decomposition. *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, 39(3):607–625, 2009.

[16] F. Zhao, J. Shin, and J. Reich. Information-driven dynamic sensor collaboration. *IEEE Signal Processing Magazine*, 19:61–72, 2002.

[17] J. Denzler and C. M. Brown. Information theoretic sensor data selection for active object recognition and state estimation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(2):145–157, 2002.

[18] S. Ferrari and C. Cai. Information-driven search strategies in the board game of clue. *Systems, Man, and Cybernetics - Part B, IEEE Transactions on*, 39(3):607–625, 2009.

[19] G. Zhang, S. Ferrari, and C. Cai. A comparison of information functions and search strategies for sensor planning in target classification. *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, 42(1):2–16, 2012.

[20] Thomas Bayes and Richard Price. An essay towards solving a problem in the doctrine of chances. by the late rev. Mr. Bayes, FRS communicated by Mr. Price, in a letter to John Canton, AMFRS. *Philosophical Transactions*, pages 370–418, 1763.

[21] Nils Lid Hjort, Chris Holmes, Peter Müller, and Stephen G Walker. *Bayesian Nonparametrics*, volume 28. Cambridge University Press, 2010.

[22] Alan E Gelfand, Susan E Hills, Amy Racine-Poon, and Adrian FM Smith. Illustration of Bayesian inference in normal data models using Gibbs sampling. *Journal of the American Statistical Association*, 85(412):972–985, 1990.

[23] Theodoros Kypraios. *Efficient Bayesian Inference for Partially Observed Stochastic Epidemics and a New Class of Semi-parametric Time Series Models*. PhD thesis, 2007.

[24] Kate Cowles, Rob Kass, and Tony OHagan. What is Bayesian analysis? *International Society for Bayesian Analysis*, 2009.

[25] George EP Box and George C Tiao. *Bayesian Inference in Statistical Analysis*, volume 40. John Wiley & Sons, 2011.

[26] Bradley Efron. Bayesians, frequentists, and scientists. *Journal of the American Statistical Association*, 100(469):1–5, 2005.

[27] J Martin Bland and Douglas G Altman. Bayesians and frequentists. *British Medical Journal*, 317(7166):1151–1160, 1998.

[28] Samuel J. Gershman and David M. Blei. A tutorial on Bayesian nonparametric models. *Journal of Mathematical Psychology*, 56(1):1–12, 2012.

[29] Gerda Claeskens and Nils Lid Hjort. *Model Selection and Model Averaging*, volume 330. Cambridge University Press Cambridge, 2008.

[30] Walter Zucchini. An introduction to model selection. *Journal of mathematical psychology*, 44(1):41–61, 2000.

[31] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Occams razor. *Readings in machine learning*, pages 201–204, 1990.

[32] José M Bernardo and Adrian FM Smith. *Bayesian Theory*, volume 405. John Wiley & Sons, 2009.

[33] Peter Müller and Fernando A. Quintana. Nonparametric Bayesian data analysis. *Statistical science*, pages 95–110, 2004.

[34] Robert Henry Klein. *Planning Under Uncertainty with Bayesian Nonparametric Models*. PhD thesis, Massachusetts Institute of Technology, 2014.

[35] Andreĭ Nikolaevich Kolmogorov. *Foundations of the Theory of Probability*. Chelsea Publishing Co., 1950.

[36] David Blackwell and James B MacQueen. Ferguson distributions via pólya urn schemes. *The annals of statistics*, pages 353–355, 1973.

[37] David J Aldous. *Exchangeability and Related Topics*. Springer, 1985.

[38] Thomas S. Ferguson. A Bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230, 1973.

[39] Charles E. Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The annals of statistics*, pages 1152–1174, 1974.

[40] Michael D. Escobar and Mike West. Bayesian density estimation and inference using mixtures. *Journal of the american statistical association*, 90(430):577–588, 1995.

