Autonomous Sensor Path Planning and Control for Active Information Gathering

by

Wenjie Lu

Department of Mechanical Engineering and Materials Science
Duke University

Date: _____

Approved:

_____
Silvia Ferrari, Supervisor

_____
Michael Zavlanos

_____
Xiaobai Sun

_____
Jerome Reiter

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
2014

Abstract

Autonomous Sensor Path Planning and Control for Active Information Gathering

by

Wenjie Lu

Department of Mechanical Engineering and Materials Science
Duke University

Date: _____

Approved:

_____
Silvia Ferrari, Supervisor

_____
Michael Zavlanos

_____
Xiaobai Sun

_____
Jerome Reiter

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
2014

Abstract

Sensor path planning and control refer to the problems of determining the trajectory and feedback control law that best support sensing objectives, such as monitoring, detection, classification, and tracking. Many autonomous systems developed, for example, to conduct environmental monitoring, search-and-rescue operations, demining, or surveillance, consist of a mobile vehicle instrumented with a suite of proprioceptive and exteroceptive sensors characterized by a bounded field-of-view (FOV) and a performance that is highly dependent on target and environmental conditions and, thus, on the vehicle position and orientation relative to the target and the environment. As a result, the sensor performance can be significantly improved by planning the vehicle motion and attitude in concert with the measurement sequence. This dissertation develops a general and systematic approach for deriving information-driven path planning and control methods that maximize the expected utility of the sensor measurements subject to the vehicle kinodynamic constraints.

The approach is used to develop three path planning and control methods: the information potential method (IP) for integrated path planning and control, the optimized coverage planning based on the Dirichlet process-Gaussian process (DP-GP) expected Kullback-Leibler (KL) divergence, and the optimized visibility planning for simultaneous target tracking and localization. The IP method is demonstrated on a benchmark problem, referred to as treasure hunt, in which an active vision sensor is mounted on a mobile unicycle platform and is deployed to classify stationary targets

characterized by discrete random variables, in an obstacle-populated environment. In the IP method, an artificial potential function is generated from the expected conditional mutual information of the targets and is used to design a closed-loop switched controller. The information potential is also used to construct an information roadmap for escaping local minima. Theoretical analysis shows that the closed-loop robotic system is asymptotically stable and that an escaping path can be found when the robotic sensor is trapped in a local minimum. Numerical simulation results show that this method outperforms rapidly-exploring random trees and classical potential methods. The optimized coverage planning method maximizes the DP-GP expected KL divergence approximated by Monte Carlo integration in order to optimize the information value of a vision sensor deployed to track and model multiple moving targets. The variance of the KL approximation error is proven to decrease linearly with the inverse of the number of samples. This approach is demonstrated through a camera-intruder problem, in which the camera pan, tilt, and zoom variables are controlled to model multiple moving targets with unknown kinematics by nonparametric DP-GP mixture models. Numerical simulations as well as physical experiments show that the optimized coverage planning approach outperforms other applicable algorithms, such as methods based on mutual information, rule-based systems, and randomized planning. The third approach developed in this dissertation, referred to as optimized visibility motion planning, uses the output of an extended Kalman filter (EKF) algorithm to optimize the simultaneous tracking and localization performance of a robot equipped with proprioceptive and exteroceptive sensors, that is deployed to track a moving target in a global positioning system (GPS) denied environment.

Because active sensors with multiple modes can be modeled as a switched hierarchical system, the sensor path planning problem can be viewed as a hybrid optimal control problem involving both discrete and continuous state and control variables.

For example, several authors have shown that a sensor with multiple modalities is a switched hybrid system that can be modeled by a hierarchical control architecture with components of mission planning, trajectory planning, and robot control. Then, the sensor performance can be represented by two Lagrangian functions, one function of the discrete state and control variables, and one function of the continuous state and control variables. Because information value functions are typically nonlinear, this dissertation also presents an adaptive dynamic programming approach for the model-free control of nonlinear switched systems (hybrid ADP), which is capable of learning the optimal continuous and discrete controllers online. The hybrid ADP approach is based on new recursive relationships derived in this dissertation and is proven to converge to the solution of the hybrid optimal control problem. Simulation results show that the hybrid ADP approach is capable of converging to the optimal controllers by minimizing the cost-to-go online based on a fully observable state vector.

To my parents and my wife

Contents

List of Tables

List of Figures

List of Abbreviations and Symbols

Symbols

$\mathcal{A}$	Geometry of sensor platform
$\mathcal{A}_i$	Geometry of ith sensor platform
$\boldsymbol{B}_0$	Index set of obstacles within the distance of influence
$\mathcal{B}_i$	Geometry of ith obstacle, $i = 1, 2, \ldots, n$
$\mathbf{c}_i$	Kernel function of the ith velocity field
$\mathcal{C}$	Configuration space
$\mathcal{C}_{free}$	Free configuration space
$\mathcal{CB}$	C-obstacle region
$\boldsymbol{\chi}$	Mobile agent state
$\mathcal{CT}$	C-target region
d_0	Influence distance
$d_l(\boldsymbol{q})$	Minimum distance from $\boldsymbol{q}$ in the configuration space
$D(p\|\|q)$	Kullback-Leibler divergence between p and q
$D_\alpha(p\|\|q)$	α-divergence between p and q
δt	Time step
$\boldsymbol{\eta}_{jis}(k+1^-)$	Predicted mean of the jisth particle
$\mathbf{e}_i$	Random variable for environmental conditions of the ith target
$\mathbb{E}$	Expectation operator
ϵ	GP RMS

$\xi(k)$	Discrete mode
$\boldsymbol{\epsilon}(k)$	Measurement history up to time k
$\boldsymbol{\epsilon}$	Largest invariant set, discrete mode set
$\mathbf{f}$	Kinematic function
$\mathbf{f}_i$	ith velocity field
$\mathbf{f}_{\nu(k)}$	System continuous kinematic function given discrete control $\nu(k)$
$\mathcal{F}$	Set of velocity fields
$D(\tau)$	Scaled Euclidian distance along path τ
$\mathcal{F}_{\mathcal{A}}$	Moving Cartesian frame embedded in $\mathcal{A}$
$\mathcal{F}_{\mathcal{A}_i}$	Moving Cartesian frame embedded in $\mathcal{A}_i$, $i = 1, 2, \ldots, r$
$\mathcal{F}_{\mathcal{W}}$	Cartesian frame embedded in $\mathcal{W}$
H_ν	Hamiltonian given discrete control ν
$H(X)$	Shannon entropy of X
$H_{R_\alpha}(X)$	Rény's entropy of order α
$I(X;Y \mid Z)$	Conditional mutual information between X and Y given Z
I_A	Index set of A
I_B	Index set of B
I_T	Index set of T
J	Objective function
$k_1 \quad k_0$	Constant parameter
K_p	Constant parameter
$l(t)$	Zoom level at time t
L	Number of points of interest
$\mathcal{L}_\nu$	Lagrangian given discrete control ν
$\boldsymbol{\Lambda}_{jis}(k+1^-)$	predicted covariance of the jisth particle
$\mathcal{L}$	Zoom level set

$\boldsymbol{\lambda}$	Gradient of value function with respect to continuous state
M	Number of velocity fields
m_i^j	Weight for direction sample function
$\mathcal{M}^k$	Set of measurements up to time k
M_i	Set of measurements on $\mathcal{T}_i$
$\mathbf{m}(k)$	Measurement vector
$\boldsymbol{\mu}_j(k)$	Estimated mean of the ith velocity field at time k
$\mathcal{N}$	Gaussian distribution
$N(j)$	The index of targets assigned to the jth robotic sensor
$M, N(t)$	Number of agents
$N_r(\boldsymbol{q}_i)$	Set of neighbors for ith sensor
N_{GPR}	Number of targets correctly classified by GPR measurements
N_{IR}	Number of targets correctly classified by IR measurements
NN_λ^ξ	Critic network
NN_c^ξ	Actor network
$\nu(k)$	Discrete control at time k
ω	Angular velocity
ω_{jis}	Weight of the jisth particle
$p(x)$	Probability density function or probability mass function for random variable X
$\boldsymbol{\pi}$	Vector of probability mass
π	Policy
$\phi(\mathbf{x})$	Terminal cost
$\boldsymbol{\phi}$	Random variables of interest
$\hat{\psi}$	Information value function
$\mathbf{P}_i(k)$	Position measurement history for the ith velocity field up to time k

$\mathcal{P}_{ji}(k)$	Set of particles
$q(x)$	Probability density function or Probability mass function for random variable X
$\mathbf{q}$	Robot configuration
$\mathbf{q}_0$	Robot initial configuration
$\mathbf{q}_i$	The ith robot configuration
$\boldsymbol{R}_0$	Index set of robots within the distance of influence
T	Set of targets $T = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_m\}$
$\mathcal{T}_i$	Geometry of ith target, $i = 1, 2, \ldots, m$
t_f	Final time
τ	Robotic sensor path
$\mathcal{S}$	Geometry of accurate sensor FOV
$\mathcal{S}_i$	Geometry of the ith sensor FOV, $i = 1, 2, \ldots, r$
$\mathcal{S}(\mathbf{q})$	Set of points in $\mathcal{W}$ occupied by $\mathcal{S}$ at configuration $\mathbf{q}$
$\mathbf{\Sigma}_j(k)$	Estimated covariance of the ith velocity field at time k
$Tr(t_k)$	Rapidly-explore random tree at time t_k
$\mathcal{U}$	Space of admissible control inputs
$\mathbf{u}(t)$	Configuration of sensor at t
$\mathbf{u}_\nu$	Continuous controller given discrete control ν
$U(\mathbf{q})$	Potential function
$U^j(\mathbf{q})$	Potential function for the jth robotic sensor
$U_{att}(\mathbf{q})$	Attractive potential function
$U^j_{att}(\mathbf{q})$	Attractive potential function for the jth robotic sensor
$U^j_r(\mathbf{q})$	Repulsive potential function for the jth robotic sensor by other robotic sensors
$U_{rep}(\mathbf{q})$	Repulsive potential function
V	Value function

V_i	Information value of the ith target
$\mathbf{V}_i(k)$	Velocity measurement history for the ith velocity field up to time k
$\mathcal{V}$	Lyapunov function
V_i	Information value ith garget
$\mathcal{W}$	Geometry of workspace
W_1	Number of targets correctly classified after measurements
W_0	Number of targets correctly classified before measurements
w_{ij}	Probability of the ith target following the jth velocity
$\mathbf{x}$	Continuous system state
X_i	Random variable for ith cell state, $i = 1, \ldots, c$
$\boldsymbol{\xi}_i$	Position vector of $\boldsymbol{q}_i$ in the configuration space
$\mathbf{X}$	Set of points of interest
$\mathcal{X}$	Finite range for X
X^k or $\mathbf{X}^k$	Random variable for cell state at time step k
Z_i or $\mathbf{Z}_i$	Measurement at the ith target, $i = 1, \ldots, c$
$\mathbf{z}^k$	Measurement at time step k
z^k	Observation of measurement at time step k
ϵ	Constant number
ε	An area with a deterministic size by the user based on the size of potential field.
θ	Parameter of sensor model distribution, or robotic sensor heading angle
$\boldsymbol{\theta}$	Random variables representing target characteristics
κ_i	ith cell, $i = 1, 2, \ldots, c$
Λ	Sample space of λ^k
$\boldsymbol{\Lambda}_i$	Random variable for environmental conditions for ith tagret

μ	The parameter matrix in the growth curve model
μ_{ij}	The element at ith row jth column of matrix μ
μ_i^j	Mean for direction sample function
$\rho_{goal}(\mathbf{q})$	The distance between $\mathbf{q}$ and the goal
$\rho_i^b(\mathbf{q})$	The distance between $\mathbf{q}$ and the ith obstacle
$\rho_i^t(\mathbf{q})$	The distance between $\mathbf{q}$ and the ith target

Abbreviations

ADP	Adaptive dynamic programming
BN	Bayesian network
BNP	Non-parametric Bayesian Model
DP	Dirichlet process
DP-GP	Dirichlet process Gaussian process
EDG	Expected discrimination gain
EER	Expected entropy reduction (conditional mutual information)
EKL	Expected Kullback-Leibler
FOV	Field-of-view
GP	Gaussian process
GPR	Ground-penetrating radar mounted
IIP	Intermediate informative prior
IP	Information potential
IRM	Information roadmap method
KL	Kullback-Leibler
LQ	Linear-quadratic
LIP	Less informative prior
MCMC	Markov Chain Monte Carlo

MI	Mutual Information
MIP	More informative prior
MPD	Motion planning with different constraints
MSP	Master-slave procedure
PDF	Probability density function
PMF	Probability mass function
PRMs	Probabilistic roadmap methods
RMS	Root mean square
PF	Potential field
PSG	Player/Stage/Gazebo
RRT	Rapidly-exploring random trees
STP	Switching table procedure
UAV	Unmanned air vehicle
UGV	Unmanned ground vehicle
UB	Upper bound

Acknowledgements

First, I would like to thank my advisor, Dr. Silvia Ferrari. Your kindness, help, support, and dedication to me were invaluable. Your consistent motivation kept me energized every day, and your academic adventure and experience enhances my interests in research.

I would also like to give special mention to those faculties who have served their time and effort: Dr. Michael Zavlanos, Dr. Xiaobai Sun, Dr. Jerry Reiter. In addition, I would like to thank those who have contributed to my better understanding of this research: Dr. Rafael Fierro, Dr. Tom Wettergren, Dr. Devendra Garg, Dr. Krish Chakrabarty. A special thank you to my great labmates and fellow graduate students, who were always available and willing to help along the way: Guoxian Zhang, Gianluca Di Muro, Greg Foderaro, Greyson Daugherty, Brian Bernard, Ashleigh Swingler, Keith Rudd, Hongchuan Wei, Xu Zhang, Wess Ross, Pingping Zhu, Vikram Raju, Hersh Tapadia.

I wish to dedicate this dissertation to my family. To my parents, Yongshi and Xinghua, my sister and brother in law, Wenjuan and Haojun. Thanks for having provided me a fantastic environment. Your passion and love give me the power and confidence. To my wife and nushen, Amanda. This will never happen without your support, understanding, and love.

1

Introduction

Autonomous sensor control for active information gathering utilizes information theoretic functions to assess the value of sensor measurements as a function of sensor control inputs, random environment variables, and unknown target variables. Subsequently, the expected value of the information function can be optimized with respect to the sensor mode, the measurement sequence, or the position and orientation of FOV [19, 117, 39, 40, 112]. As a result, the effectiveness of autonomous sensor systems can be greatly improved in a variety of applications, including mine hunting [85, 80]; classification and tracking [39, 40]; and the monitoring of urban environments [28], underwater objects [33], manufacturing plants [22], and endangered species [44]. Furthermore, in many sensor applications, such as monitoring, maintenance, or surveillance, the set of all measurements that could be acquired by a sensor significantly exceeds its available power, time, and computational capabilities [22], such that it is also desirable to minimize distance traveled or energy consumption. Then, the sensor controller can be designed to account for the FOV geometry and the robot kinodynamic constraints, such that the sensor configurations that enable the most informative measurements with the minimum energy can

be determined [111, 100]. Thus, in this dissertation, the sensor is viewed as an information-gathering agent that must make decisions on its configuration (position, orientation, and mode), in order to maximize the sensor performance and minimize the robot energy consumption.

A key challenge in sensor path planning is to assess the sensor performance that will result from the sensor decisions before obtaining the future sensor measurements [16, 19, 117]. The sensor performance can be shown to depend on the amount of information or conversely on the uncertainty associated with a set of unknown target variables to be inferred from repeated sensor measurements. Thus, the utility of future measurements may be represented by their expected information value conditioned on the prior measurements and on the environmental variables. Information theoretic value functions can be used to quantify the amount of information associated with the probabilistic model of one or more unknown random variables. The uncertainty of the random variables can then be minimized by optimizing the information value functions [101, 51, 52, 86, 93]. Computing information theoretic functions for one or more random variables in a stochastic process requires knowledge of their joint probability mass (or density) functions. Because the posterior belief state in the sensor planning problem is typically unknown, a general approach was recently presented for estimating the *expected* information value of the future sensor measurements, where the expectation is with respect to the future measurements [23].

In Chapter 3, a systematic approach for estimating information theoretic functions for future sensor measurements is reviewed [111]. Several information value functions have been proposed in the literature to measure the information value in sensor planning and management problems. Relative entropy was used to solve a multisensor-multitarget assignment problem in [82]. The expected discrimination gain (EDG) derived from the Kullback-Leibler (KL) divergence was used to manage

agile sensors with Gaussian models for target detection and classification in [45]. Recently, mutual information for sensor planning was studied in multi-target detection, classification, and feature inference by ground-penetrating radars and infrared sensors in [16, 112]. In [23], an approach based on mutual information was also presented for adjusting the configuration of a camera in an object recognition application. In Chapter 3, the approach taken from [111] is extended to develop a new information value function for the DP-GP models based on the expected KL divergence. This new information value function quantifies the expected utility associated with future measurements for updating the current DP-GP mixture model and is defined as the expectation of KL divergence between the current (prior) and posterior DP-GP models over future measurements given sensor control inputs, in situations where discretization is not feasible due to high computational complexities. The DP-GP mixture model provides the necessary flexibility to capture spatial phenomena from data without overfitting [43].

Because robot motion planning approaches deal with the intersections of discrete geometric objects that are possibly moving, subject to a kinematic or dynamic equation, many sensor path planning methods are inspired by existing robot motion planning approaches. Chapter 4 reviews three existing sensor motion planning approaches originally presented in [15], [113], and [66]: the information cell decomposition approach, the information probability roadmap deploy (IPD), and the rapidly exploring random information trees (RRIT) approach. However, these existing methods cannot be implemented in sensor planning when sensor kinodynamic constraints are considered, the targets are moving and their model is complex, or the proprioceptive and exteroceptive sensor are deployed, respectively. To this end, in Chapter 4, three sensor planning methods are developed, which are the information potential method (IP) for integrated path planning and control, the optimized coverage planning based on the DP-GP expected KL divergence, and the optimized visibility

planning for simultaneous target tracking and localization. The IP method is demonstrated on a benchmark problem, referred to as treasure hunt, in which an active vision sensor is mounted on a mobile unicycle platform and is deployed to classify stationary targets characterized by discrete random variables, in an obstacle-populated environment. In the IP method, an artificial potential function is generated from the expected conditional mutual information of the targets and is used to design a closed-loop switched controller. The information potential is also used to construct an information roadmap for escaping local minima. Mutual information is a measure of the information contained in one random variable about another random variable, and the information value is represented by a conditional mutual information function that is developed in [27].

Although many potential navigation functions have been developed for robot motion planning, they are not applicable to sensor path planning because they do not take into account the geometries of the targets or sensor FOV, nor do they consider the target information value [77, 87, 47, 91, 53]. Additionally, the effectiveness of classical potential field methods is limited by their inability to escape local minima, their lack of stabilization, and their inability to enter narrow passages [50]. In this dissertation, a switched control approach based on switched potentials [76] is used to integrate sensor path planning and control. The resulting switched controller can be proven to be asymptotically stable and is guaranteed to converge to the target with the highest information value. Additionally, the same information potential function is utilized to build a local roadmap for escaping local minima. The numerical simulation results show that the IP controlled robotic sensor is capable of obtaining measurements from the most valuable targets, entering narrow passages, and escaping local minima, while avoiding obstacles online. Numerical simulation results also show that this method outperforms rapidly-exploring random trees and classical potential methods.

The above sensor planning approaches typically assume that a measurement is obtained if the sensor FOV intersects with the geometry of a stationary target. In the problem of monitoring moving targets, however, the locations of these targets are unknown and are estimated by time-varying probability density functions (PDFs). Therefore, it is difficult to formulate a target with rigid geometry. The next two approaches are designed to obtain measurements from moving targets. The optimized coverage planning approach maximizes the DP-GP expected KL divergence (approximated by Monte Carlo integration) in order to obtain optimal sensor control for learning a DP-GP model. This type of problems is derived from problems that require learning spatial phenomena, such as a temperature function over a given workspace, a set of velocity fields, or mappings between two spaces. For example, the camera intruder problem requires determining the optimal camera control to collect informative measurements of targets for estimating unknown target kinematics, which are modeled with a DP-GP mixture. The objective is to maximize the estimation accuracy of the learned target kinematics, i.e., the accuracy of the DP-GP mixture model. By assuming the camera's position is fixed and its FOV is a free-flying object without rotation, the optimized coverage planning is designed to generate the sensor control by maximizing the DP-GP expected KL divergence at each time step. The proposed DP-GP expected KL divergence can be approximated via Monte Carlo integration, and the variance of the KL approximation error is proven to decrease linearly with the inverse of the number of samples. Numerical simulations as well as physical experiments show that the optimized coverage planning approach outperforms other applicable algorithms, such as methods based on mutual information, rule-based systems, and randomized planning.

The optimized visibility motion planning approach uses an extended Kalman filter (EKF) to simultaneously track the target and localize the robotic sensor [120, 118, 121]. Within this estimation framework, a controller is derived analytically by

assuming the FOV of the exteroceptive sensor can be approximated by a sector with a fixed orientation and a fixed aperture with respect to the robot. The proposed optimized visibility approach is applicable to any robot equipped with exteroceptive sensors, such as a laser scanner or camera, for tracking and localizing moving targets, and with proprioceptive sensors, such as an odometer, for providing ego-motion information. The results show that the proposed method is effective at tracking and localizing a moving target with low target loss rates and outperforms a state-of-the-art potential method.

Because active sensors with multiple modes can be modeled as a switched hierarchical system, the sensor path planning problem can be viewed as a hybrid optimal control problem involving both discrete and continuous state and control variables. For example, several authors have shown that a sensor with multiple modalities is a switched hybrid system that can be modeled by a hierarchical control architecture with components of mission planning, trajectory planning, and robot control. Then, the sensor performance can be represented by two Lagrangian functions, one function of the discrete state and control variables, and one function of the continuous state and control variables. Because information value functions are typically nonlinear, this dissertation also presents an adaptive dynamic programming approach for the model-free control of nonlinear switched systems (hybrid ADP), which is capable of learning the optimal continuous and discrete controllers online.

A switched hybrid system consists of time-driven and event-driven kinematics. Event-driven kinematics are described by discrete states and control inputs that are represented by finite alphabets. Time-driven kinematics (differential or difference equations) are used to represent systems with continuous states and control in a Euclidean space. The optimal control of switched systems seeks to determine an optimal discrete controller that decides the system mode (event-driven kinematics) and multiple optimal continuous controllers that regulate the system motion (time-driven

kinematics) given the system mode. The discrete and continuous optimal controllers are determined such that a scalar objective function of the hybrid system states and control is minimized over a period of time [12]. Switched hybrid systems have been used to model centralized multi-agent networks in [29, 34] and decentralized multi-agent networks in pursuit-evasion games in [110]. The optimality conditions for the optimal control of switched systems are derived in [78] using the Pontryagin Minimum Principle [79]. A master-slave procedure (MSP) that provides an open-loop solution for a given initial state and a switching table procedure (STP) based on dynamic programming were developed in [83] for a switched affine system with a piecewise quadratic cost function. A parametric-optimization method was proposed in [109] to optimize the continuous controller and switching time instants for a given (predesigned) fixed switching sequence. However, these existing approaches cannot adapt to the uncertainty in system modeling and environment.

Adaptive Dynamic Programming (ADP) is an effective approach for solving nonlinear optimal control problems in the absence of a dynamic model in closed form that optimizes the performance in the face of unforeseen changes and uncertainties [31, 73, 75, 84, 59]. In recent decades, ADP has been implemented in a number of applications involving the optimal control of systems described by continuous state and control variables [107, 98, 97, 106, 72, 41, 24, 3, 74] or by stochastic systems characterized by discrete-event state and decisions [99, 7]. ADP has also been used to determine model-free optimal controllers for zero-sum multi-agent games in [115, 61, 96, 94] and for affine nonlinear systems in [62, 114]. Despite its successful implementation and guaranteed convergence [107, 1, 97], the applicability of ADP to hybrid systems has yet to be fully demonstrated in the literature.

In the remainder of this dissertation, a general hybrid ADP approach is presented for the model-free control of switched sensor systems, where the optimal continuous and discrete controllers are determined via online learning. The proposed approach is

based on new recursive relationships derived in this dissertation and is proven to converge to the solution of the hybrid optimal control problem. Simulation results show that the hybrid ADP approach is capable of converging to the optimal controllers by minimizing the cost-to-go online based on a fully observable state vector.

This dissertation is organized as follows. Chapter 2 presents the autonomous sensor planning and control problem for active information gathering. The proposed DP-GP expected KL information function is introduced, after a review on information value functions, in Chapter 3. Three proposed sensor motion planning approaches are presented in Chapter 4, and the results from the numerical simulations are summarized in Chapter 5. In Chapter 6, the optimal planning problem for a sensor with multiple modalities is generalized into an optimal control problem of a switched hybrid system and a proposed hybrid ADP approach is introduced to solve this problem. Finally, the conclusions are given in Chapter 7.

2

Problem Formulation and Assumptions

This dissertation considers the problem of autonomous sensor control for active information gathering with the objective of decreasing the uncertainty of one or more random variables. The sensor performance can be greatly enhanced by using control and data processing methods that consider the sensors, platforms, and their environment as an integrated sensor system. The sensor or sensor network is deployed in a workspace $\mathcal{W} \in \mathbb{R}^d$, where d is either 2 or 3. The workspace is populated with M rigid targets that are denoted by $\mathcal{T}_1, \ldots, \mathcal{T}_M$, where $\mathcal{T}_i$ is a compact subset of $\mathcal{W}$. The set of targets is represented by an index set I_T. Each target $\mathcal{T}_i$, where $i \in I_T$, is characterized by a random state vector $\boldsymbol{\theta}_i$, which may vary with time. For example, $\boldsymbol{\theta}_i$ can be used to denote the position and velocity of a target $\mathcal{T}_i$ that is moving. Let $\boldsymbol{\phi}$ denote the vector consisting of the random variables whose values are to be inferred or classified by the sensor. The vector $\boldsymbol{\phi}$ is also referred to as the random vector of interest. The vector could simply be a vector consisting only of $\boldsymbol{\theta}_i, i \in I_T$, or it could be any other unknown vector relating to the targets' characteristics, measurements, or environment conditions through a model (e.g., the DP-GP model).

Let a compact set $\mathcal{S} \in \mathbb{R}^d$ denote the geometry of the sensor FOV and $\mathcal{F}_{\mathcal{W}}$ denote

a fixed Cartesian frame with origin $\mathcal{O}_{\mathcal{W}}$ and embedded in $\mathcal{W}$, and let $\mathcal{F}_{\mathcal{S}}$ denote a moving Cartesian frame with origin $\mathcal{O}_{\mathcal{S}}$ and embedded in $\mathcal{S}$. Assuming $\mathcal{S}$ is rigid, the FOV configuration $\mathbf{q} = [x\ y\ \theta]^T$ can be used to specify the position and orientation of every point in $\mathcal{S}$ with respect to $\mathcal{F}_{\mathcal{W}}$, where x, y, and θ are the coordinates and orientation of $\mathcal{F}_{\mathcal{S}}$ with respect to $\mathcal{F}_{\mathcal{W}}$. Let $\mathcal{C}$ denote the configuration space and $\mathcal{S}(\mathbf{q})$ denote the subset of $\mathcal{W}$ occupied by $\mathcal{S}$ for configuration $\mathbf{q}$; this subset represents the set of all points in $\mathcal{W}$ that can be measured by the sensor when the sensor FOV is at $\mathbf{q}$. It then follows that the sensor can obtain measurements $\mathbf{z}_i$ from a target $\mathcal{T}_i$ *iff* $\mathcal{T}_i \cap \mathcal{S}(\mathbf{q}) \neq \varnothing$.

For a mobile robotic sensor, the geometry of its platform is described by a rigid object, $\mathcal{A}$, that is a compact subset of a workspace $\mathcal{W} \subset \mathbb{R}^d$. Assuming $\mathcal{A}$ is fixed with respect to $\mathcal{S}$, let $\mathcal{A}(\mathbf{q})$ denote the subset of $\mathcal{W}$ occupied by $\mathcal{A}$ at configuration $\mathbf{q}$. Here, $\mathcal{A}$ must be considered during planning in order to avoid collisions with any of the N rigid obstacles, which are denoted by $\mathcal{B}_1, \ldots, \mathcal{B}_N$. An example of a mobile robotic sensor is shown in Fig. 2.1. The following definitions are then adopted from [16] and [55] :

Definition 1 (C-target)**.** *The target $\mathcal{T}_i$ in $\mathcal{W}$ maps in the robot's configuration space $\mathcal{C}$ to the C-target region $\mathcal{CT}_i = \{\mathbf{q} \in \mathcal{C} \mid \mathcal{S}(\mathbf{q}) \cap \mathcal{T}_i \neq \varnothing\}$.*

Definition 2 (C-obstacle)**.** *The obstacle $\mathcal{B}_i$ in $\mathcal{W}$ maps in the robot's configuration space $\mathcal{C}$ to the C-obstacle region $\mathcal{CB}_i = \{\mathbf{q} \in \mathcal{C} \mid \mathcal{A}(\mathbf{q}) \cap \mathcal{B}_i \neq \varnothing\}$.*

The configuration of FOV $\mathbf{q}$ is usually controlled by an mechatronical system. For example, when the sensor is mounted on a mobile platform and fixed with respect to the platform, the change of the FOV configuration is realized by controlling the mobile platform, perhaps through the linear acceleration and angular velocity of the platform. Therefore, the configuration of FOV $\mathbf{q}$ must also satisfy certain kinematic equation imposed by the mechatronical system. This kinematic equation can be

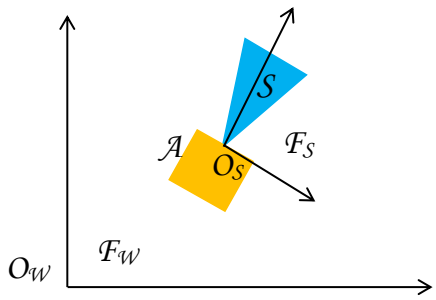


FIGURE 2.1: A robotic sensor with vehicle geometry $\mathcal{A}$ and sensor FOV $\mathcal{A}$ ($d = 2$).

described by

$$\dot{\mathbf{q}}(t) = f_{\nu(t)}[\mathbf{q}(t), \mathbf{u}(t)], \tag{2.1}$$

where $\mathbf{u}(t)$ is the control input and $\nu(t) \in \mathcal{E}$ is the discrete mode control. Note that this kinematic equation can be relaxed when the FOV is modeled as a free-flying object [55].

The target state $\boldsymbol{\theta}_i$, the sensor measurement $\mathbf{z}_i$, and the sensor parameter/mode or environmental condition $\mathbf{e}_i$ are assumed to be continuous random vectors. Then, the measurement process can be described by a joint PDF that, typically, can be factorized as

$$p(\boldsymbol{\theta}_i, \mathbf{z}_i, \mathbf{e}_i) = p(\mathbf{z}_i \mid \boldsymbol{\theta}_i, \mathbf{e}_i) p(\boldsymbol{\theta}_i) p(\mathbf{e}_i), \tag{2.2}$$

because both $\boldsymbol{\theta}_i$ and $\mathbf{e}_i$ are independent random vectors. The conditional PDF of $\mathbf{z}_i$ given $\boldsymbol{\theta}_i$ and $\mathbf{e}_i$ is usually referred to as the sensor model. This PDF is obtained from the physical principles characterizing the measurement process. The PDFs of $\boldsymbol{\theta}_i$ and $\mathbf{e}_i$, known as *priors*, are computed from available prior environmental information. These PDFs are assumed to be uniformly distributed if no prior information is available. Various sensors have been modeled by (2.2), either from first principles or from data [32, 17]. A sensor measurement $\mathbf{z}_i$ is obtained when the sensor FOV intersects the ith target geometry, i.e., $\mathcal{T}_i \cap \mathcal{S}(\mathbf{q}) \neq \varnothing$ under condition $\mathbf{e}_i$, where $\mathbf{e}_i$ is assumed to be known for simplicity.

If the unknown vector of interest $\boldsymbol{\phi}$ only consists of $(\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_M)$ and the targets are independent and time invariant, then the original problem is reduced to inferring $\boldsymbol{\theta}_i$ for all i in I_T. The posterior PDF of $\boldsymbol{\theta}_i$ can be obtained from the measurement model (2.2) and Bayes' rule,

$$p(\boldsymbol{\theta}_i \mid \mathbf{z}_i, \mathbf{e}_i) = \frac{p(\mathbf{z}_i \mid \boldsymbol{\theta}_i, \mathbf{e}_i)\ p(\boldsymbol{\theta}_i)}{\int_{\boldsymbol{\Theta}} p(\mathbf{z}_i \mid \boldsymbol{\theta}_i, \mathbf{e}_i)\ p(\boldsymbol{\theta}_i) d\boldsymbol{\theta}_i}, \tag{2.3}$$

for $i = 1, \ldots, M$, where $\boldsymbol{\Theta}$ is the value space of $\boldsymbol{\theta}_i$ because (2.2) holds for all targets. In other cases, a statistical model is required to build the relation between $\boldsymbol{\phi}$ and $\mathbf{z}_i$ for all i in I_T. For example, as shown in Chapter 3, the DP-GP mixture model assumes that $\boldsymbol{\phi}$ and $\mathbf{z}_i$ for all i in I_T follow an unknown Gaussian process. The DP-GP model is used to infer $\boldsymbol{\phi}$ and to design an information value function.

The problem considered in this dissertation is to determine the optimal control for the autonomous sensor such that (I) the uncertainty of $\boldsymbol{\phi}$ is minimized, (II) the energy consumption or distance traveled is minimized, and (III) collisions with all obstacles are avoided for autonomous sensors with mobile platforms. A block diagram of this sensor system is summarized in Fig. 2.2. The sensor can be viewed as an information-gathering agent that must make decisions on its configuration (position, orientation, and mode), in order to optimize the sensor performance. The sensor performance prior to obtaining the sensor measurements depends on the amount of information, or lack thereof, associated with these variables and is quantified by information value functions. A systematic approach for generating information value functions is reviewed in Chapter 3. These information value functions are used to design motion planning approaches in Chapter 4.

Three types of problems that cannot be solved by existing sensor planning methods are now presented, which will be separately used to demonstrate that the approaches developed in Chapter 4. In the first problem, the sensor is mounted on a

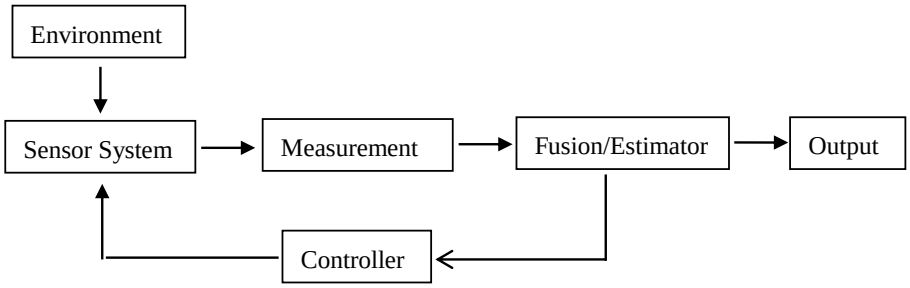


FIGURE 2.2: Block diagram of autonomous sensor control

mobile platform and is deployed in a workspace populated with position-fixed targets and obstacles. This problem is referred to as the treasure hunt problem. The second problem is referred to as the camera intruder problem. This problem involves a position-fixed camera monitoring multiple moving targets to learn unknown target kinematics, where the camera FOV can only cover a portion of the entire workspace at any given time. This problem assumes that the FOV is a free-flying object. The third problem involves one mobile robotic sensor with a bounded FOV tracking a moving target where GPS is unavailable. Thus, the mobile robotic sensor is required to not only optimally track the moving target but also localize itself.