[41] Radford M. Neal. Markov chain sampling methods for Dirichlet process mixture models. *Journal of computational and graphical statistics*, 9(2):249–265, 2000.

[42] Stephen G Walker, Paul Damien, Purushottam W Laud, and Adrian FM Smith. Bayesian nonparametric inference for random distributions and related functions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):485–527, 1999.

[43] C. E. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[44] Kihwan Kim, Dongryeol Lee, and Irfan Essa. Gaussian process regression flow for analysis of motion trajectories. In *Computer Vision (ICCV), 2011 IEEE International Conference On*, pages 1164–1171. IEEE, 2011.

[45] Simo Sarkka, Arno Solin, and Jouni Hartikainen. Spatiotemporal learning via infinite-dimensional Bayesian filtering and smoothing: A look at Gaussian process regression through Kalman filtering. *Signal Processing Magazine, IEEE*, 30(4):51–61, 2013.

[46] Jouni Hartikainen and Simo Sarkka. Kalman filtering and smoothing solutions to temporal Gaussian process regression models. In *Machine Learning for Signal Processing (MLSP), 2010 IEEE International Workshop On*, pages 379–384. IEEE, 2010.

[47] Hannes Nickisch and Carl Edward Rasmussen. Approximations for binary Gaussian process classification. *Journal of Machine Learning Research*, 9:2035–2078, 2008.

[48] J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. Smith. Regression and classification using Gaussian process priors. In *Bayesian Statistics 6: Proceedings of the Sixth Valencia International Meeting*, volume 6, page 475, 1998.

[49] Radford M. Neal. Monte Carlo implementation of Gaussian process models for Bayesian regression and classification. 1997.

[50] C. K. Williams and D. Barber. Bayesian classification with Gaussian processes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(12):1342–1351, 1998.

[51] Manfred Opper and Ole Winther. Gaussian processes for classification: Mean-field algorithms. *Neural Computation*, 12(11):2655–2684, 2000.

[52] Jim Pitman et al. Combinatorial stochastic processes. Technical report, Springer, 2002.

[53] John R. Anderson. The adaptive nature of human categorization. *Psychological Review*, 98(3):409, 1991.

[54] C. E. Rasmussen. The infinite Gaussian mixture model. In *Proc. of Advances in Neural Information Processing Systems (NIPS)*, volume 12, pages 554–560, Denver, CO, USA, Dec 1999.

[55] Zoubin Ghahramani and David Knowles. Nonparametric Bayesian sparse factor models with application to gene expression modelling. 2010.

[56] Thomas Griffiths and Zoubin Ghahramani. Infinite latent feature models and the indian buffet process. 2005.

[57] Zoubin Ghahramani, Thomas L. Griffiths, and Peter Sollich. Bayesian nonparametric latent feature models. 2007.

[58] David M. Blei, Perry Cook, and Matthew Hoffman. Bayesian nonparametric matrix factorization for recorded music. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 439–446, 2010.

[59] Thomas Griffiths and Zoubin Ghahramani. The indian buffet process: An introduction and review. *The Journal of Machine Learning Research*, 12:1185–1224, 2011.

[60] Matthew J. Beal, Zoubin Ghahramani, and Carl E. Rasmussen. The infinite hidden markov model. In *Advances in Neural Information Processing Systems*, pages 577–584, 2001.

[61] John Paisley and Lawrence Carin. Hidden markov models with stick-breaking priors. *Signal Processing, IEEE Transactions on*, 57(10):3905–3917, 2009.

[62] Jurgen V. Gael, Yee W. Teh, and Zoubin Ghahramani. The infinite factorial hidden markov model. In *Advances in Neural Information Processing Systems*, pages 1697–1704, 2009.

[63] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the american statistical association*, 101(476), 2006.

[64] Zhuowen Tu and Song-Chun Zhu. Image segmentation by data-driven markov chain monte carlo. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):657–673, 2002.