2.1 Problem 1: Mobile Sensor Planning for Target Classification

This problem considers integrated navigation and control for a robotic sensor to classify multiple targets in an obstacle-populated environment. The robotic sensor consists of an unmanned ground vehicle (UGV) equipped with an on-board sensor. As schematized in Fig. 2.3, the sensor FOV, denoted by $\mathcal{S} \subset \mathbb{R}^3$, is defined as a compact subset of $\mathcal{W}$ from which the robot can obtain sensor measurements. The configuration vector $\mathbf{q}$ must also satisfy the robot kinematic equation that, in this

dissertation, is given by the unicycle model in four-dimensional phase space [57],

$$\dot{\boldsymbol{\chi}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ w \\ a \end{bmatrix} = \mathbf{f}(\boldsymbol{\chi}, \mathbf{u}) \tag{2.4}$$

where $\boldsymbol{\chi} \in \mathbb{R}^4$ is the robot state, v is the linear velocity, w is the angular velocity, and a is the linear acceleration. The robot control vector is $\mathbf{u} = [u_1 \; u_2]^T = [a \; w]^T \in \mathcal{U} \subset \mathbb{R}^2$, where $\mathcal{U}$ represents the space of admissible control inputs.

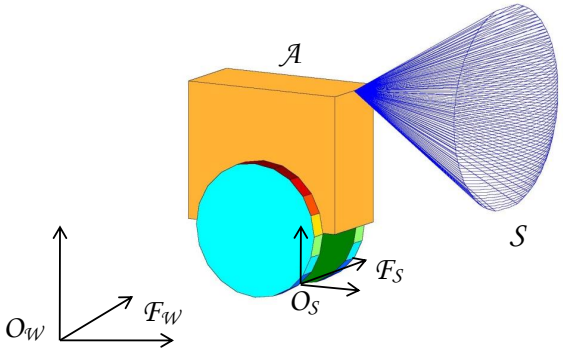


FIGURE 2.3: A robotic sensor with vehicle geometry $\mathcal{A}$ and sensor FOV $\mathcal{A}$.

The target state $\boldsymbol{\theta}_i$, the sensor measurement $\mathbf{z}_i$, and the sensor parameter/mode or environmental condition $\mathbf{e}_i$ are assumed to be discrete random vectors. Note that, in this problem, the random vector of interest $\boldsymbol{\phi}$ in Chapter 2 is given as $\boldsymbol{\phi} = [\boldsymbol{\theta}_1^T, \ldots, \boldsymbol{\theta}_M^T]^T$. The problem is to plan the path and control for the robotic sensor in (2.4) such that (I) the uncertainty of $\boldsymbol{\phi}$ is minimized, (II) the energy consumption or distance traveled is minimized, and (III) no collisions with any obstacles occur. Additionally, in order to reduce the algorithm complexity, it is also assumed that $\mathcal{A}$ is a right prism with the base face adjacent to the xy-plane and that $\mathcal{S}$ is a three-dimensional cone. Obstacles and targets are assumed to be right prisms with their base faces parallel to the xy-plane. Chapter 4 presents an IP integrated path planning and control method that achieves (I)-(III) and guarantees asymptotic closed-loop stability for the robotic sensor.

2.2 Problem 2: Camera Control for Learning Nonlinear Target Kinematics

This problem consists of determining the control input, denoted by $\mathbf{u}$, for a camera with two possible FOV zoom levels, $\mathcal{L} = \{1, 2\}$, to monitor a two-dimensional convex workspace, $\mathcal{W} \subset \mathbb{R}^2$. The first zoom level enables the sensor to measure a small area with high accuracy, and the second zoom level enables the sensor to observe a larger area with less accuracy. As illustrated in Fig. 2.4, if the position of $\mathcal{O}_s$ with respect to $\mathcal{F}_w$ is denoted by $\mathbf{q}(t) \in \mathcal{W}$ and the FOV is assumed to translate in $\mathcal{W}$ without rotation as a free-flying object, the control vector that fully determines the configuration of the sensor FOV is $\mathbf{u}(t) = [\mathbf{q}^T(t)\ l(t)]^T$, where $l(t) \in \mathcal{L}$ denotes the choice of zoom level. The set of points covered by the FOV at time t is denoted by $\mathcal{S}[\mathbf{u}(t)] \subset \mathcal{W}$. Then, at any time t, the noisy vector measurement of the jth target position, $\mathbf{x}_j(t) \in \mathcal{W}$, and velocity, $\dot{\mathbf{x}}_j(t) \in \mathbb{R}^2$,

$$\mathbf{m}_j(t) \triangleq \begin{bmatrix} \mathbf{y}_j(t) \\ \mathbf{z}_j(t) \end{bmatrix} = \begin{bmatrix} \mathbf{x}_j(t) + \mathbf{n}_x \\ \dot{\mathbf{x}}_j(t) + \mathbf{n}_v \end{bmatrix}, \begin{matrix} \mathbf{n_x} \sim \mathcal{N}\{\mathbf{0}, \mathbf{\Sigma}_x[l(t)]\} \\ \mathbf{n_v} \sim \mathcal{N}\{\mathbf{0}, \mathbf{\Sigma}_v[l(t)]\}, \end{matrix} \tag{2.5}$$

for $j = 1, \ldots, M(t)$, is obtained *iff* $\mathbf{x}_j(t) \in \mathcal{S}[\mathbf{u}(t)]$, where $M(t)$ is the number of targets that have entered the workspace up to time t and "$\sim$" denotes "is distributed as". Velocity measurements are obtained though target position difference in two consecutive video frames. When $\mathbf{x}_j(t) \notin \mathcal{S}[\mathbf{u}(t)]$, the measurement of target j at t is an empty set, i.e., $\mathbf{m}_j(t) = \varnothing$. Additionally, it is assumed that data-target association is perfect. The covariance of noise vector $\mathbf{n_x}$ ($\mathbf{\Sigma}_x$) is assumed to be $\mathbf{0}$, and $\mathbf{n_v}$ is assumed to be normally distributed with zero mean and zero covariance; thus, the covariance matrix $\mathbf{\Sigma}_v \in \mathbb{R}^{2\times 2}$ has zero off-diagonal entries. For the two zoom levels, $\mathbf{\Sigma}_v(1) \prec \mathbf{\Sigma}_v(2)$, where $\prec$ denotes an element-wise matrix comparison.

An unknown number of targets are allowed to travel through $\mathcal{W}$. Although the true target states are unknown, it can be assumed that all target behaviors can be

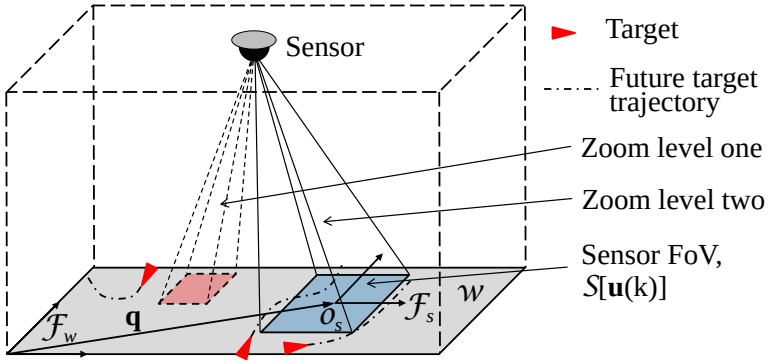


FIGURE 2.4: Illustration of the camera system, where one FOV is zoomed in and the other is zoomed out.

modeled by a possibly nonlinear time-invariant system,

$$\dot{\mathbf{x}}_j(t) = \mathbf{f}_i[\mathbf{x}_j(t)], \quad j = 1, \ldots, M(t). \tag{2.6}$$

The vector function $\mathbf{f}_i : \mathbb{R}^2 \to \mathbb{R}^2$, referred to as a *velocity field*, is also unknown and is drawn from a set $\mathcal{F} = \{\mathbf{f}_1, \ldots, \mathbf{f}_N\}$ of unknown velocity fields to be learned from data, where N is unknown. For simplicity, it is assumed that $M(t)$ can be determined without error. Note that there does not exist a one-to-one correspondence between $\mathcal{F}$ and the set of targets. This is because one or more targets in $\mathcal{W}$ may be described by the same velocity field in $\mathcal{F}$, while some velocity fields in $\mathcal{F}$ may not describe any of the targets in $\mathcal{W}$.

The problem is to determine both $\mathcal{F}$ and the association between the velocity fields in $\mathcal{F}$ and the targets in $\mathcal{W}$ based on the sensor measurements obtained up to the present time according to the model in (2.5). Let a discrete random variable g_j, with range $\mathcal{I} = \{1, \cdots, N\}$ denote the index of the velocity field that describes the behavior of the jth target. The event $\{g_j = i\}$ represents the association of target j with velocity field $\mathbf{f}_i \in \mathcal{F}$, as shown in (2.6). It is assumed that g_j obeys an unknown N-dimensional categorical distribution [8] denoted by $\text{Cat}(\boldsymbol{\pi})$, where $\boldsymbol{\pi} = [\pi_1 \ \ldots \ \pi_N]^T$ describes the prior probabilities of every possible outcome of g_j, for any $j = 1, \ldots, M(t)$, that are assumed independent and identically distributed

(i.i.d.) such that

$$\Pr\{g_j = i\} = \pi_i, \ \forall i, j, \tag{2.7}$$

where $\Pr\{g_j = i\}$ is the probability of event $\{g_j = i\}$.

Let $\boldsymbol{\xi}_i$, $i = 1, \ldots, L$, denote the L points of interest selected to represent the velocity field over the workspace. For example, the points can be L evenly spaced grid points in the workspace. Let $\mathbf{X} = [\boldsymbol{\xi}_1 \ \ldots \ \boldsymbol{\xi}_L]$ be shorthand notation for the points of interest such that

$$\mathbf{f}_i(\mathbf{X}) = [\mathbf{f}_i(\boldsymbol{\xi}_1)^T \ \ldots \ \mathbf{f}_i(\boldsymbol{\xi}_L)^T]^T. \tag{2.8}$$

Then, the random vector of interest is given by

$$\boldsymbol{\phi} \triangleq [\mathbf{f}_1(\mathbf{X})^T \ \ldots \ \mathbf{f}_N(\mathbf{X})^T]^T. \tag{2.9}$$

Chapter 3 introduces the DP-GP model connecting $\boldsymbol{\phi}$, the measurements, and the DP-GP expected KL divergence that is used in the optimized coverage planning approach in Chapter 4 in order to determing the optimal control, $\mathbf{u}^*(t)$, that enables the sensors to collect the most valuable measurements for learning $\{\boldsymbol{\phi}, \boldsymbol{\pi}\}$.

2.3 Problem 3: Mobile Sensor Planning for Target Tracking and Localization

This problem considers a mobile robotic sensor deployed to track a moving target in a two-dimensional workspace, $\mathcal{W} \subset \mathbb{R}^2$. The objective is to obtain a controller for the robotic sensor such that its ability to track and localize the target is optimized without losing the target, based on proprioceptive and exteroceptive measurements. Let the target state, position, and velocity be respectively denoted by $\mathbf{q}_t = [x_t \ y_t \ \dot{x}_t \ \dot{y}_t]^T$, $\mathbf{x}_t = [x_t \ y_t]^T$, and $\dot{\mathbf{x}}_t = [\dot{x}_t \ \dot{y}_t]^T$. Note that the random vector of interest is given by $\boldsymbol{\phi} = \mathbf{q}_t(k+1)$. The target motion in $\mathcal{W}$ is assumed to be governed by a linear stochastic motion model that, in discrete time, can be written as a difference

equation,

$$\mathbf{q}_t(k+1) \triangleq \mathbf{f}_t[\mathbf{q}_t(k)] + \mathbf{G}\mathbf{w} = \mathbf{\Phi}_t \mathbf{q}_t(k) + \mathbf{G}\mathbf{w}, \tag{2.10}$$

where $\mathbf{w}$ is zero-mean Gaussian white noise with covariance matrix $\mathbf{Q}_t$, $\mathbf{\Phi}_t$ is the state transition matrix, and $\mathbf{G}$ is the noise Jacobian matrix. Both $\mathbf{\Phi}_t$ and $\mathbf{G}$ are assumed to be time-invariant and known *a priori*.

Let $\mathbf{q}_r = [x_r \ y_r \ \theta_r]^T$ denote the robot configuration or *state* with respect to an inertial (or global) frame of reference $\mathcal{F}_\mathcal{W}$, and let $\mathbf{u}_r = [v_r \ \omega_r]^T$ denote the robot control vector, where v_r is the translational speed and ω_r is the angular velocity, where $\mathbf{u}_r \in \mathcal{U}$ and $\mathcal{U}$ is the space of admissible control inputs. The kinematics of this robotic sensor in $\mathcal{W}$ can be described by the unicycle motion model [105],

$$\mathbf{q}_r(k+1) \triangleq \mathbf{f}_r[\mathbf{q}_r(k), \mathbf{u}_r(k), k] = \mathbf{q}_r(k) + \mathbf{B}_r(k)\mathbf{u}_r(k), \tag{2.11}$$

where

$$\mathbf{B}_r(k) = \begin{bmatrix} \cos\theta_r(k)\delta t & 0 \\ \sin\theta_r(k)\delta t & 0 \\ 0 & \delta t \end{bmatrix}, \tag{2.12}$$

and δt is the time step size.

The proprioceptive sensor (e.g. odometer) obtains noisy measurements of the control vector,

$$\mathbf{z}_r(k) \triangleq \mathbf{h}_r[\mathbf{u}_r(k)] = \mathbf{u}_r(k) + \mathbf{v}_r(k), \tag{2.13}$$

where $\mathbf{v}_r(k)$ is white Gaussian noise with a time-invariant and known covariance matrix $\mathbf{Q}_r$, i.e., $\mathbf{v}_r(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_r)$. The exteroceptive sensor is characterized by a sector-shaped FOV, denoted by $\mathcal{S} \subset \mathcal{W}$, that is rigidly connected to the robot and has an aperture or central angle α and a range or radius γ, as shown in Fig.2.5. Then, the motion of any point in $\mathcal{S}$ can be described by the robot configuration vector $\mathbf{q}_r$, which includes the robot inertial position $\mathbf{x}_r = [x_r \ y_r]^T$ and heading θ_r. When the target is inside the FOV, the exteroceptive sensor can measure its relative distance

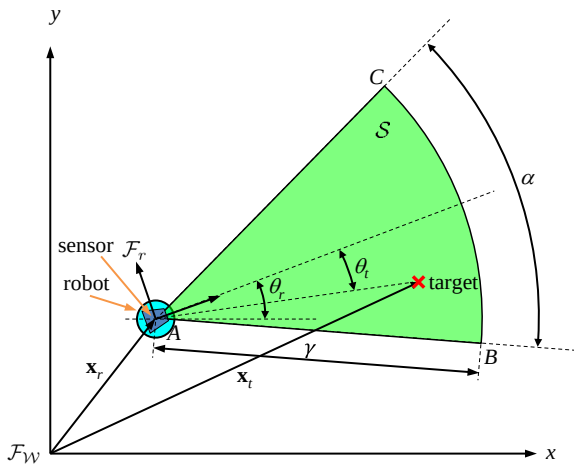


FIGURE 2.5: FOV of exteroceptive sensor.

and bearing according to the model,

$$\mathbf{z}_t \triangleq \mathbf{h}_t(\mathbf{q}_r, \mathbf{q}_t) = \begin{cases} [\rho_t \ \theta_t]^T + \mathbf{v}_t, & \mathbf{x}_t \in \mathcal{S}(\mathbf{q}_r) \\ \varnothing, & \mathbf{x}_t \notin \mathcal{S}(\mathbf{q}_r) \end{cases} \tag{2.14}$$

where $\rho_t = \|\mathbf{x}_r - \mathbf{x}_t\|$ denotes the Euclidean distance between $\mathbf{x}_r$ and $\mathbf{x}_t$, θ_t is the angle between the robot heading and the direction from robot to target, and $\mathbf{v}_t$ is zero-mean Gaussian noise with covariance $\mathbf{R}_t$.

The workspace $\mathcal{W}$ is populated with L stationary landmarks with positions $\mathbf{x}_l = [x_1 \ y_1 \ \ldots \ x_L \ y_L]^T$ that can be used to aid localization. The measurement of the landmarks also consists of the relative distance and bearing,

$$\mathbf{z}_{l_i} \triangleq \mathbf{h}_l(\mathbf{q}_r, \mathbf{x}_{l_i}) = \begin{cases} [\rho_{l_i} \ \theta_{l_i}]^T + \mathbf{v}_l, & \mathbf{x}_{l_i} \in \mathcal{S}(\mathbf{q}_r) \\ \varnothing, & \mathbf{x}_{l_i} \notin \mathcal{S}(\mathbf{q}_r) \end{cases} \tag{2.15}$$

for $i = 1, \ldots, L$, where $\rho_{l_i} = \|\mathbf{x}_r - \mathbf{x}_{l_i}\|$, θ_{l_i} is the relative angle between the robot heading and the ith landmark location and $\mathbf{v}_l$ is zero-mean Gaussian noise with covariance $\mathbf{R}_l$.

Based on the above robot and target motion model and on the most recent proprioceptive and exteroceptive measurements, $\mathbf{z}_r(k)$, $\mathbf{z}_t(k)$, and $\mathbf{z}_l(k)$, the goal is

to obtain a controller such that its ability to track and localize the target is optimized without losing the target.

3

Information Gain

Information theory addresses the quantification of the amount and the quality of information, which is accomplished by evaluating the uncertainty of one or more random variables based on their PMF or PDF and on the environment condition. 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 and control problems, the utility of the sensor control may be represented by the expected information value, where the expectation is with respect to future sensor measurements. The expected information value can then be used to estimate the utility of the sensor control prior to obtaining the measurements and therefore be used to determine the sensor control. Because the posterior belief state in sensor planning is typically unknown, a general approach is reviewed in this chapter that utilizes information theoretic functions to estimate the expected information value of a measurement resulting from the sensor control prior to obtaining the actual sensor measurements and the posterior belief. The information theoretic functions are first reviewed in the next section. An approach for deriving the expected information value functions is subsequently reviewed, followed by its

new extension that can be used to derive the expected KL divergence information function for the DP-GP model.

3.1 Information Theoretic Functions

Information theoretic functions are widely used in many applications to evaluate the information value of sensor measurements. One such function is the Shannon entropy, which measures the uncertainty of a discrete random variable $\boldsymbol{\theta}$ with a range $\boldsymbol{\Theta}$. From the PMF $p(\boldsymbol{\theta})$ for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, the Shannon entropy is defined as

$$H(\boldsymbol{\theta}) = -\sum_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} p(\boldsymbol{\theta}) \log p(\boldsymbol{\theta}). \tag{3.1}$$

Similarly, the differential entropy (also referred to as the continuous entropy) extends the Shannon entropy to PDF and is defined as

$$H(\boldsymbol{\theta}) = -\int_{\boldsymbol{\Theta}} p(\boldsymbol{\theta}) \log p(\boldsymbol{\theta}) d\boldsymbol{\theta} \tag{3.2}$$

where $\boldsymbol{\theta}$ is a continuous random variable and $\boldsymbol{\Theta}$ is the value space of $\boldsymbol{\theta}$.

The Rény information or α-divergence measures the difference (also called distance) between two PMFs (or PDFs). According to [20], the Rény divergence of order α for discrete random variables is defined as

$$D_\alpha(p \parallel q) \;=\; \frac{1}{\alpha - 1} \log \sum_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} p^\alpha(\boldsymbol{\theta})\, q^{1-\alpha}(\boldsymbol{\theta}) \tag{3.3}$$

where p and q are two PMFs of $\boldsymbol{\theta}$. The Rény divergence for a continuous random variable is obtained by replacing the summation in (3.3) with an integral, as follows:

$$D_\alpha(p \parallel q) \;=\; \frac{1}{\alpha - 1} \log \int_{\boldsymbol{\Theta}} p^\alpha(\boldsymbol{\theta})\, q^{1-\alpha}(\boldsymbol{\theta}) d\boldsymbol{\theta}. \tag{3.4}$$

In the sensor motion planning literature, p is the posterior belief of $\boldsymbol{\theta}$ given new and prior measurements, while q is the prior belief of $\boldsymbol{\theta}$ given prior measurements. As α

converges to 1, (3.3) and (3.4) reduce to the KL divergence, as follows:

$$D_{KL}(p \parallel q) = \sum_{\boldsymbol{\theta}\in\boldsymbol{\theta}} p(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} \tag{3.5}$$

for discrete random variables, and

$$D(p \parallel q) = \int_{\boldsymbol{\Theta}} p(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} dq(\boldsymbol{\theta}) \tag{3.6}$$

for continuous random variables, respectively.

As shown in [20], mutual information is a measure of the information content of one random variable regarding another random variable. The conditional mutual information of two random variables $\boldsymbol{\theta}$ and $\mathbf{z}$, given $\mathbf{y}$, represents the reduction in uncertainty in $\boldsymbol{\theta}$ due to knowledge of $\mathbf{z}$ when $\mathbf{y}$ is given; it is defined as

$$\begin{aligned} I(\boldsymbol{\theta}; \mathbf{z} \mid \mathbf{y}) &= H(\boldsymbol{\theta} \mid \mathbf{y}) - H(\boldsymbol{\theta} \mid \mathbf{z}, \mathbf{y}) \\ &= \sum_{\boldsymbol{\Theta}} \sum_{\mathcal{Y}} \sum_{\mathcal{Z}} p(\boldsymbol{\theta}, \mathbf{y}, \mathbf{z}) \log \frac{p(\boldsymbol{\theta}, \mathbf{z} \mid \mathbf{Z})}{p(\boldsymbol{\theta} \mid \mathbf{y}) p(\mathbf{z} \mid \mathbf{y})} \end{aligned} \tag{3.7}$$

where $H(\boldsymbol{\theta} \mid \mathbf{y})$ denotes the conditional entropy of $\boldsymbol{\theta}$ given $\mathbf{y}$. Similarly, the conditional mutual information of continuous random variables is obtained by replacing the summation with a triple integral, as follows:

$$\begin{aligned} I(\boldsymbol{\theta}; \mathbf{z} \mid \mathbf{y}) &= H(\boldsymbol{\theta} \mid \mathbf{y}) - H(\boldsymbol{\theta} \mid \mathbf{z}, \mathbf{y}) \\ &= \int_{\boldsymbol{\Theta}} \int_{\mathcal{Y}} \int_{\mathcal{Z}} p(\boldsymbol{\theta}, \mathbf{y}, \mathbf{z}) \log \frac{p(\boldsymbol{\theta}, \mathbf{z} \mid \mathbf{y})}{p(\boldsymbol{\theta} \mid \mathbf{y}) p(\mathbf{z} \mid \mathbf{y})} d\boldsymbol{\theta} d\mathbf{z} d\mathbf{y} \end{aligned} \tag{3.8}$$

The Cauchy-Schwarz divergence is quite useful when $q(\boldsymbol{\theta})$ and $p(\boldsymbol{\theta})$ are in non-parametric forms and is based on the Cauchy-Schwarz (CS) inequality. The Cauchy-Schwarz divergence is also a measure of the difference between two probability distributions $p(\boldsymbol{\theta})$ and $q(\boldsymbol{\theta})$ and is defined for discrete random variables as

$$D_{CS}(p, q) = \log \frac{\sum_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} p^2(\boldsymbol{\theta}) \sum_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} q^2(\boldsymbol{\theta})}{\left[\sum_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} p(\boldsymbol{\theta}) q(\boldsymbol{\theta})\right]^2}, \tag{3.9}$$

while for continuous random variables, it is defined as

$$D_{CS}(p,q) = \log \frac{\int_{\boldsymbol{\Theta}} p^2(\boldsymbol{\theta}) d\boldsymbol{\theta} \int_{\boldsymbol{\Theta}} q^2(\boldsymbol{\theta}) \boldsymbol{\theta}}{\left[\int_{\boldsymbol{\Theta}} p(\boldsymbol{\theta}) q(\boldsymbol{\theta}) d\boldsymbol{\theta}\right]^2}. \tag{3.10}$$

The entropy, the α-divergence, the KL divergence, and the Cauchy-Schwarz divergence require knowledge of the posterior $p(\boldsymbol{\theta})$. Therefore, they cannot be used to compute the expected information value because the posterior PMF is unknown prior to obtaining the measurements [117]. A general approach for designing information value function based on expected information theoretic functions is reviewed in the next section.

3.2 Information Value Functions for Sensor Planning

As shown in the previous section, computing these information theoretic functions requires knowledge of the PMFs that represent the prior and posterior belief state of $\boldsymbol{\theta}$. Although it is assumed that the sensor parameters are known *a priori*, the approach can be easily extended to the case in which they must also be controlled.

For simplicity, let the random variables of interest be denoted by $\boldsymbol{\phi} = [\boldsymbol{\theta}_1^T \cdots \boldsymbol{\theta}_M^T]^T$. The following derivation assumes that $\boldsymbol{\theta}_1^T \cdots \boldsymbol{\theta}_M^T$ are independent, continuous random variables. Therefore, without loss of generality, the general approach for deriving the information value function of a measurement $\mathbf{z}_i$ with respect to $\boldsymbol{\theta}_i$ proceeds as follows. If $\mathbf{z}_i$ is known, the information acquired through $\mathbf{z}_i$ can be represented by the KL divergence between the prior belief state, $p(\boldsymbol{\theta}_i \mid \mathcal{M}^{k-1}, \mathbf{e}_i)$, and the posterior belief state, $p(\boldsymbol{\theta}_i \mid \mathbf{z}_i, \mathcal{M}^{k-1}, \mathbf{e}_i)$, as

$$D_{KL}[p(\boldsymbol{\theta}_i \mid \mathcal{M}^{k-1}, \mathbf{z}_i, \mathbf{e}_i) \parallel p(\boldsymbol{\theta}_i \mid \mathcal{M}^{k-1}, \mathbf{e}_i)], \tag{3.11}$$

where $\mathcal{M}^{k-1}$ denotes the accumulated measurement up to time $k-1$. At time k, the expected change in belief state brought about by $\mathbf{z}_i$ can be estimated by taking the

expectation with respect to $\mathbf{z}_i$. Then, from (3.11), the expected KL divergence can be represented by

$$\begin{aligned}
&\hat{\varphi}_{D_{KL}}(\boldsymbol{\theta}_i; \mathbf{z}_i \mid \mathcal{M}^{k-1}, \mathbf{e}_i) \\
&\equiv \mathbb{E}_{\mathbf{z}_i}\left\{D_{KL}[p(\boldsymbol{\theta}_i \mid \mathbf{z}_i, \mathcal{M}^{k-1}, \mathbf{e}_i) \parallel p(\boldsymbol{\theta}_i \mid \mathcal{M}^{k-1}, \mathbf{e}_i)]\right\} \\
&= \int_{\mathcal{Z}} D_{KL}[p(\boldsymbol{\theta}_i \mid \mathbf{z}_i, \mathcal{M}^{k-1}, \mathbf{e}_i) \parallel p(\boldsymbol{\theta}_i \mid \mathcal{M}^{k-1}, \mathbf{e}_i)] p(\mathbf{z}_i \mid \mathcal{M}^{k-1}, \mathbf{e}_i) d\mathbf{z}_i. \qquad (3.12)
\end{aligned}$$

The KL divergence can be computed from $\mathcal{M}^{k-1}$ and the sensor model as follows.

When a measurement $\mathbf{z}_i^{k-1}$ is obtained from the target $\mathcal{T}_i$ at time $(k-1)$, the PDF of $\boldsymbol{\theta}_i$ given $\mathcal{M}^{k-1}$ and $\mathbf{e}_i$ can be updated using Bayes' rule,

$$\begin{aligned}
p(\boldsymbol{\theta}_i \mid \mathcal{M}^{k-1}, \mathbf{e}_i) &= p(\boldsymbol{\theta}_i \mid z^{k-1}, \mathcal{M}^{k-2}, \mathbf{e}_i) \\
&= \frac{p(z^{k-1} \mid \boldsymbol{\theta}_i, \mathcal{M}^{k-2}, \mathbf{e}_i) p(\boldsymbol{\theta}_i \mid \mathcal{M}^{k-2}, \mathbf{e}_i)}{p(z^{k-1} \mid \mathcal{M}^{k-2}, \mathbf{e}_i)} \qquad (3.13) \\
&= \frac{p(z^{k-1} \mid \boldsymbol{\theta}_i, \mathbf{e}_i) p(\boldsymbol{\theta}_i \mid \mathcal{M}^{k-2}, \mathbf{e}_i)}{\int_{\boldsymbol{\Theta}} p(z^{k-1} \mid \boldsymbol{\theta}_i, \mathbf{e}_i) p(\boldsymbol{\theta}_i \mid \mathcal{M}^{k-2}, \mathbf{e}_i) d\boldsymbol{\theta}_i},
\end{aligned}$$

because measurements can be assumed to be conditionally independent given the target state, i.e.,

$$p(\mathbf{z}^{k-1} \mid \boldsymbol{\theta}_i, \mathcal{M}^{k-2}, \mathbf{e}_i) = p(\mathbf{z}^{k-1} \mid \boldsymbol{\theta}_i, \mathbf{e}_i). \qquad (3.14)$$

Because $p(\boldsymbol{\theta}_i \mid \mathcal{M}^{k-2}, \mathbf{e}_i)$ is known from the previous time step $(k-2)$ and additional measurements are obtained at subsequent time steps, (3.13) can be implemented iteratively. Finally, the posterior belief inside the expectation in (3.12) is computed by applying Bayes' rule for every possible value of $\mathbf{z}_i$ as follows:

$$p(\boldsymbol{\theta}_i \mid \mathbf{z}_i, \mathcal{M}^{k-1}, \mathbf{e}_i) = \frac{p(\mathbf{z}_i \mid \boldsymbol{\theta}_i, \mathbf{e}_i) p(\boldsymbol{\theta}_i \mid \mathcal{M}^{k-2}, \mathbf{e}_i)}{\int_{\boldsymbol{\Theta}} p(\mathbf{z}_j \mid \boldsymbol{\theta}_i, \mathbf{e}_i) p(\boldsymbol{\theta}_i \mid \mathcal{M}^{k-2}, \mathbf{e}_i) d\boldsymbol{\theta}_i}. \qquad (3.15)$$

Similarly, the expected α-divergence can be obtained by replacing the KL diver-

gence with the α divergence as follows:

$$\begin{aligned}
&\hat{\varphi}_{D_\alpha}(\boldsymbol{\theta}_i; \mathbf{z}_i \mid \mathcal{M}^{k-1}, \mathbf{e}_i) \\
&\equiv \mathbb{E}_{\mathbf{z}_i}\left\{D_\alpha[p(\boldsymbol{\theta}_i \mid \mathbf{z}_i, \mathcal{M}^{k-1}, \mathbf{e}_i) \parallel p(\boldsymbol{\theta}_i \mid \mathcal{M}^{k-1}, \mathbf{e}_i)]\right\} \\
&= \int_{\mathcal{Z}} D_\alpha[p(\boldsymbol{\theta}_i \mid \mathbf{z}_i, \mathcal{M}^{k-1}, \mathbf{e}_i) \parallel p(\boldsymbol{\theta}_i \mid \mathcal{M}^{k-1}, \mathbf{e}_i)]p(\mathbf{z}_i \mid \mathcal{M}^{k-1}, \mathbf{e}_i)d\mathbf{z}_i
\end{aligned} \tag{3.16}$$

where $p(\boldsymbol{\theta}_i \mid \mathbf{z}_i, \mathcal{M}^{k-1}, \mathbf{e}_i)$ and $p(\mathbf{z}_i \mid \mathcal{M}^{k-1}, \mathbf{e}_i)$ can be obtained from (3.13) and (3.15).

The conditional mutual information consists of an expectation of the unknown measurement in nature and is used to represent the reduction in the uncertainty of $\boldsymbol{\theta}_i$ caused by $\mathbf{z}_i$, which is given by

$$\begin{aligned}
\hat{\varphi}_I(\boldsymbol{\theta}_i; \mathbf{z}_i \mid \mathcal{M}^{k-1}, \mathbf{e}_i) &\equiv \mathbb{E}_{\mathbf{z}_i}\left\{I(\boldsymbol{\theta}_i; \mathbf{z}_i \mid \mathcal{M}^{k-1}, \mathbf{e}_i)\right\} \\
&= H(\boldsymbol{\theta}_i \mid \mathcal{M}^{k-1}, \mathbf{e}_i) - \mathbb{E}_{\mathbf{z}_i}\left\{H(\boldsymbol{\theta}_i \mid \mathbf{z}_i, \mathcal{M}^{k-1}, \mathbf{e}_i)\right\} \\
&= H(\boldsymbol{\theta}_i \mid \mathcal{M}^{k-1}, \mathbf{e}_i) - \int_{z_j} H(\boldsymbol{\theta}_i \mid \mathbf{z}_i, \mathcal{M}^{k-1}, \mathbf{e}_i)p(\mathbf{z}_i \mid \mathcal{M}^{k-1}, \mathbf{e}_i)d\mathbf{z}_i
\end{aligned} \tag{3.17}$$

where the entropy $H(\boldsymbol{\theta}_i \mid \mathbf{z}_i, \mathcal{M}^{k-1}, \mathbf{e}_i)$ is computed from (3.15). The expected Cauchy-Schwartz information function is defined as

$$\begin{aligned}
\hat{\varphi}_{CS}(\boldsymbol{\theta}_i; \mathbf{z}_i \mid \mathcal{M}^{k-1}, \mathbf{e}_i) &\equiv \mathbb{E}_{\mathbf{z}_i}\left\{D_{CS}[p(\boldsymbol{\theta}_i \mid \mathbf{z}_i, \mathcal{M}^{k-1}, \mathbf{e}_i), p(\boldsymbol{\theta}_i \mid \mathcal{M}^{k-1}, \mathbf{e}_i)]\right\} \\
&= \int_{\mathcal{Z}} \log \frac{\int_{\boldsymbol{\Theta}} p^2(\boldsymbol{\theta}_i \mid \mathbf{z}_i, \mathcal{M}^{k-1}, \mathbf{e}_i)d\boldsymbol{\theta} \int_{\boldsymbol{\Theta}} p^2(\boldsymbol{\theta}_i \mid \mathcal{M}^{k-1}, \mathbf{e}_i)d\boldsymbol{\theta}}{\left[\int_{\boldsymbol{\Theta}} p(\boldsymbol{\theta}_i \mid \mathbf{z}_i, \mathcal{M}^{k-1}, \mathbf{e}_i)(\boldsymbol{\theta})p(\boldsymbol{\theta}_i \mid \mathcal{M}^{k-1}, \mathbf{e}_i)d\boldsymbol{\theta}\right]^2} d\mathbf{z}_i,
\end{aligned} \tag{3.18}$$

and it can be used to obtain an alternative measure of the distance between the prior and the posterior belief states.

As shown in [103], the expected KL divergence can be specialized to the DP-GP expected KL divergence, where the random vector of interest $\boldsymbol{\phi}$ does not consist of $\boldsymbol{\theta}_i$. Instead, $\boldsymbol{\phi}$ is a vector of other random variables defined as follows:

$$\boldsymbol{\phi} \triangleq [\mathbf{f}_1(\mathbf{X})^T \quad \dots \quad \mathbf{f}_N(\mathbf{X})^T]^T, \tag{3.19}$$

where $\mathbf{f}_i$ is an unknown velocity field function. The DP-GP model is used to connect $\boldsymbol{\phi}$ and the measurements that are denoted by $\mathbf{m}(k)$.