[65] Mingyuan Zhou, Haojun Chen, Lu Ren, Guillermo Sapiro, Lawrence Carin, and John W. Paisley. Non-parametric Bayesian dictionary learning for sparse image representations. In *Advances in Neural Information Processing Systems*, pages 2295–2303, 2009.

[66] Mingyuan Zhou, Haojun Chen, John Paisley, Lu Ren, Lingbo Li, Zhengming Xing, David Dunson, Guillermo Sapiro, and Lawrence Carin. Nonparametric Bayesian dictionary learning for analysis of noisy and incomplete images. *Image Processing, IEEE Transactions on*, 21(1):130–144, 2012.

[67] Peter Orbanz and Joachim M. Buhmann. Nonparametric Bayesian image segmentation. *International Journal of Computer Vision*, 77(1-3):25–45, 2008.

[68] David M. Blei, Thomas L. Griffiths, and Michael I. Jordan. The nested chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *Journal of the ACM (JACM)*, 57(2):7, 2010.

[69] David M. Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

[70] D. Griffiths and M. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. *Advances in neural information processing systems*, 16:17, 2004.

[71] Sinead Williamson, Chong Wang, Katherine Heller, and David Blei. The ibp compound Dirichlet process and its application to focused topic modeling. 2010.

[72] Jyri J. Kivinen, Erik B. Sudderth, and Michael I. Jordan. Learning multiscale representations of natural scenes using Dirichlet processes. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference On*, pages 1–8. IEEE, 2007.

[73] Agathe Girard, Carl Edward Rasmussen, Joaquin Qui nonero Candela, and Roderick Murray-Smith. Gaussian process priors with uncertain inputs application to multiple-step ahead time series forecasting. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 545–552. MIT Press, 2003.

[74] Jonathan Ko, Daniel J Klein, Dieter Fox, and Dirk Haehnel. Gaussian processes and reinforcement learning for identification and control of an autonomous blimp. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 742–747. IEEE, 2007.

[75] Jonathan Ko, Daniel J Klein, Dieter Fox, and Dirk Haehnel. GP-UKF: Unscented Kalman Filters with Gaussian process prediction and observation models. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 1901–1907. IEEE, 2007.

[76] Jack Wang, David Fleet, and Aaron Hertzmann. Gaussian process dynamical models for human motion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):283–298, 2008.

[77] J. Wang, Y. Yin, and M. Hong. Multiple human tracking using particle filter with Gaussian process dynamical model. *EURASIP Journal on Image and Video Processing*, 2008.

[78] Jonathan Ko and Dieter Fox. GP-Bayesfilters: Bayesian filtering using Gaussian process prediction and observation models. *Autonomous Robots*, 27(1):75–90, 2009.

[79] Marc Peter Deisenroth, Carl Edward Rasmussen, and Jan Peters. Gaussian process dynamic programming. *Neurocomputing*, 72(7):1508–1524, 2009.

[80] Jonathan Ko and Dieter Fox. Learning GP-Bayesfilters via Gaussian process latent variable models. *Autonomous Robots*, 30(1):3–23, 2011.

[81] David Ellis, Eric Sommerlade, and Ian Reid. Modelling pedestrian trajectory patterns with Gaussian processes. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 1229–1234. IEEE, 2009.

[82] Richard Mann, Robin Freeman, Michael Osborne, Roman Garnett, Jessica Meade, Chris Armstrong, Dora Biro, Tim Guilford, Stephen Roberts, Paul M Goggans, et al. Gaussian processes for prediction of homing pigeon flight trajectories. In *AIP Conference Proceedings*, volume 1193, page 360, 2009.

[83] C. Fulgenzi, C. Tay, A. Spalanzani, and C. Laugier. Probabilistic navigation in dynamic environment using rapidly-exploring random trees and Gaussian processes. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 1056–1062. IEEE, 2008.

[84] Peter Trautman and Andreas Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In *Intelligent Robots and Systems (IROS), IEEE International Conference On*, pages 797–803, Taipei, Taiwan, Oct 2010.

[85] Peter Trautman, Jiaxin Ma, Richard M Murray, and Anna Krause. Robot navigation in dense human crowds: the case for cooperation. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 2153–2160. IEEE, 2013.

[86] Sungjoon Choi, Eunwoo Kim, and Songhwai Oh. Real-time navigation in crowded dynamic environments using Gaussian process motion control. In *Robotics and Automation (ICRA), 2014 IEEE International Conference On*, pages 3221–3226. IEEE, 2014.

[87] Georges Aoude, Joshua Joseph, Nicholas Roy, and Jonathan How. Mobile agent trajectory prediction using Bayesian nonparametric reachability trees. *Proc. of AIAA Infotech@ Aerospace*, pages 1587–1593, 2011.

[88] Georges Aoude, Brandon D. Luders, Joshua Joseph, Nicholas Roy, and Jonathan P. How. Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns. *Autonomous Robots*, 35(1):51–76, 2013.

[89] Steven Reece, Richard Mann, Iead Rezek, Stephen Roberts, Ali Mohammad-Djafari, Jean-Franc¸ois Bercher, and Pierre Bessie´ re. Gaussian process segmentation of co-moving animals. In *AIP Conference Proceedings-American Institute of Physics*, volume 1305, page 430, 2011.

[90] Pete Trautman, Jeremy Ma, Richard M Murray, and Andreas Krause. Robot navigation in dense human crowds: Statistical models and experimental studies of human-robot cooperation. *The International Journal of Robotics Research*, 34(3):335–356, 2015.

[91] Emily B Fox, David S Choi, and Alan S Willsky. Nonparametric Bayesian methods for large scale multi-target tracking. In *Signals, Systems and Computers, 2006. ACSSC'06. Fortieth Asilomar Conference on*, pages 2009–2013. IEEE, 2006.

[92] Xiaogang Wang, Keng Teck Ma, Gee-Wah Ng, and W Eric L Grimson. Trajectory analysis and semantic region modeling using nonparametric hierarchical Bayesian models. *International journal of computer vision*, 95(3):287–312, 2011.

[93] Yu Fan Chen, Miao Liu, Shih-Yuan Liu, Justin Miller, and Jonathan P How. Predictive modeling of pedestrian motion patterns with Bayesian nonparametrics. In *AIAA Guidance, Navigation, and Control Conference*, page 1861, 2016.

[94] J. Joseph, F. Doshi-Velez, and N. Roy. A Bayesian nonparametric approach to modeling mobility patterns. In *AAAI*, 2010.

[95] J. Joseph, F. Doshi-Velez, A. S. Huang, and N. Roy. A Bayesian nonparametric approach to modeling motion patterns. *Autonomous Robots*, 31(4):383–400, 2011.

[96] Oene Bottema and Bernard Roth. *Theoretical Kinematics*. Courier Corporation, 1990.

[97] Sean Meyn and Richard L Tweedie. *Markov Chains and Stochastic Stability; 2nd Ed.* Cambridge Mathematical Library. Cambridge Univ. Press, Leiden, 2009.

[98] G J Jeffers, Hoogenboom. Practitioner's handbook on the modelling of dynamic change in ecosystems, 1990.

[99] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *The Journal of Machine Learning Research*, 9:235–284, 2008.

[100] J. L. Ny and G. J. Pappas. On trajectory optimization for active sensing in Gaussian process models. In *Proc. of the IEEE Conference on Decision and Control*, pages 6286 –6292, Shanghai, China, Dec 2009.

[101] H. Wei, W. Lu, and S. Ferrari. An information value function for nonparametric Gaussian processes. In *Proc. Neural Information Processing Systems Conference*, Lake Tahoe, NV, 2012.

[102] H. Wei, W. Lu, P. Zhu, S. Ferrari, R. H. Klein, S. Omidshafiei, and J. P. How. Camera control for learning nonlinear target dynamics via Bayesian nonparametric Dirichlet-Process Gaussian-Process (DP-GP) models. In *Intelligent Robots and Systems, IEEE/RSJ International Conference On*, pages 95–102. IEEE, 2014.