The DP-GP mixture model for describing target behaviors is studied in [43]. Based on the model of the targets' kinematics (2.6), every velocity field, $\mathbf{f}_i$, projects the jth target position, $\mathbf{x}_j(k)$, to the target velocity, $\mathbf{v}_j(k)$, and it can thus be viewed as a two-dimensional spatial phenomenon, which can be modeled by multiple-output Gaussian processes (GPs) [38]. Then, a PMF $\boldsymbol{\pi}$ describing the prior probability of an association between a target and a velocity field (GP) is learned from data to cluster the velocity fields using Dirichlet processes (DPs) [9]. DPs can be successfully applied to data clustering without specifying the number of clusters *a priori* because they allow the creation and deletion of clusters when necessary as new data is obtained over time. The DP-GP mixture model taken from [43] is given as follows:

$$\begin{aligned} \{\boldsymbol{\theta}_i, \boldsymbol{\pi}\} &\sim \mathrm{DP}(\alpha, \mathrm{GP}_0),\ i = 1, \ldots, \infty \\ G_j &\sim \mathrm{Cat}(\boldsymbol{\pi}),\ j = 1, \ldots, M \\ \mathbf{f}_{G_j}(\mathbf{x}) &\sim \mathrm{GP}(\boldsymbol{\theta}_{G_j}, \mathbf{c}),\ \forall \mathbf{x} \in \mathcal{W},\ j = 1, \ldots, M, \end{aligned} \tag{3.20}$$

where the strength parameter is denoted by α [92]. For a rigorous definition and a comprehensive review of DPs, the reader is referred to [25]. In this dissertation, the base distribution is chosen to be a Gaussian process, $\mathrm{GP}_0 = \mathrm{GP}(\mathbf{0}, \mathbf{c})$. Let a Gaussian process GP_i represent the distribution of velocities over the workspace specified by the ith velocity field, $\mathbf{f}_i$, such that

$$\mathbf{f}_i(\mathbf{x}) \sim \mathrm{GP}_i, \quad \forall \mathbf{x} \in \mathcal{W}, \tag{3.21}$$

for $i = 1, \ldots, N$. The Gaussian process, GP_i, is completely specified by its mean function $\boldsymbol{\theta}_i : \mathbb{R}^2 \to \mathbb{R}^2$,

$$\boldsymbol{\theta}_i(\mathbf{x}) = \mathbb{E}[\mathbf{f}_i(\mathbf{x})], \quad \forall \mathbf{x} \in \mathcal{W}, \tag{3.22}$$

and its covariance function,

$$\mathbf{c}_i(\mathbf{x}_\imath, \mathbf{x}_\jmath) = \mathbb{E}\left\{[\mathbf{f}_i(\mathbf{x}_\imath) - \boldsymbol{\theta}_i(\mathbf{x}_\imath)][\mathbf{f}_i(\mathbf{x}_\jmath) - \boldsymbol{\theta}_i(\mathbf{x}_\jmath)]^T\right\} \triangleq \begin{bmatrix} c_{xx}(\mathbf{x}_\imath, \mathbf{x}_\jmath) & c_{xy}(\mathbf{x}_\imath, \mathbf{x}_\jmath) \\ c_{yx}(\mathbf{x}_\imath, \mathbf{x}_\jmath) & c_{yy}(\mathbf{x}_\imath, \mathbf{x}_\jmath) \end{bmatrix} \tag{3.23}$$

for all $\mathbf{x}_\imath, \mathbf{x}_\jmath \in \mathcal{W}$.

Similar to the approach for deriving the information function previously introduced, the proposed *DP-GP expected KL divergence* is the expectation of the utility of $\mathbf{m}_j(k+1)$ for updating the current DP-GP model. In this case, the future measurement $\mathbf{m}_j(k+1)$ and the association between the target and the velocity field are unknown. Let $\mathcal{E}(k)$ denote the measurement histories of all of the targets already used for updating the DP-GP model, and let $\mathcal{E}_i(k)$ denote the measurement history assigned to the ith velocity field by the DP-GP mixture model.

Then, following (3.16), the DP-GP expected KL divergence for a future measurement $\mathbf{m}_j(k+1)$ can be defined as

$$\hat{\varphi}_j[\boldsymbol{\phi}; \mathbf{m}_j(k+1)] =$$

$$\sum_{i=1}^{N} w_{ij} \mathbb{E}_{\mathbf{m}_j(k+1)} D_{KL}\Big\{p[\mathbf{f}_i(\mathbf{X})|\mathbf{m}_j(k+1), \mathcal{M}_j(k), \mathcal{E}_i(k)] \parallel p[\mathbf{f}_i(\mathbf{X})|\mathcal{M}_j(k), \mathcal{E}_i(k)]\Big\} \tag{3.24}$$

where w_{ij} is the probability of the target j following $\mathbf{f}_i$ and $\mathcal{M}_j(k)$ denotes the measurements from target j up to time k.

The DP-GP expected KL divergence (3.24) requires w_{ij} and a PDF of $\mathbf{m}_j(k+1)$ that is a function of $\mathbf{x}_j(k+1)$ and $\mathbf{y}_j(k+1)$, which are also unknown. The PDF of $\mathbf{m}_j(k+1)$ is obtained as follows. When $G_j = i$ and $\mathbf{x}_j(k+1) \in \mathcal{S}[\mathbf{u}(k+1)]$, the estimated measurement distribution is obtained by marginalizing the measurement model over the estimated target position distribution such that

$$p[\mathbf{m}_j(k+1)|\mathcal{M}_j(k), \mathcal{E}_i(k)] = \int_{\mathcal{X}} p[\mathbf{m}_j(k+1)|\mathcal{M}_j(k), \mathcal{E}_i(k, \mathbf{x}_j(k+1)] \times p[\mathbf{x}_j(k+1)|\mathcal{M}_j(k), \mathcal{E}_i(k)] d\mathbf{x}_j(k+1) \tag{3.25}$$

where $p[\mathbf{x}_j(k+1)|\mathcal{M}_j(k), \mathcal{E}_i(k), \mathbf{u}(k)]$ is obtained from the current DP-GP model and the current target position estimation as follows. Recalling (2.6), the target kinematics can be integrated using the Euler method for a time interval δt such that

$$\mathbf{x}_j(k+1) = \mathbf{x}_j(k) + \mathbf{v}_j(k)\delta t \tag{3.26}$$

where $\mathbf{v}_j(k)$ is from the current DP-GP model given $\mathbf{x}_j(k)$. Therefore, the estimated target position distribution at $k+1$ is obtained by the following integral:

$$p[\mathbf{x}_j(k+1)|\mathcal{M}_j(k), \mathcal{E}_i(k)] = \int_{\mathcal{V}} f_X[\mathbf{x}_j(k+1) - \mathbf{v}_j(k)\delta t] f_V[\mathbf{v}_j(k), \mathbf{x}_j(k)] d\mathbf{v}_j \tag{3.27}$$

where f_X and f_V represent the probability density functions of the target position and velocity at time step k, respectively:

$$f_X[\mathbf{x}_j(k)] \triangleq p[\mathbf{x}_j(k)|\mathcal{M}_j(k), \mathcal{E}_i(k)], \tag{3.28}$$

$$f_V[\mathbf{v}_j(k), \mathbf{x}_j(k)] \triangleq \mathcal{N}[\mathbf{v}_j(k); \boldsymbol{\mu}_j(k), \boldsymbol{\Sigma}_j(k)], \tag{3.29}$$

where the mean and variance of the jth target velocity at position $\mathbf{x}_j(k)$ are given by

$$\boldsymbol{\mu}_j(k) = \boldsymbol{\theta}_i(\mathbf{x}_j(k)) + \mathbf{C}(\mathbf{x}_j(k), \mathbf{P}_i(k))[\mathbf{C}(\mathbf{P}_i(k), \mathbf{P}_i(k)) + \sigma_v^2 \mathbf{I}]^{-1}[\mathbf{V}_i(k) - \boldsymbol{\theta}_i(\mathbf{P}_i(k))], \tag{3.30}$$

and

$$\boldsymbol{\Sigma}_j(k) = \mathbf{c}[\mathbf{x}_j(k), \mathbf{x}_j(k)] - \mathbf{C}[\mathbf{x}_j(k), \mathbf{P}_i(k)][\mathbf{C}(\mathbf{P}_i(k), \mathbf{P}_i(k)) + \sigma_v^2 \mathbf{I}]^{-1} \mathbf{C}[\mathbf{P}_i(k), \mathbf{x}_j(k)]. \tag{3.31}$$

Here, the matrices of the position and velocity measurements in $\mathcal{E}_i(k)$ are defined as

$$\mathbf{P}_i(k) \triangleq [\mathbf{y}_1(1) \quad \dots \quad \mathbf{y}_j(\ell) \quad \dots \quad \mathbf{y}_M(k)], \quad \forall [\mathbf{y}_j^T(\ell) \quad \mathbf{z}_j^T(\ell)]^T \in \mathcal{E}_i(k), \tag{3.32}$$

and

$$\mathbf{V}_i(k) \triangleq [\mathbf{z}_1(1) \quad \dots \quad \mathbf{z}_j(\ell) \quad \dots \quad \mathbf{z}_M(k)], \quad \forall [\mathbf{y}_j^T(\ell) \quad \mathbf{z}_j^T(\ell)]^T \in \mathcal{E}_i(k). \tag{3.33}$$

The probability of target j following velocity field i can be inferred from Bayes' theorem,

$$w_{ij} \triangleq p[G_j = i|\mathcal{M}_j(k), \mathcal{E}(k)] = \frac{p[\mathcal{M}_j(k)|\mathcal{E}_i(k)]p[G_j = i|\mathcal{E}(k)]}{\sum_{i=1}^{M} p[\mathcal{M}_j(k)|\mathcal{E}_i(k)]p[G_j = i|\mathcal{E}(k)]} \tag{3.34}$$

where the prior PMF of G_j given $\mathcal{E}(k)$ and $\mathbf{u}(k)$ is the categorical distribution of the current DP-GP model. Taken from [43], the likelihood $p[\mathcal{M}_j(k)|\mathcal{E}_i(k), \mathbf{u}(k)]$ is obtained from the ith GP (velocity field) of the current DP-GP model:

$$p[\mathcal{M}_j(k)|\mathcal{E}_i(k)] = \prod_{\ell=k'}^{k} \mathcal{N}[\mathbf{z}_j(\ell); \hat{\boldsymbol{\mu}}_j(\ell), \hat{\boldsymbol{\Sigma}}_j(\ell)], \tag{3.35}$$

where the estimated mean, $\hat{\boldsymbol{\mu}}_j(\ell)$, and covariance, $\hat{\boldsymbol{\Sigma}}_j(\ell)$, are calculated by replacing $\mathbf{x}_j(k)$ with $\mathbf{y}_j(\ell)$ in both (3.30) and (3.31). When $\mathbf{x}_j(k+1) \notin \mathcal{S}[\mathbf{u}(k+1)]$, no measurement is obtained and the prior and posterior DP-GP models are the same; in this case, the DP-GP expected KL divergence (3.24) is zero.

From the above analysis, with the given probability models of $\mathbf{m}_j(k+1)$, (3.24) becomes

$$\begin{aligned}
&\hat{\varphi}_j[\boldsymbol{\phi}; \mathbf{m}_j(k+1)] \\
&= \sum_{i=1}^{N} w_{ij} \int_{\mathcal{S}[\mathbf{u}(k+1)]} \int_{\mathcal{Z}} D\left\{p[\mathbf{f}_i(\mathbf{X})|\mathcal{M}_j(k+1), \mathcal{E}_i(k)] \parallel p[\mathbf{f}_i(\mathbf{X})|\mathcal{M}_j(k), \mathcal{E}_i(k)]\right\} \\
&\times \mathcal{N}[\mathbf{z}_j(k+1); \boldsymbol{\mu}_j(k+1), \boldsymbol{\Sigma}_v] d\mathbf{z}_j(k+1) \\
&\times p[\mathbf{x}_j(k+1)|\mathcal{M}_j(k), \mathcal{E}_i(k)] d\mathbf{x}_j(k+1).
\end{aligned} \tag{3.36}$$

The calculation of (3.36) involves a 6th-order integral (including an implicit double integral in D) that is reduced to a double integral in Chapter 4 with proper assumptions. Then, the optimized coverage approach is used to maximize the proposed DP-GP expected KL divergence information function to obtain the optimal sensor control for the camera intruder problem.

4

Motion Planning

In many applications, the sensors are mounted on platforms that consist of autonomous mobile robots or mechatronic structures whose kinematics can be modeled by differential equations. To optimize the sensor performance, the sensor motion planning and control must consider the stochastic measurement process and filters for estimation, classification, and prediction. Additionally, sensor motion planning must take into account the geometry of the sensor's platform and FOV as well as the geometry and location of targets and obstacles to determine a path that optimizes the sensing objectives while avoiding collisions with obstacles and other sensors.

Because robot motion planning approaches deal with the intersections of discrete geometric objects that are possibly moving, subject to a kinematic or dynamic equation, many sensor path planning methods are inspired by existing robot motion planning approaches. Chapter 4 reviews three existing sensor motion planning approaches originally presented in [15], [113], and [66]: the information cell decomposition approach, the information probability roadmap deploy (IPD), and the rapidly exploring random information trees (RRIT) approach. However, these existing methods cannot solve sensor planning problems when sensor kinodynamic constraints are

considered, the target model is complex, or the proprioceptive and exteroceptive sensor are deployed, respectively. The IP approach is presented for generating a potential navigation function and roadmap based on a probabilistic model of the measurement process and on the geometries of targets and sensor FOV [68]. The above approaches assume that a measurement is obtained once the sensor FOV intersects with the geometry of the stationary target. For the problem of monitoring moving targets, the locations of these targets are unknown and are estimated by time-varying probability distributions. Therefore, it is difficult to formulate a target with a rigid geometry. To this end, two sensor motion planning approaches are developed for problems where positions of mobile targets are unknown: the optimized coverage planning based on the DP-GP expected KL divergence and the optimized visibility planning for simultaneous target tracking and localization.

4.1 Information Cell Decomposition

Cell decomposition is a well-known approach for decomposing the obstacle-free robot configuration space into a finite collection of non-overlapping convex polygons that are referred to as cells, with the purpose of obtaining a robot path without collisions with obstacles. In classical cell decomposition, the union of these non-overlapping cells is equivalent to the free configuration space through a line-sweeping algorithm. Then, a connectivity graph is constructed based on these cells by adding an arc between two cells if the two cells are adjacent. The connectivity graph can then be searched for the shortest path between the two cells containing the desired initial and final robot configurations. One advantage of cell decomposition is that it guarantees collision avoidance between a robot with any discrete geometry and obstacles of any shape that are not necessarily convex. One commonly used decomposition method is known as approximate-and-decompose [119].

The cell decomposition approach for sensor motion planning was proposed in

[16]; it modifies the classical cell decomposition approach by taking into account the presence of the targets and the sensors' FOV. This approach maps the information values of position-fixed targets into the cells formed by decomposing the free configuration space, with the goal of classifying the targets located in an obstacle-populated workspace. Then, similar to the classical cell decomposition approach, a connectivity graph is built to represent the connectivity relationship between cells and is further transformed into a decision tree from which an optimal sensor path can be found.

Contrary to classical cell decomposition approaches, the free configuration space is decomposed into two types of cells. The first type, called an observation cell, is a convex polygon $\mathcal{D}_z$ and a sensor at any configuration in this observation cell can make an observation of at least one target. In other words, the information value of an observation cell is positive. The remaining cells are referred to as void cells that have zero information values, meaning that a sensor at any configuration in a void cell cannot take measurements of any target. The cell decomposition approach for sensor planning consists of three steps. The first step is to generate a convex polygonal decomposition, $\mathcal{D}_{void}$, of the configuration space that is not covered by any C-obstacles or C-targets. Then, the second step generates a convex polygonal decomposition, $\mathcal{D}_i$, of each obstacle-free C-target, constructing $\mathcal{D}_z$. In the last step, a connectivity graph G using all cells in $\mathcal{D} = \mathcal{D}_{void} \cup \mathcal{D}_z$ is constructed. Note that the presence of obstacles and targets makes the decomposition in the first step NP-hard [60]. The cells containing the initial configuration $\mathbf{q}_0$ and the final configuration $\mathbf{q}_f$ are referred to as the initial cell ($\mathbf{m}_0$) and final cell ($\mathbf{m}_f$), respectively. Examples of the polygonal decomposition and connectivity graph are illustrated in Fig. 4.1.

As shown in [2], the A^* algorithm is the most effective in searching for the path of minimum total distance in G. The A^* algorithm explores G iteratively, starting at $\mathbf{m}_0$ and visiting every neighbor node $\mathbf{m}_i$, where a cost function is assigned by estimating the minimum-cost path from $\mathbf{m}_0$ to $\mathbf{m}_f$, through $\mathbf{m}_i$. After a node in G

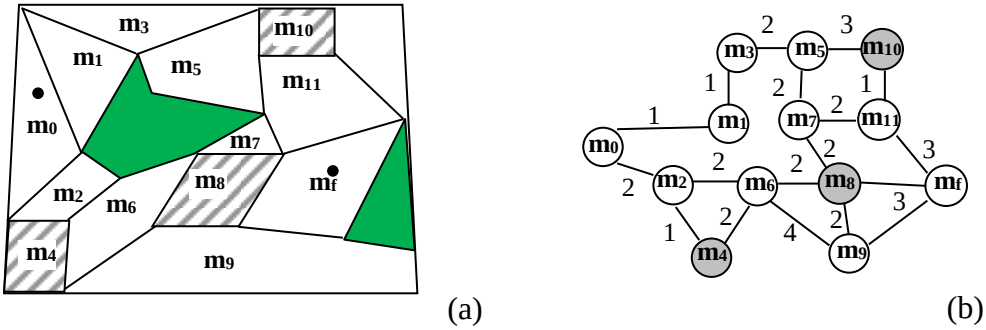


FIGURE 4.1: Example of cell decomposition with void (white) and observation (grey) cells and C-obstacles (green) (a), and connectivity graph G (b).

is visited, the algorithm stores only the path of minimum cost and labels the node as visited, assigning it a pointer to its parent node. This process forms a spanning tree of the subset of G that has already been explored and produces considerable computational savings compared to other graph-searching algorithms [55, 80].

4.2 Information Roadmap Deployment

Probability Roadmap Method (PRM) algorithms have been shown to be very effective at planning collision-free paths for robots with many degrees of freedom [90, 11]. The PRM method samples milestones from the free configuration space and constructs a roadmap graph. Then, a collision-free path from the initial configuration to the final configuration is determined from the roadmap by searching the resultant roadmap graph [46]. An information value-based probabilistic roadmap method, referred to as the Information Probability Roadmap Deploy (IRD) approach, was proposed in [112]. The IRD approach samples a roadmap using a hybrid sampling method for a robotic sensor deployed to classify multiple position-fixed targets in a workspace populated with obstacles. The hybrid sampling method consists of a PDF constructed from an information theoretic function that favors samples with a high expected value of information, a Gaussian distribution covering narrow passages, and

a uniform distribution covering wide-open regions.