[103] Matthias Seeger, Christopher Williams, and Neil Lawrence. Fast forward selection to speed up sparse Gaussian process regression. In *Artificial Intelligence and Statistics 9*, number EPFL-CONF-161318, 2003.

[104] Yunfei Xu, Jongeun Choi, and Songhwai Oh. Mobile sensor network navigation using Gaussian processes with truncated observations. *Robotics, IEEE Transactions on*, 27(6):1118–1131, 2011.

[105] Neil Lawrence, Matthias Seeger, and Ralf Herbrich. Fast sparse Gaussian process methods: The informative vector machine. In *Proceedings of the 16th Annual Conference on Neural Information Processing Systems*, number EPFL-CONF-161319, pages 609–616, 2003.

[106] Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Advances in neural information processing systems*, pages 1257–1264, 2005.

[107] Joaquin Quinonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate Gaussian process regression. *The Journal of Machine Learning Research*, 6:1939–1959, 2005.

[108] C. E. Rasmussen. Gaussian processes in machine learning. In *Advanced Lectures on Machine Learning*, pages 63–71. Springer, 2004.

[109] Krishna B Athreya and Soumendra N Lahiri. *Measure theory and probability theory*. Springer Science & Business Media, 2006.

[110] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias W Seeger. Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. *Information Theory, IEEE Transactions on*, 58(5):3250–3265, 2012.

[111] Matthias Seeger. Gaussian processes for machine learning. *International Journal of Neural Systems*, 14(02):69–106, 2004.

[112] David Duvenaud. *Automatic Model Construction with Gaussian Processes*. PhD thesis, University of Cambridge, 2014.

[113] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., 2006.

[114] Steven N. MacEachern and Peter Müller. Estimating mixture of Dirichlet process models. *Journal of Computational and Graphical Statistics*, 7(2):223–238, 1998.

[115] David M. Blei and Michael I. Jordan. Variational methods for the Dirichlet process. In *Proceedings of the Twenty-First International Conference on Machine Learning*, page 12. ACM, 2004.

[116] Steven MacEachern and Peter Müller. Efficient mcmc schemes for robust model extensions using encompassing Dirichlet process mixture models. In *Robust Bayesian Analysis*, pages 295–315. Springer, 2000.

[117] Michael D. Escobar. Estimating normal means with a Dirichlet process prior. *Journal of the American Statistical Association*, 89(425):268–277, 1994.

[118] Steven N MacEachern. Estimating normal means with a conjugate style Dirichlet process prior. *Communications in Statistics-Simulation and Computation*, 23(3):727–741, 1994.

[119] Sonia Jain and Radford M. Neal. A split-merge markov chain monte carlo procedure for the Dirichlet process mixture model. *Journal of Computational and Graphical Statistics*, 13(1), 2004.

[120] Kenichi Kurihara, Max Welling, and Nikos A. Welling. Accelerated variational Dirichlet process mixtures. In *Advances in Neural Information Processing Systems*, pages 761–768, 2006.

[121] Kenichi Kurihara, Max Welling, and Yee Whye Teh. Collapsed variational Dirichlet process mixture models. In *IJCAI*, volume 7, pages 2796–2801, 2007.

[122] Chong Wang, John W. Paisley, and David M Blei. Online variational inference for the hierarchical Dirichlet process. In *International Conference on Artificial Intelligence and Statistics*, pages 752–760, 2011.

[123] Yee Whye Teh. Dirichlet process. In *Encyclopedia of machine learning*, pages 280–287. Springer, 2011.

[124] Terence Tao. *An Introduction to Measure Theory*, volume 126. American Mathematical Soc., 2011.

[125] Richard M Dudley. *Real Analysis and Probability*, volume 74. Cambridge University Press, 2002.

[126] C. E. Rasmussen and Z. Ghahramani. Infinite mixtures of Gaussian process experts. *Proc. of Advances in neural information processing systems (NIPS)*, 2:881–888, Dec 2002.

[127] Edward Meeds and Simon Osindero. An alternative infinite mixture of Gaussian process experts.

[128] Chao Yuan and Claus Neubauer. Variational mixture of Gaussian process experts. In *Advances in Neural Information Processing Systems*, pages 1897–1904, 2009.

[129] E. Jackson, M. Davy, A. Doucet, and W. J. Fitzgerald. Bayesian unsupervised signal classification by Dirichlet process mixtures of Gaussian processes. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference On*, volume 3, pages III–1077. IEEE, 2007.

[130] Dilan Görür and Carl Edward Rasmussen. Dirichlet process Gaussian mixture models: Choice of the base distribution. *Journal of Computer Science and Technology*, 25(4):653–664, 2010.

[131] Bolei Zhou, Xiaogang Wang, and Xiaoou Tang. Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2871–2878. IEEE, 2012.

[132] Xiaogang Wang, Kinh Tieu, and Eric Grimson. Learning semantic scene models by trajectory analysis. In *Computer Vision–ECCV 2006*, pages 110–123. Springer, 2006.

[133] Yunfei Xu, Jongeun Choi, Sarat Dass, and Tapabrata Maiti. Sequential Bayesian prediction and adaptive sampling algorithms for mobile sensor networks. *Automatic Control, IEEE Transactions on*, 57(8):2078–2084, 2012.

[134] N. Rao, S. Hareti, W. Shi, and S. Iyengar. Robot navigation in unknown terrains: Introductory survey of non-heuristic algorithms. In *Technical Report ORNL/TM-12410*. Oak Ridge National Laboratory, Oak Ridge, TN, 1993.

[135] H. Wei, W. Ross, S. Varisco, P. Krief, and S. Ferrari. Modeling of human driver behavior via receding horizon and artificial neural network controllers. In *Decision and Control, IEEE Annual Conference On*, pages 6778–6785, Florence, Italy, Dec 2013.

[136] J. Ross and J. Dy. Nonparametric mixture of Gaussian processes with constraints. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 1346–1354, 2013.

[137] J. Hensman, M. Rattray, and N. D. Lawrence. Fast nonparametric clustering of structured time-series. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(2):383–393, 2015.

[138] Zhe Chen. Bayesian filtering: From Kalman filters to particle filters, and beyond. *Statistics*, 182(1):1–69, 2003.

[139] Simo Särkkä. *Bayesian Filtering and Smoothing*, volume 3. Cambridge University Press, 2013.

[140] Jayesh H Kotecha and Petar M Djurić. Gaussian particle filtering. *Signal Processing, IEEE Transactions on*, 51(10):2592–2601, 2003.

[141] Jayesh H Kotecha and Petar M Djurić. Gaussian sum particle filtering. *Signal Processing, IEEE Transactions on*, 51(10):2602–2612, 2003.

[142] David JC MacKay. Introduction to Monte carlo methods. In *Learning in graphical models*, pages 175–204. Springer, 1998.

[143] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley-Interscience, New York, NY, 1991.

[144] W. Lu, G. Zhang, S. Ferrari, R. Fierro, and I. Palunko. An information potential approach for tracking and surveilling multiple moving targets using mobile sensor agents. In *SPIE Defense, Security, and Sensing*, pages 80450T–80450T. International Society for Optics and Photonics, 2011.

[145] W. Lu, G. Zhang, and S. Ferrari. An information potential approach to integrated sensor path planning and control. *Robotics, IEEE Transactions on*, 30(4):919–934, 2014.

[146] K. Kastella. Discrimination gain to optimize detection and classification. *IEEE Transactions on Systems, Man, and Cybernetics-Part A*, 27(1):112–116, 1997.

[147] F. Nobile, R. Tempone, and C. G. Webster. A sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 46(5):2309–2345, 2008.