The information value of the measurements that can be obtained from a robotic sensor at configuration $\mathbf{q}$ is the cumulative information values of these C-targets, denoted by $V(\mathbf{q})$. Then, the sampling PDF based on the information value function is given by

$$p_V(\mathbf{q}) = \frac{V(\mathbf{q})}{\int_{\mathcal{CT}} V(\mathbf{q}) d\mathbf{q}}. \tag{4.1}$$

It can be shown that this PDF favors milestones with higher information values. The above PDF is used together with a uniform PDF $p_U(\mathbf{q})$ and a Gaussian PDF $p_G(\mathbf{q})$ to form a hybrid sampling approach. The probability of using p_V, p_U, and p_G to sample a given milestone is v_1, v_2, and $1 - v_1 - v_2$, respectively. The IRD approach first samples one index from $\{1, 2, 3\}$ with categorical probabilities v_1, v_2, and $1 - v_1 - v_2$, then the corresponding distribution is used to sample a milestone $\mathbf{m}_i$. This sampling process iterates L times to create the nodes of the roadmap $G = (M, E)$. Subsequently, the set of arcs E is obtained by a local planner that connects every milestone $\mathbf{m}_i \in M$ with its k nearest milestones. The arc from $\mathbf{m}_i$ to its neighbor $\mathbf{m}_j$ is associated with a traveling distance $\|W[\mathbf{m}_{i+1} - \mathbf{m}_i]\|$, where $\|\cdot\|$ represents the Euclidian norm [88]. The constant parameter k is a user-defined parameter that is chosen based on N_m and on the complexity of $\mathcal{W}$.

Let τ denote a trajectory; then, the total path distance is given by the sum of all weighted Euclidian norms along a path τ:

$$D(\tau) = \sum_{i=0}^{f-1} \|W[\mathbf{q}(i) - \mathbf{q}(i+1)]\|, \tag{4.2}$$

where $\mathbf{q}(i)$ is the robot configuration at i along τ. The total reward is defined as

$$V(\tau) = \sum_{i=0}^{f-1} V[\mathbf{q}(i)]. \tag{4.3}$$

Because the goal of the robotic sensor is to maximize the measurement information profit and minimize the traveling distance, after connecting the initial configuration and final configuration in G, the objective function to be maximized is given by

$$R(\tau) = w_V \; V(\tau) - w_D \; D(\tau) \tag{4.4}$$

where the user-defined constants w_V and w_D weigh the trade-off between $V(\tau)$ and $D(\tau)$. As shown in [2], the A^* algorithm is the most effective algorithm in searching for the path of minimum total distance in G.

4.3 Rapidly Exploring Information Random Trees

Rapidly-Exploring Random Trees (RRTs) provide an efficient way to search for a path in a configuration space online and have been successfully applied to nonholonomic robots in high-dimensional workspaces. Using the initial robot configuration, the tree is expanded by iterating incrementally over the discrete time index $t_k = 1, 2, \ldots$ as follows. First a configuration $\mathbf{q}$ is randomly sampled from the free configuration space using a PDF $p(\mathbf{q})$, possibly uniform or Gaussian. Then, based on a distance metric, the closest node to $\mathbf{q}$ in the tree is computed and extended toward $\mathbf{q}$ within a predefined distance ϵ to obtain $\mathbf{q}'$. If the path lies in the free configuration space, $\mathbf{q}'$ and this path are added to the tree; otherwise, they are discarded and a new random configuration is re-sampled.

The modified RRT method with a new sampling method is presented for sensor path planning [66], where the PDF $p(\mathbf{q})$ is generated based on the geometry and information value of the target using a normal mixture. The sampled configurations are ordered based on the robotic sensor state (exploration or exploitation), the expected information value of the target assigned to the sensor, and the distance to the target. The RRTs are expanded by verifing the feasibility of these sampled configurations and connecting feasible milestones to the current trees.

Let $\Theta_i = (\theta_i^1, \theta_i^2, \ldots, \theta_i^n) \in \mathbb{R}$ denote the directions of all the rays emitted from the center of $\mathcal{F}_{\mathcal{A}i}$, and let μ_i^j denote the orientation of θ_i^j in $\mathcal{F}_{\mathcal{A}}$. Let $L_i(\mathbf{q}) = (l_i^1, l_i^2, \ldots, l_i^n) \in \Re$ denote the magnitude for each ray. Assume the distribution of θ_i^s is a mixture of normal distributions with n components. Each normal distribution corresponds to an orientation of the vector of the 2-dimensional FOV. Then, for the ith robotic sensor, the direction of the sth ray is given by

$$\theta_i^s \sim \sum_{j=1}^{n} m_i^j N(\mu_i^j, \sigma_{1i}^2) \tag{4.5}$$

where m_i^j is the weight for the jth normal distribution, μ_i^j is the mean and is set to the direction of the jth reflex, and σ_{1i} is the standard deviation. Similarly, the magnitudes of $L_i(\mathbf{q})$ can also be sampled.

Once a number of milestones are sampled for the ith sensor, they are ordered based on their importance (or priorities). When the robotic sensor is in the exploration state, the importance of a milestone is proportional to the distance between the sample and the robot's current state, given by

$$R(\mathbf{q}) = \rho_i(\mathbf{q}) \tag{4.6}$$

where $\rho_i(\mathbf{q})$ is the distance between $\mathbf{q}$ and the agent. From the above definition, the robotic sensor favors a milestone that is far away from its current configuration. For the robotic sensor in the exploitation state (i.e., the priority is to make a measurement from a nearby target), the importance is defined as

$$R(\mathbf{q}) = k_2 e^{-\frac{1}{2f\rho_i(\mathbf{q})^2}} + k_1 \sum_{j \in N_i} e^{-\frac{\rho_j(\mathbf{q})^2}{2eV(j)^2}} \tag{4.7}$$

where k_1 and k_2 are two constant representing the weight, N_i is the index of targets that is assigned to the ith robotic sensor, $\rho_j(\mathbf{q})$ is the distance between $\mathbf{q}$ and the jth target, and $V(j)$ is the information value of the jth target. Thus, the sampler

prefers to generate a sample with a large distance from its current configuration and a small distance from its assigned targets.

During online sensor path planning, a global RRIT does not exist and is not necessary for each robotic sensor. A local RRIT is constructed and updated for each robotic sensor during its movement. Because the robotic sensor always moves towards sub-root of the subtree expanded to the milestone with highest value of R, it is not necessary to keep other sibling trees. The tree of milestones is updated when the robotic sensor reaches the root of the tree. The tree is updated online by iterating between the following three steps. In the first step, a number of milestones (configurations) are sampled and are sorted in descended order of their importance values. Then, the feasibility of the sampled milestone is checked by computing the expected path to the selected milestone $\mathbf{q}$ from the nearest milestone (measured with the Euclidian distance) stored at the nodes of the tree. In other words, this step verifies that the entire expected path lies in the free configuration space. During the last step, the feasible node of the highest importance value is chosen as the goal configuration. Then, the robotic sensor navigates to this node by the controller used to check the path feasibility. One example is shown in Fig. 4.2.

4.4 Information Potential Approach for Integrated Control and Navigation

This section presents an approach for building a potential navigation function and roadmap based on the information value and geometry of the targets, referred to as the information potential (IP) method. A novel information potential function is introduced, followed by a switched feedback controller for integrated sensor path planning and control and an information roadmap algorithm for escaping local minima.

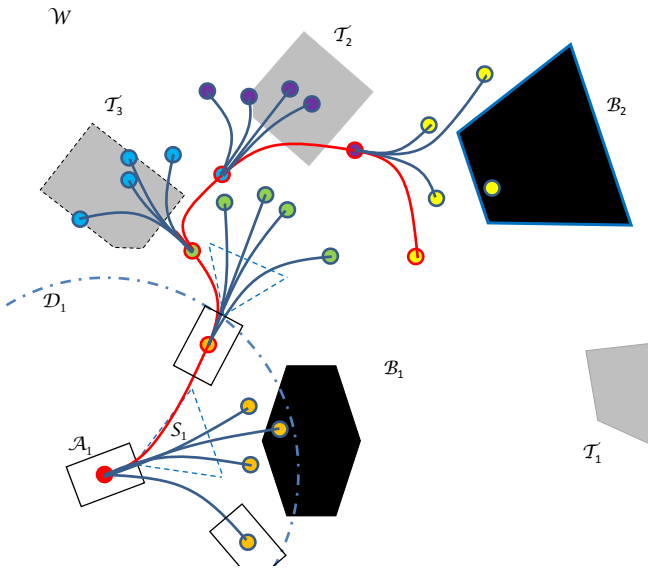


FIGURE 4.2: Example of RRIT with milestones, trees, and actual path.

4.4.1 Information Potential Function

Because the number of targets correctly classified by a sensor cannot be established *a priori*, objective (I) is achieved by maximizing the expected *information value* of the measurements, defined as the reduction of uncertainty in the target state $\boldsymbol{\theta}_i$ brought about by $\mathbf{z}_i$. As shown in Chapter 3, prior to obtaining a noisy measurement value $\mathbf{z}_i$, the *expected* information value of target $\mathcal{T}_i$ can be measured by the expected conditional mutual information,

$$\begin{aligned} V_i &\triangleq \mathbb{E}_{\mathbf{z}_i}\{I(\boldsymbol{\theta}_i; \mathbf{z}_i \mid \mathbf{e}_i)\} = H(\boldsymbol{\theta}_i \mid \mathbf{e}_i) - \mathbb{E}_{\mathbf{z}_i}\{H(\boldsymbol{\theta}_i \mid \mathbf{z}_i, \mathbf{e}_i)\} \\ &= H(\boldsymbol{\theta}_i) - \sum_{\mathbf{z}_i} p(\mathbf{z}_i \mid \mathbf{e}_i) H(\boldsymbol{\theta}_i \mid \mathbf{z}_i, \mathbf{e}_i) \end{aligned} \tag{4.8}$$

where $H(\boldsymbol{\theta}_i \mid \mathbf{e}_i) = H(\boldsymbol{\theta}_i)$ because $\mathbf{x}_i$ and $\mathbf{e}_i$ are independent. This information value is mapped onto the C-target to construct a potential field.

Similar to classical potential approaches, the robotic sensor's potential function is the sum of the attractive and repulsive potentials,

$$U(\mathbf{q}) = U_{att}(\mathbf{q}) + U_{rep}(\mathbf{q}), \tag{4.9}$$

where the total attractive potential is obtained by multiplying attractive potentials from all targets together, given by

$$U_{att}(\mathbf{q}) \triangleq \prod_{i=1}^{M} U_{trg}^{i}(\mathbf{q}). \tag{4.10}$$

In the proposed IP approach, the attractive potential is obtained by mapping the information value of a target onto the C-target region of this target in $\mathcal{W}$, given by

$$U_{trg}^{i}(\mathbf{q}) \triangleq \eta_1 \, \sigma \, V_i^b \left\{ 1 - \exp\left[-\frac{\rho_i(\mathbf{q})^2}{2\,\sigma\, V_i^b} \right] \right\}, \; i = 1, \ldots, M \tag{4.11}$$

where $\rho_i(\mathbf{q})$ denotes the minimum Euclidian distance between $\mathbf{q}$ and $\mathcal{T}_i$ in $\mathcal{C}$. The constant η_1 is positive, representing the importance of targets relative to other path planning objectives, such as avoiding collisions with obstacles. The influence distance of the target is determined by σ and b that are two positive constants. It can be shown that every C-target in $\mathcal{C}$ is a local minimum.

The total repulsive potential consists of two different repulsive potentials that are defined for fixed and moving obstacles and is given by

$$U_{rep}(\mathbf{q}) \triangleq \sum_{l \in B_0} U_{obs}^{l}(\mathbf{q}) + \sum_{j \in R_0} U_{rob}^{j}(\mathbf{q}), \tag{4.12}$$

where sets B_0 and R_0 are the index sets of the considered obstacles and robots. For fixed obstacles, a potential barrier is generated around the C-obstacle region to prevent collisions and, at the same time, to allow the robot to obtain measurements from nearby targets. For a fixed obstacle $\mathcal{B}_l \subset \mathcal{W}$, the C-obstacle $\mathcal{CB}_l = \{\mathbf{q} \in \mathcal{C} \mid \mathcal{A}(\mathbf{q}) \cap \mathcal{B}_l \neq \varnothing\}$ is computed and used to determine the minimum distance from $\mathbf{q}$ in the configuration space:

$$\varrho_l(\mathbf{q}) = \min_{\mathbf{q}' \in \mathcal{CB}_l} \|\mathbf{q} - \mathbf{q}'\|. \tag{4.13}$$

Let B denote the index set of fixed obstacles detected in $\mathcal{W}$ up to the present time. Then, the repulsive potential for $\mathcal{B}_l$, $l \in B$, is

$$U_{obs}^{l}(\mathbf{q}) \triangleq \begin{cases} \frac{1}{2}\eta_2 \left(\frac{1}{\varrho_l(\mathbf{q})} - \frac{1}{d_0}\right)^2 U_{att}(\mathbf{q}) & \text{if} \quad \varrho_l(\mathbf{q}) \leqslant d_0 \\ 0 & \text{if} \quad \varrho_l(\mathbf{q}) > d_0 \end{cases} \tag{4.14}$$

where η_2 is a positive scaling factor that represents the importance of fixed obstacles relative to other path-planning objectives and d_0 is the obstacle distance of influence [55].

The repulsive potential of a moving obstacle creates a virtual barrier in $\mathcal{C}$ regardless of the presence of targets within the distance of influence. Let R denote the index set of moving obstacles detected in $\mathcal{W}$ up to the present time. Then, the repulsive potential for $\mathcal{B}_j$ with $j \in R$ is

$$U_{rob}^{j}(\mathbf{q}) \triangleq \begin{cases} \frac{1}{2}\eta_3 \left(\frac{1}{\varrho_j(\mathbf{q})} - \frac{1}{d_0}\right)^2 & \text{if} \quad \varrho_j(\mathbf{q}) \leqslant d_0 \\ 0 & \text{if} \quad \varrho_j(\mathbf{q}) > d_0 \end{cases} \tag{4.15}$$

where η_3 is a positive scaling factor that represents the importance of moving obstacles relative to other path-planning objectives.

As in classical potential field methods [55], a virtual force proportional to the negative gradient of the potential function (4.9) is used to control the robotic sensor and is comprised of the sum of an attractive and a repulsive force, generated by the corresponding potentials. The gradient of the potential function (4.9) can be obtained as follows:

$$\begin{aligned} \nabla U(\mathbf{q}) &= \nabla U_{att}(\mathbf{q}) + \nabla U_{rep}(\mathbf{q}) \\ &= \sum_{l \in B_0} F_l(\mathbf{q})\mathbf{v}_l(\mathbf{q}) + \sum_{i=1}^{M}[N_i(\mathbf{q}) + A_i(\mathbf{q})]\mathbf{n}_i(\mathbf{q}) - \sum_{j \in R_0} \eta_3 \left(\frac{1}{\varrho_j(\mathbf{q})} - \frac{1}{d_0}\right) \frac{\mathbf{v}_j(\mathbf{q})}{\varrho_j(\mathbf{q})^2} \end{aligned} \tag{4.16}$$

where

$$A_i(\mathbf{q}) \triangleq \frac{1}{2}\eta_1\eta_2 \left[\sum_{l\in B_0}\left(\frac{1}{\varrho_l(\mathbf{q})}-\frac{1}{d_0}\right)^2\right]\prod_{i\neq j} U^j_{trg}(\mathbf{q})\rho_i(\mathbf{q})\exp\left[-\frac{\rho_i(\mathbf{q})^2}{2\sigma V_i^b}\right], \tag{4.17}$$

$$N_i(\mathbf{q}) = \prod_{j\neq i} U^j_{trg}(\mathbf{q})\,\eta_1\,\rho_i(\mathbf{q})\,\exp\left[-\frac{\rho_i(\mathbf{q})^2}{2\sigma V_i^b}\right], \tag{4.18}$$

$$F_l(\mathbf{q}) \triangleq \eta_2\left(\frac{1}{\varrho_l(\mathbf{q})}-\frac{1}{d_0}\right)\frac{U_{att}(\mathbf{q})}{\varrho_l(\mathbf{q})^2}, \tag{4.19}$$

and $\mathbf{v}_l(\cdot) \triangleq \nabla\varrho_l(\cdot)$ is a vector supported by a vector between $\mathbf{q}$ and the closest point in $\mathcal{CB}_l$, pointing away from $\mathcal{CB}_l$. As can be seen, in the proposed IP approach, the attractive force is proportional to the information value of the target that generated it.

4.4.2 Switched Controller

Several issues arise when a nonholonomic robot is controlled by the negative gradient of the potential function, which are i) trapping in local minima; ii) the goals are non-reachable due to nearby obstacles; iii) stabilization. Other issues include oscillations and no passages between closely spaced obstacles. A switched controller is designed to ensure closed-loop stability while also enabling sensor measurements, based on the following observation. When the robotic sensor is far away from the target, the position vector plays a dominant role in control, whereas the robotic sensor heading should be considered when it is close to the target in order to obtain measurements. This is accomplished by introducing a vector $\mathbf{h}_i \in \mathcal{W}$ that points from $\mathcal{O}_\mathcal{A}$ to the target $\mathcal{T}_i$. Then, every vector,

$$\mathbf{h}_i \triangleq \{\arg\min_{\mathbf{y}\in\mathcal{CT}_i}\|\mathbf{y}-\mathbf{x}\|-\mathbf{x}\}, \quad i\in I_T, \tag{4.20}$$

specifies a goal orientation β_i defined as the angle that the projection of $\mathbf{h}_i$ onto the inertial xy-plane makes with the x-axis. As illustrated in Fig. 4.3, the goal of

the switched controller is to navigate the robot into the cylinder defined as the set $\lambda_i = \{(x, y, \theta) : [x\ y]^T = \boldsymbol{\xi}_i,\ 0 \leqslant \theta \leqslant 2\pi\}$ and then to adjust the sensor to meet the desired heading.