[148] Alan E. Gelfand, Athanasios. Kottas, and Steven N. MacEachern. Bayesian nonparametric spatial modeling with Dirichlet process mixing. *Journal of the American Statistical Association*, 100(471):1021–1035, 2005.

[149] Robert F. Stengel. *Flight Dynamics*. Princeton University Press, 2015.

[150] W. H. Press. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge university press, 2007.

[151] D. S. Bernstein. *Matrix Mathematics*. Princeton University Press, Princeton, NJ, 2005.

[152] R. E. Caflisch. Monte Carlo and Quasi-Monte Carlo methods. *Acta numerica*, 7:1–49, 1998.

[153] Chun-Wa Ko, Jon Lee, and Maurice Queyranne. An exact algorithm for maximum entropy sampling. *Operations Research*, 43(4):684–691, 1995.

[154] Michael R. Garey and David S. Johnson. *Computers and Intractability*, volume 29. wh freeman, 2002.

[155] Seapahn Megerian, Farinaz Koushanfar, Miodrag Potkonjak, and Mani B. Srivastava. Worst and best-case coverage in sensor networks. *IEEE Transactions on Mobile Computing*, 4(2):84–92, 2005.

[156] Joseph M Kahn, Randy H Katz, and Kristofer SJ Pister. Next century challenges: Mobile networking for "Smart Dust". In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 271–278. ACM, 1999.

[157] Fabrice Bonjean and Gary SE Lagerloef. Diagnostic model and analysis of the surface currents in the tropical Pacific Ocean. *Journal of Physical Oceanography*, 32(10):2938–2954, 2002.

[158] Chris Urmson, Joshua Anhalt, Drew Bagnell, Christopher Baker, Robert Bittner, MN Clark, John Dolan, Dave Duggins, Tugrul Galatali, Chris Geyer, et al. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics*, 25(8):425–466, 2008.

[159] Gill Barequet, Matthew Dickerson, and Petru Pau. Translating a convex polygon to contain a maximum number of points. *Computational Geometry*, 8(4):167–179, 1997.

[160] M. Berg, M. Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry*. Springer, 2000.

[161] R Timothy Marler and Jasbir S. Arora. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6):369–395, 2004.

[162] Achille Messac. From dubious construction of objective functions to the application of physical programming. *AIAA journal*, 38(1):155–163, 2000.

[163] Achille Messac and Christopher A Mattson. Generating well-distributed sets of pareto points for engineering design using physical programming. *Optimization and Engineering*, 3(4):431–450, 2002.

[164] K. A. Proos, G.P. Steven, O.M. Querin, and Y.M. Xie. Multicriterion evolutionary structural optimization using the weighting and the global criterion methods. *AIAA journal*, 39(10):2006–2012, 2001.

[165] Brad Nelson and Pradeep K. Khosla. Integrating sensor placement and visual tracking strategies. In *Experimental Robotics III*, pages 167–181. Springer, 1994.

[166] Thomas Lange Vincent and Walter Jervis Grantham. *Optimality in Parametric Systems*. John Wiley & Sons, 1981.

[167] Yu E. Nesterov and Michael J. Todd. Primal-dual interior-point methods for self-scaled cones. *SIAM Journal on optimization*, 8(2):324–364, 1998.

[168] Fabrice Bonjean and Gary SE Lagerloef. Diagnostic model and analysis of the surface currents in the tropical Pacific Ocean. *Journal of Physical Oceanography*, 32(10):2938–2954, 2002.

[169] Ofir Avni, Francesco Borrelli, Gadi Katzir, Ehud Rivlin, and Hector Rotstein. Scanning and tracking with independent cameras: A biologically motivated approach based on model predictive control. *Autonomous Robots*, 24(3):285–302, 2008.

[170] Janne Heikkilä. Geometric camera calibration using circular control points. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(10):1066–1077, 2000.