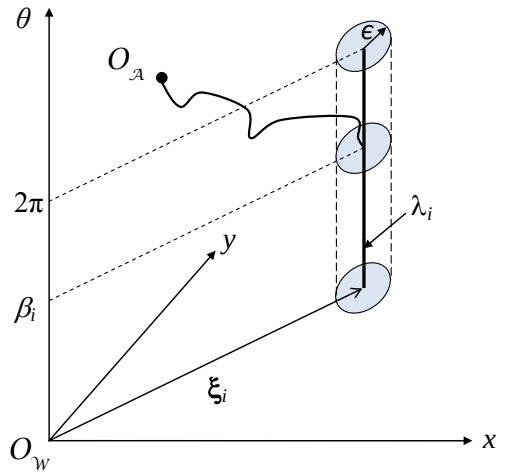


FIGURE 4.3: Goal of switched control law for a given inscribed circle with center $\boldsymbol{\xi}_i$ and a positive constant ϵ.

When the sensor is not in a cylinder of any target, i.e., $\|\mathbf{h}\| > \epsilon$ where $\mathbf{h} \triangleq \arg\min\{\|\mathbf{h}_i\|, i \in I_T\}$, the potential function U is defined in terms of the distance,

$$\rho_i(\mathbf{q}) = \|\boldsymbol{\xi}_i - \mathbf{x}\|, \tag{4.21}$$

to bring $\mathbf{q}$ within an ϵ of λ_i. Then, the potential-based controller is given by

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} a \\ w \end{bmatrix} = \begin{bmatrix} -\mathbf{S}(\mathbf{q})^T \nabla U(\mathbf{q}) - k_1 v \\ \dot{\alpha}[U(\mathbf{q})] + k_0\{\alpha[U(\mathbf{q})] - \theta\} \end{bmatrix}, \tag{4.22}$$

where $\mathbf{S}(\mathbf{q}) \triangleq [\cos\theta\ \ \sin\theta\ \ 0]^T$, k_0 and k_1 are a positive constants, and α is the orientation angle of vector $\partial_{\mathbf{x}} U \triangleq [\partial_x U(\mathbf{q})\ \partial_y U(\mathbf{q})]^T$ in the inertial xy-plane. The orientation angle, α, and its time derivative can be obtained from the components of (4.16) as follows:

$$\alpha[U(\mathbf{q})] = 2\arctan\left(\frac{\frac{\partial U(\mathbf{q})}{\partial y}}{\sqrt{(\frac{\partial U(\mathbf{q})}{\partial y})^2 + (\frac{\partial U(\mathbf{q})}{\partial x})^2} + \frac{\partial U(\mathbf{q})}{\partial x}}\right) + \pi, \tag{4.23}$$

$$\dot{\alpha}[U(\mathbf{q})] = \frac{\frac{\partial U(\mathbf{q})}{\partial x}}{(\frac{\partial U(\mathbf{q})}{\partial x})^2 + (\frac{\partial U(\mathbf{q})}{\partial y})^2} \left(\frac{\partial^2 U(\mathbf{q})}{\partial x \partial y} \dot{x} + \frac{\partial^2 U(\mathbf{q})}{\partial y^2} \dot{y} \right)$$

$$- \frac{\frac{\partial U(\mathbf{q})}{\partial y}}{(\frac{\partial U(\mathbf{q})}{\partial x})^2 + (\frac{\partial U(\mathbf{q})}{\partial y})^2} \left(\frac{\partial^2 U(\mathbf{q})}{\partial x \partial y} \dot{y} + \frac{\partial^2 U(\mathbf{q})}{\partial x^2} \dot{x} \right). \tag{4.24}$$

When the robot is in a cylinder of a target, i.e., $\|\mathbf{h}\| \leqslant \epsilon$, the heading β_i is considered to construct the controller. The distance between the robot and the target is computed with respect to the geometric dilatation of the C-target,

$$\mathcal{CT}'_i \triangleq \{\mathbf{q} \in \mathbb{R}^3 \mid \mathbf{x} = \delta_r(\mathbf{x}' - \boldsymbol{\xi}_i) + \boldsymbol{\xi}_i, \forall \mathbf{x}' \in \mathcal{PT}_i, 0 \leqslant \theta \leqslant 2\pi\}, \tag{4.25}$$

where

$$\delta_r = (r_i - C) / \max_{\mathbf{x} \in \mathcal{PT}_i} \|\mathbf{h}_i\| \tag{4.26}$$

is the scale factor and $C \in (0,\ r_i)$ is a constant chosen by the user. Then, for $\|\mathbf{h}\| \leqslant \epsilon$, the potential-based control law is switched to

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} a \\ w \end{bmatrix} = \begin{bmatrix} -k_p \mathbf{S}(\mathbf{q})^T \nabla U(\mathbf{q}) - k_1 v \\ k_0(\beta_i - \theta) \end{bmatrix}, \tag{4.27}$$

where k_0, k_1, and k_p are positive constants.

4.4.3 Information Roadmap for Escaping Local Minima

When there exist multiple targets and obstacles in $\mathcal{W}$, a well-known limitation of potential field methods is that the robot can be trapped in local minima of U [55]. This subsection presents a PRM-based method for escaping local minima while increasing the probability of obtaining sensor measurements. The method uses the information potential function defined in (4.9) to construct a PDF for sampling milestones and then builds a local roadmap representation of the free configuration space. Contrary to traditional sampling methods for path planning [46], the method uses the robot kinematics (2.4) and the switched controllers to verify connectivity between

milestones. As a result, after escaping a local minimum, the robotic sensor's configuration can be proven to asymptotically converge to the milestone with the lowest potential (or highest information value).

Similar to the sampling approach in the RRT method, a milestone $\mathbf{m}_l$ is sampled from the PDF of a three-dimensional continuous random vector, given by a nonnegative function $f_{\mathbf{q}}$ such that

$$\mathbf{P}(\mathbf{q} \in Q) = \int_Q f_{\mathbf{q}}(\mathbf{q}) d\mathbf{q} \tag{4.28}$$

for any subspace $Q \subset \mathcal{C}$ randomly chosen, where

$$f_{\mathbf{q}}(\mathbf{q}) = \begin{cases} \frac{\exp[-U(\mathbf{q})]}{\int_Q \exp[-U(\mathbf{q})] d\mathbf{q}}, & \mathbf{q} \in Q \\ 0, & \mathbf{q} \notin Q \end{cases}. \tag{4.29}$$

From (4.29), it can be seen that the probability of a sample falling in a region of Q is higher (or lower) where the value of U is lower (or higher). As a result, configurations in Q that are close to, or inside, C-targets with high information value and that are far away from obstacles are sampled with higher probability.

Following a direct sampling approach [18], κ milestones are sampled from (4.29) and used to construct an ordered set $\mathcal{M} = \{\mathbf{m}_0, \ldots, \mathbf{m}_\kappa\}$. A local roadmap is then constructed, as shown in Fig. 4.4, starting with the local minimum $\mathbf{m}_0 \in Q$ and using a local planner to connect $\mathbf{m}_0$ to other milestones in $\mathcal{M}$ until no reachable milestones remain in $\mathcal{M}$. At every step of the algorithm, all of the milestones in $\mathcal{M}$ that can be connected to a milestone already in G are added to G and are deleted from $\mathcal{M}$ (Fig. 4.4.b-c). The algorithm continues until there are no more milestones in $\mathcal{M}$, and milestones that remain unconnected (Fig. 4.4.d) are discarded. As shown in the next section, after building the roadmap G, an escape path leading to a target can be obtained in a finite number of iterations. Also, through this path, the robotic

sensor has a higher probability of converging to a target with higher information value.

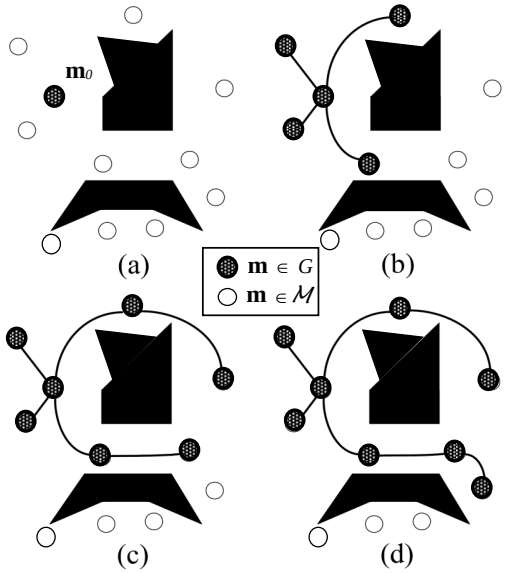


FIGURE 4.4: Roadmap Construction: (a) initial milestones; (b) first step; (c) second step; (d) final step. Dash circle: local minimum; white circle: milestones; black area: C-obstacles.

4.4.4 Propterties of Information Potential Method

The information potential presented satisfies the properties of potential navigation functions [35]: i) U^i_{trg} is an increasing function of ρ_i; ii) as $\rho_i \to \infty$, U^i_{trg} converges to a finite positive value. Also, it is shown that the switched controller is asymptotically stable, that the information roadmap method is guaranteed to find an escape path to a C-target using a finite number of iterations, and that the target with the highest information value has the highest probability of being measured by the robotic sensor.

Closed-loop Stability of Switched Feedback Control Law

The switched controller can be proven to be asymptotically stable under the following simplifying assumptions: (i) $\mathbf{q}$ is within the influence distance of only one target $\mathcal{T}_i$; and (ii) there are no obstacles within a distance d_0, i.e. $B_0 = R_0 = \varnothing$. Let $\mathcal{PT}_i$

denote the intersection of $\mathcal{CT}_i$ with the horizontal plane $\{x, y, \beta_i\}$, as shown in 4.5. Now, let $\boldsymbol{\xi}_i$ and r_i denote the center and radius of the inscribed circle for $\mathcal{PT}_i$, respectively, and let $\epsilon \in (0,\ r_i)$ denote a positive constant chosen by the user.

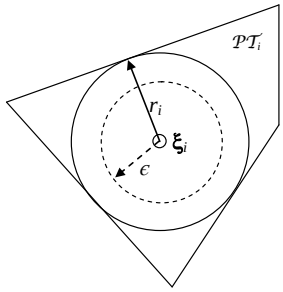


FIGURE 4.5: Inscribed circle for polygon $\mathcal{PT}_i$, with center $\boldsymbol{\xi}_i$ and radius r_i.

Proof. When $\|\mathbf{h}\| \geqslant \epsilon$, consider the Lyapunov function candidate

$$\mathcal{V}(\boldsymbol{\chi}) = U(\mathbf{q}) + \frac{1}{2}v^2 + \frac{1}{2}\{\alpha[U(\mathbf{q})] - \theta\}^2. \tag{4.30}$$

It can be shown that $\mathcal{V}(\boldsymbol{\chi}) > 0$ for all $\boldsymbol{\chi} \in \mathbb{R}^4$, because $U(\mathbf{q}) > 0$ and the term $v^2 + \{\alpha[U(\mathbf{q})] - \theta\}^2 \geqslant 0$. Under assumptions (i)-(ii), the gradient of the potential function (4.16) is

$$\nabla U(\mathbf{q}) = \begin{bmatrix} K(\boldsymbol{\xi}_i - \mathbf{x}) & 0 \end{bmatrix}^T, \tag{4.31}$$

where $K \triangleq \eta_1 \exp[-\rho_i(\mathbf{q})^2/(2\sigma V_i^b)]$. Then, the time derivative of $\mathcal{V}$ for the *closed-*

loop system can be shown to be non-negative, as follows:

$$\begin{aligned}
\dot{\mathcal{V}}(\boldsymbol{\chi}) &= \nabla U(\mathbf{q})^T \dot{\mathbf{q}} + v\dot{v} + \{\alpha[U(\mathbf{q})] - \theta\}\{\dot{\alpha}[U(\mathbf{q})] - w\} \\
&= \nabla U(\mathbf{q})^T \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \\
&+ v\left[-\mathbf{S}(\mathbf{q})^T \nabla U(\mathbf{q}) - k_1 v\right] - k_0 \left\{\alpha[U(\mathbf{q})] - \theta\right\}^2 \\
&= \nabla U(\mathbf{q})^T \mathbf{S}(\mathbf{q}) v \\
&- \mathbf{S}(\mathbf{q})^T \nabla U(\mathbf{q}) v - k_1 v^2 - k_0 \left\{\alpha[U(\mathbf{q})] - \theta\right\}^2 \\
&= -k_1 v^2 - k_0 \{\alpha[U(\mathbf{q})] - \theta\}^2 \leqslant 0. \qquad (4.32)
\end{aligned}$$

Because $\mathcal{V}(\boldsymbol{\chi})$ is radially unbounded, given any positive constant c, the set $\Omega_c = \{\boldsymbol{\chi} \in \mathbb{R}^4 \mid \mathcal{V}(\boldsymbol{\chi}) \leqslant c\}$ is a compact, positively invariant set, where the value of c, and thus the region of attraction, can be determined based on the maximum linear velocity v_{max}.

It can be shown, as follows, that every trajectory starting in Ω_c approaches the equilibrium set $\mathcal{E} = \{\boldsymbol{\chi} \in \mathbb{R}^4 \mid \boldsymbol{\chi} = [\boldsymbol{\xi}_i^T \ \alpha[U(\boldsymbol{\xi}_i)] \ 0]^T\}$ as $t \to \infty$. The set Υ of all points in Ω_c where $\dot{\mathcal{V}}(\boldsymbol{\chi}) = 0$ is $\Upsilon = \{\boldsymbol{\chi} \in \Omega_c \mid \theta = \alpha[U(\mathbf{q})], \ v = 0\}$. Let $\{\Upsilon \backslash \mathcal{E}\}$ denotes the complement set of $\mathcal{E}$ in Υ. Then, it can be shown that $\mathcal{E}$ is the largest invariant set in Υ because for any $\boldsymbol{\chi} \in \{\Upsilon \backslash \mathcal{E}\}$, as

$$\begin{aligned}
\dot{v} &= u_1 = -\mathbf{S}(\mathbf{q})^T \nabla U(\mathbf{q}) - k_1 v \\
&= [\cos\{\alpha[U(\mathbf{q})\} \ \sin\{\alpha[U(\mathbf{q})\} \ 0] \nabla U(\mathbf{q}) \\
&= \frac{\nabla U(\mathbf{q})^T}{||\nabla U(\mathbf{q})||} \nabla U(\mathbf{q}) = ||\nabla U(\mathbf{q})|| \neq 0. \qquad (4.33)
\end{aligned}$$

Thus, any trajectory starting from $\{\Upsilon \backslash \mathcal{E}\}$ cannot stay identically in Υ. Additionally, any trajectory starting from $\mathcal{E}$ will remain identically in Υ because $\dot{\boldsymbol{\chi}} = 0$. Then $\mathcal{E}$ is the largest invariant set in Υ and, according to LaSalle's Invariance Principle [49, pg. 128], every trajectory starting in Ω_c approaches $\mathcal{E}$ as $t \to \infty$.

When $\|\mathbf{h}\| \leqslant \epsilon$, another candidate Lyapunov function is considered,

$$\mathcal{V}(\boldsymbol{\chi}) = k_p\, U(\mathbf{q}) + \frac{1}{2}v^2 + \frac{1}{2}(\beta_i - \theta)^2, \tag{4.34}$$

and it is true that $\mathcal{V}(\boldsymbol{\chi})$ is non-negative. From the unicycle model (2.4) and the controller (4.27), the time derivative of $\mathcal{V}$ for the *closed-loop* system can be written as

$$\dot{\mathcal{V}}(\mathbf{q}) = -k_1 v^2 - k_0(\beta_i - \theta)^2 \leqslant 0. \tag{4.35}$$

Because $\epsilon < r_i$ any time $\|\mathbf{h}\| \leqslant \epsilon$, the state of the closed-loop system is a point in the set $\Omega_r = \{\boldsymbol{\chi} \in \mathbb{R}^4 \mid \|\mathbf{h}\| \leqslant r_i, 0 \leqslant \theta \leqslant 2\pi, \|v\| < v_{max}\}$. Let $\phi = \min_{\|\mathbf{h}\|=r_i} \mathcal{V}(\boldsymbol{\chi})$ in Ω_r, where it can be easily shown that $\phi > 0$. Take $c \in (0,\ \phi)$ and let $\Omega_c = \{\boldsymbol{\chi} \in \Omega_r \mid \mathcal{V}(\boldsymbol{\chi}) \leqslant c\}$.

Then, as $t \to \infty$, every trajectory starting in Ω_c approaches the equilibrium set $\mathcal{E} = \{\boldsymbol{\chi} \in \Omega_c \mid [\cos\beta_i\ \ \sin\beta_i]\partial_{\mathbf{x}} U = 0, \theta = \beta_i, v = 0\}$. The set Υ of all points in Ω_c where $\dot{\mathcal{V}}(\boldsymbol{\chi}) = 0$ is $\Upsilon = \{\boldsymbol{\chi} \in \Omega_c \mid \theta = \beta_i, v = 0\}$. The set $\mathcal{E}$ is the largest invariant set in Υ because, for any $\boldsymbol{\chi} \in \{\Upsilon \backslash \mathcal{E}\}$,

$$\begin{aligned} \dot{v} &= u_1 = -k_p \mathbf{S}(\mathbf{q})^T \nabla U(\mathbf{q}) - k_1 v \\ &= -k_p[\cos\beta_i\ \ \sin\beta_i\ 0]\nabla U(\mathbf{q}) \\ &= -k_p[\cos\beta_i\ \ \sin\beta_i]\partial_{\mathbf{x}} U \neq 0 \end{aligned} \tag{4.36}$$

by definition of $\mathcal{E}$. Because any trajectory starting in $\{\Upsilon \backslash \mathcal{E}\}$ cannot remain identically in Υ, $\mathcal{E}$ is the largest invariant set in Υ. On the other hand, any trajectory starting at a point in $\mathcal{E}$ will remain identically in Υ, because $\dot{\boldsymbol{\chi}} = 0$. It follows from LaSalle's Invariance Principle [49, pg. 128] that every trajectory starting in Ω_c approaches $\mathcal{E}$ as $t \to \infty$.