[171] Slobodan N. Vukosavic. *Digital Control of Electrical Drives*. Springer Science & Business Media, 2007.

[172] N.R. Gans, G. Hu, and W.E. Dixon. Keeping multiple objects in the field of view of a single ptz camera. In *American Control Conference*, pages 5259–5264. IEEE, 2009.

[173] Mariano Giaquinta and Giuseppe Modica. *Mathematical Analysis: Foundations and Advanced Techniques for Functions of Several Variables*. Springer Science & Business Media, 2011.

[174] Axis Communications. Product comparison tables, network video, 2015.

[175] Hongchuan Wei, Wenjie Lu, Pingping Zhu, Silvia Ferrari, Miao Liu, Robert Klein, Shayegan Omidshafiei, and Jonathan P. How. Information value in nonparametric Dirichlet-Process Gaussian-Process (DPGP) mixture models. *Automatica*, 2015.

[176] Pietro Salvagnini, Federico Pernici, Marco Cristani, Giuseppe Lisanti, Iacopo Masi, Alberto Del Bimbo, and Vittorio Murino. Information theoretic sensor management for multi-target tracking with a single pan-tilt-zoom camera. In *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference On*, pages 893–900. IEEE, 2014.

[177] Alexey S Matveev, Hamid Teimoori, and Andrey V Savkin. A method for guidance and control of an autonomous vehicle in problems of border patrolling and obstacle avoidance. *Automatica*, 47(3):515–524, 2011.

[178] Mathworks. *Matlab Optimization Toolbox*. [Online]. Avaliable: http://www.mathworks.com, 2004. function: fmincon.

[179] Simon Ashton and Peter Sollich. Learning curves for multi-task Gaussian process regression. In *Advances in Neural Information Processing Systems*, pages 1781–1789, 2012.

[180] Manfred Opper and Francesco Vivarelli. General bounds on Bayes errors for regression with Gaussian processes. *Advances in Neural Information Processing Systems*, 11:302–308, 1999.

[181] Peter Sollich. Gaussian process regression with mismatched models. In *Advances in Neural Information Processing Systems*, pages 519–526, 2002.

[182] Matthew Urry and Peter Sollich. Exact learning curves for Gaussian process regression on large random graphs. In *Advances in Neural Information Processing Systems*, pages 2316–2324, 2010.

[183] Peter Sollich and Anason Halees. Learning curves for Gaussian process regression: Approximations and bounds. *Neural computation*, 14(6):1393–1428, 2002.

[184] K Krishna and M Narasimha Murty. Genetic K-means algorithm. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 29(3):433–439, 1999.

[185] Nikolaos Chatzipanagiotis, Darinka Dentcheva, and Michael M Zavlanos. An augmented Lagrangian method for distributed optimization. *Mathematical Programming*, 152(1-2):405–434, 2015.

[186] H. Wei and S. Ferrari. A geometric transversals approach to sensor motion planning for tracking maneuvering targets. *Automatic Control, IEEE Transactions on*, 60(10):2773–2778, Oct 2015.

[187] H. Wei and S. Ferrari. Active sensing for multiple target dynamics learning by a PTZ camera. *Automatica*, 2015.

[188] Pingping Zhu, Hongchuan Wei, Wenjie Lu, and Silvia Ferrari. Multi-kernel probability distribution regressions. In *Neural Networks (IJCNN), 2015 International Joint Conference On*, pages 1–7. IEEE, 2015.

Biography

Hongchuan Wei born in Mishan, China, on January 28th 1988, received a Ph.D in mechanical engineering from Duke University, USA in 2016, a M.S. in computer science from Duke University, USA in 2015, and B.S. degree in vehicle engineering from Tsinghua University, China, 2010. His research focuses on information theory, Bayesian nonparametric models and sensor planning.

He has published [1], [186], [4] on sensor network planning, [135] on human modeling by artificial neural networks, and [101] [102], [175], [187], [188] on information value functions for Bayesian nonparametric models.