□

Expected Number of Iterations of Information Roadmap Algorithm

The roadmap G containing an escape path leading to a C-target can be obtained in a finite number of iterations, when a robot is trapped in a local minimum. Assume there exists a finite number of local minima in U and that all of the configurations in $Q \subset \mathcal{C}$ are reachable under the switched controllers (4.22)-(4.27). Then, partition Q into $(m + n)$ compact subspaces, $g_1, \ldots, g_m$ and $h_1, \ldots, h_n$, such that for any $\mathbf{q} \in g_j \subset Q$, the robot will converge to a configuration $\mathbf{q} \in \mathcal{CT}_j \subset Q$, and for any $\mathbf{q} \in h_l \subset Q$, the robot will converge to local minimum $\tilde{\mathbf{q}}_l \in Q$.

Because all of the milestones in G are reachable, they are connected. Let $p(h_i, g_j)$ denote the probability that the robot will converge from $\tilde{\mathbf{q}}_i$ to $\mathcal{CT}_j$, and let $p(h_i, h_l)$ denote the probability that it will converge from $\tilde{\mathbf{q}}_i$ to another local minimum $\tilde{\mathbf{q}}_l$. Because $p(h_i, g_j)$ and $p(h_i, h_l)$ are independent of the robot position, the movement of the robotic sensor can be modeled as a Markov chain, as shown in [54]. In particular, for a robot controlled by the IP method, the transition matrix, denoted by $\mathbf{M}$, can be written as

$$\mathbf{M} = \begin{bmatrix} p(g_1, g_1) & \ldots & p(g_1, g_m) & p(g_1, h_1) & \ldots & p(g_1, h_n) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ p(g_m, g_1) & \ldots & p(g_m, g_m) & p(g_m, h_1) & \ldots & p(g_m, h_n) \\ p(h_1, g_1) & \ldots & p(h_1, g_m) & p(h_1, h_1) & \ldots & p(h_1, h_n) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ p(h_n, g_1) & \ldots & p(h_n, g_m) & p(h_n, h_1) & \ldots & p(h_n, h_n) \end{bmatrix}, \tag{4.37}$$

where for a robot trapped at $\tilde{\mathbf{q}}_i$ the probability of sampling a milestone $\mathbf{m}$ in g_j is

$$p(h_i, g_j) \triangleq p(\{\mathbf{m} \in g_j\} \mid \{\mathbf{q} = \tilde{\mathbf{q}}_i\}) = \int_{\mathbf{q}' \in g_j} f_{\mathbf{q}}(\mathbf{q}')\, d\mathbf{q}', \tag{4.38}$$

and the probability of sampling a milestone $\mathbf{m}$ in h_i is,

$$p(h_i, h_l) \triangleq p(\{\mathbf{m} \in h_l\} \mid \{\mathbf{q} = \tilde{\mathbf{q}}_i\}) = \int_{\mathbf{q}' \in h_l} f_{\mathbf{q}}(\mathbf{q}')\, d\mathbf{q}'. \tag{4.39}$$

It can be shown that (4.37) can be partitioned as follows,

$$\mathbf{M} = \begin{bmatrix} \mathbf{I}_m & \mathbf{R} \\ \mathbf{A} & \mathbf{B} \end{bmatrix}, \tag{4.40}$$

where $\mathbf{I}_m$ is an $m \times m$ identity matrix. Under the stated assumptions, $\mathbf{R} \in \mathbb{R}^{m \times n}$ is a zero matrix because when $\tilde{\mathbf{q}}_i \in g_j$ no sampling is necessary. $\mathbf{A} \in \mathbb{R}^{n \times m}$ and $\mathbf{B} \in \mathbb{R}^{n \times n}$ are matrices with nonnegative entries because $f_{\mathbf{q}}(\cdot)$ is a nonnegative function. Then, based on the properties of probability functions, the sum of the entries in every row of $\mathbf{B}$ is less than one, i.e., every element of the vector $\mathbf{Bn}$ is less than one, where $\mathbf{n} \triangleq 1_{n \times 1}$. From Gershgorin's theorem [95] it follows that all eigenvalues of $\mathbf{B}$ are less than one [89] and $\lim_{\iota \to \infty} \mathbf{B}^{\iota} \to 0$, where $\mathbf{B}^{\iota}$ represents the matrix $\mathbf{B}$ raised to the ι power. It also follows that the matrix inverse $(\mathbf{I}_n - \mathbf{B})^{-1} \triangleq \mathbf{C}$ exists and can be written as $\mathbf{C} = \mathbf{CB} + \mathbf{I}_n$. The expected number of times that the Markov chain will visit h_k prior to g_j when starting at h_i is equal to $\mathbf{C}_{(i,k)}$, by using the approach in [58]. Then, since $\mathbf{C}_{(i,k)} \geqslant 0$, whenever sampling starts at the event $\{\mathbf{q} = \tilde{\mathbf{q}}_i\}$, the sampling event $\{\mathbf{m} \in h_l\}$ is expected to take place c_i times prior to event $\{\mathbf{m} \in g_j\}$, where $c_i \triangleq \sum_{k=1}^{n} \mathbf{C}_{(i,k)}$. Multiplying $\mathbf{C}$ by $\mathbf{n}$, and letting $\mathbf{c} \triangleq [c_1 \cdots c_n]^T$, we have

$$\mathbf{Cn} = \mathbf{CBn} + \mathbf{I}_n\mathbf{n} = \mathbf{CBn} + \mathbf{n} = \mathbf{c} \preceq \gamma\mathbf{Cn} + \mathbf{n} = \gamma\mathbf{c} + \mathbf{n}, \tag{4.41}$$

where γ denotes the largest element in $\mathbf{Bn}$. For two matrices $\mathbf{A}$ and $\mathbf{B}$ with the same dimensions, the inequality $\mathbf{A} \preceq \mathbf{B}$ means that $\mathbf{A}_{(i,j)} \leqslant \mathbf{B}_{(i,j)}$ for all (i, j) . Thus, the expected number of iterations c_i is finite for all i and has the upper bound $1/(1-\gamma)$.

Properties of Information Roadmap Method

It can be shown that given two targets, $\mathcal{T}_j$ and $\mathcal{T}_\ell$, with the same geometry but different information values (say, e.g., $V_j > V_\ell$), the robot configuration has a higher probability of converging to $\mathcal{CT}_j$ than to $\mathcal{CT}_\ell$, assuming the two paths from $\tilde{\mathbf{q}}_i$ to the targets are otherwise equivalent. Then, according to (4.29), the PDF $f_{\mathbf{q}}(\mathbf{q})$ is

an increasing function of the information value associated with $\mathbf{q}$. Additionally, the information value for all $\mathbf{q} \in \mathcal{CT}_j$ is V_j, and the information value for all $\mathbf{q} \in \mathcal{CT}_\ell$ is V_ℓ. Because g_j and g_ℓ have the same geometry, it follows from (4.38) that $p(h_k, g_j) > p(h_k, g_\ell)$ for any $h_k \in Q$.

From (4.37), the probability that the robotic sensor will move from a milestone in g_j to a milestone in h_i is

$$P_G(h_i, g_j) = \sum_{k=1}^{n} p(h_i, h_k) P_G(h_k, g_j) + p(h_i, g_j). \tag{4.42}$$

Then, we have

$$P_G(h_i, g_j) = \sum_{k=1}^{n} \mathbf{C}_{(i,k)} p(h_k, g_j) > \sum_{k=1}^{n} \mathbf{C}_{(i,k)} p(h_k, g_\ell) = P_G(h_i, g_\ell). \tag{4.43}$$

Therefore, by using the information roadmap G to escape a local minimum $\tilde{\mathbf{q}}_i \in h_i$, the robotic sensor has a higher probability of converging to target $\mathcal{T}_j$ than to $\mathcal{T}_\ell$, when $V_j > V_\ell$.

4.5 Optimized Coverage Planning

The approaches previously introduced assume that a measurement is obtained once the sensor FOV intersects with the geometry of the stationary target. For the problem of monitoring moving targets, the locations of these targets are unknown and are estimated by time-varying probability distributions. Therefore, it is difficult to formulate a target with a rigid geometry. Additionally, the information value functions for this type of problem are difficult to evaluate. For example, the DP-GP expected KL divergence consists of a 6th order integral. This section presents the optimized coverage planning approach for a sensor, where the FOV is assumed to be a free-flying object. The DP-GP expected KL divergence is first approximated

by Monte Carlo integration based on a set of particles sampled from the prior (predicted) target position distribution at $k+1$ in order to reduce the computational complexity. The variance of the resulting approximation error is shown to shrink at a rate inversely proportional to the number of samples. Then, these sampled particles are used to facilitate the searching for the optimal FOV configuration.

4.5.1 Particle Filter

The position of the FOV centroid and zoom levels of all cameras must be planned in advance in order to obtain measurements of the moving targets. Thus, the estimation of every target's position propagated one time step ahead (at k+1), denoted by $\mathbf{x}_j(k{+}1^-)$, must be obtained for planning purposes, where $k{+}1^-$ denotes the moment immediately before $k+1$ when $\mathbf{m}_j(k+1)$ is not available. When the camera loses sight of targets, measurements can be empty sets, resulting in a nonlinear observation model. Therefore, a classical particle filter algorithm, sequential importance resampling (SIR) [37], is adopted to estimate prior and posterior target position distribution using a Gaussian mixture model as a proposal distribution. This Gaussian mixture model is the transition probability distribution of the targets' positions at $k+1^-$, which is built upon the learned DP-GP model and particles representing the posterior distributions of the targets' positions at k. The position propagation of target j under the estimated DP-GP model ($\{\mathcal{F}, \boldsymbol{\pi}\}$) from k to $k+1$ is

$$\mathbf{x}_j(k+1) = \mathbf{x}_j(k) + f_{g_j}[\mathbf{x}_j(k)]\delta t. \tag{4.44}$$

The particle filter consists of three steps. In the first step, the samples from the posterior distribution of the jth target position at time step k given $\mathcal{M}_j(k)$ and $\mathcal{E}_i(k)$ are represented by a set of particle and weight pairs,

$$\mathcal{P}_{ji}(k) \triangleq \big\{\big(\omega_{jis}(k), \boldsymbol{\chi}_{jis}(k)\big) : 1 \leqslant s \leqslant S\big\}, \tag{4.45}$$

where S is the number of particles for each velocity field, $\boldsymbol{\chi}_{jis}(k)$ represents sth

particle for velocity field j and target i, and $\omega_{jis}(k)$ represents the associated weight, such that

$$\sum_{s=1}^{S} \omega_{jis}(k) = w_{ji}, \tag{4.46}$$

where w_{ji} is given by (3.34).

In the second step, utilizing the target position prorogation (4.44) and the DP-GP model, $\mathbf{x}_j(k+1^-)$ can be obtained and is represented by a Gaussian mixture,

$$\mathbf{x}_j(k+1^-) \sim \sum_{i=1}^{N}\sum_{s=1}^{S} \omega_{jis}(k)\mathcal{N}[\boldsymbol{\eta}_{jis}(k+1^-), \boldsymbol{\Lambda}_{jis}(k+1^-)],$$

where

$$\boldsymbol{\eta}_{jis}(k+1^-) = \boldsymbol{\chi}_{jis}(k)+\boldsymbol{\mu}_{jis}(k)\ \delta t, \tag{4.47}$$

$$\boldsymbol{\Lambda}_{jis}(k+1^-) = \boldsymbol{\Sigma}_{jis}(k)\ \delta t^2. \tag{4.48}$$

Here, $\boldsymbol{\mu}_{jis}$ and $\boldsymbol{\Sigma}_{jis}$ are the mean and variance of the sth Gaussian component of target j for velocity field i at $\boldsymbol{\chi}_{jis}(k)$, which can be calculated from (3.30-3.31) by replacing $\mathbf{x}_j(k)$ with $\boldsymbol{\chi}_{jis}(k)$. This Gaussian mixture is used as the optimal proposal distribution to sample *transient* particles representing the probability distribution of $\mathbf{x}_j(k+1^-)$, as follows:

$$\boldsymbol{\chi}_{jis}(k+1^-) \sim \sum_{s=1}^{S} \left\{ \frac{\omega_{jis}(k)}{w_{ji}} \mathcal{N}[\boldsymbol{\eta}_{jis}(k+1^-), \boldsymbol{\Lambda}_{jis}(k+1^-)] \right\}, \tag{4.49}$$

and the transient particle sets are denoted by

$$\mathcal{P}_{ji}(k+1^-) \triangleq \left\{ \left(\omega_{jis}(k+1^-), \boldsymbol{\chi}_{jis}(k+1^-)\right) : 1 \leqslant s \leqslant S \right\}, \tag{4.50}$$

where

$$\omega_{jis}(k+1^-) = w_{ji}/S. \tag{4.51}$$

Finally, when an empty measurement of target j is obtained, i.e., $\mathbf{m}_j(k+1) = \varnothing$, $\omega_{jis}(k+1)$ is set to zero if $\boldsymbol{\chi}_{jis}(k+1^-) \in \mathcal{S}[\mathbf{u}(k+1)]$. Then, the weights of all of the

particles for target j are normalized. The particles stay the same as the transient particles, such that $\boldsymbol{\chi}_{jis}(k+1) = \boldsymbol{\chi}_{jis}(k+1^-)$. Therefore, similar to (4.45), the samples from the posterior probability distribution of target j at time $k+1$ can also be represented by the weighted particles:

$$\mathcal{P}_{ji}(k+1) = \big\{\big(\omega_{jis}(k+1), \boldsymbol{\chi}_{jis}(k+1)\big) : 1 \leqslant s \leqslant S\big\}.$$

If a non-empty measurement $\mathbf{m}_j(k+1)$ is obtained, the weights associated with the particles are updated via the measurement model (2.14) as follows:

$$\omega_{jis}(k+1) = \frac{w_{ji}\ \omega_{jis}(k+1^-)\mathcal{N}[\mathbf{z}_j(k+1); \boldsymbol{\chi}_{jis}(k+1^-), \boldsymbol{\Sigma}_j(k+1)]}{\sum_{s=1}^{S} \omega_{jis}(k+1^-)\mathcal{N}[\mathbf{z}_j(k+1); \boldsymbol{\chi}_{jis}(k+1^-), \boldsymbol{\Sigma}_j(k+1)]}. \tag{4.52}$$

In the following sections, all of the transient particle sets obtained by (4.49), denoted by $\mathcal{P}_{ji}(k+1^-)$, are utilized to facilitate the calculation of the DP-GP expected KL values for the search of the optimal camera control.

4.5.2 Approximation of DP-GP Expected KL Divergence

The calculation of (3.36) involves a 6th-order integral (including an implicit double integral in D), which can be reduced to a double integral. Let $h_i[\mathbf{x}_j(k+1)]$ denote the multiple integral of the KL divergence over $\mathbf{y}_j(k+1)$ and $\mathbf{z}_j(k+1)$:

$$h_i[\mathbf{x}_j(k+1)] = \int_{\mathbf{y}_j}\int_{\mathbf{z}_j} D\left\{p[\mathbf{f}_i(\mathbf{X})|\mathcal{M}_j(k+1), \mathcal{E}_i(k)] \parallel p[\mathbf{f}_i(\mathbf{X})|\mathcal{M}_j(k), \mathcal{E}_i(k)]\right\}$$

$$\times \mathcal{N}[\mathbf{z}_j(k+1); \boldsymbol{\mu}_j(k+1), \boldsymbol{\Sigma}_v]\mathcal{N}[\mathbf{y}_j(k+1); \mathbf{x}_j(k+1), \boldsymbol{\Sigma}_x] d\mathbf{z}_j(k+1) d\mathbf{y}_j(k+1). \tag{4.53}$$

If $p[\mathbf{f}_i(\mathbf{X})|\mathcal{M}_j(k+1), \mathcal{E}_i(k), \mathbf{u}(k)]$ and $p[\mathbf{f}_i(\mathbf{X})|\mathcal{M}_j(k), \mathcal{E}_i(k)]$ in $D[\cdot\|\cdot]$ are Gaussian distributions, then for any $\mathbf{x}_j(k+1) \in \mathcal{S}[\mathbf{u}(k+1)]$

$$h_i[\mathbf{x}_j(k+1)] = \frac{1}{2}\left[\mathrm{tr}\left(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_2\right) - \ln\left(\frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|}\right) - 2L + \mathrm{tr}(\mathbf{Q}^{-1}\mathbf{R}^T\boldsymbol{\Sigma}_1^{-1}\mathbf{R}\mathbf{Q}^{-1})\mathrm{tr}(\Sigma_v)/2\right] \tag{4.54}$$

where $\mathrm{tr}(\cdot)$ denotes the trace of a matrix and

$$\boldsymbol{\Sigma}_1 = \mathbf{C}(\mathbf{X},\mathbf{X}) - \mathbf{C}[\mathbf{X},\mathbf{P}(k)]\{\mathbf{C}[\mathbf{P}(k),\mathbf{P}(k)] + \boldsymbol{\Sigma}_v\}^{-1}\mathbf{C}[\mathbf{P}(k),\mathbf{X}] \tag{4.55}$$

$$\boldsymbol{\Sigma}_2 = \mathbf{C}(\mathbf{X},\mathbf{X}) - \mathbf{C}[\mathbf{X},\mathbf{P}(k{+}1)]\{\mathbf{C}[\mathbf{P}(k{+}1),\mathbf{P}(k{+}1)] + \boldsymbol{\Sigma}_v\}^{-1}\mathbf{C}[\mathbf{P}(k{+}1),\mathbf{X}] \tag{4.56}$$

$$\mathbf{R} = \mathbf{C}[\mathbf{X},\mathbf{x}(k{+}1)] - \mathbf{C}[\mathbf{X},\mathbf{P}(k)]\{\mathbf{C}[\mathbf{P}(k),\mathbf{P}(k)] + \boldsymbol{\Sigma}_v\}^{-1}\mathbf{C}[\mathbf{P}(k),\mathbf{x}(k{+}1)] \tag{4.57}$$

$$\mathbf{Q} = \boldsymbol{\Sigma}_j(k) + \boldsymbol{\Sigma}_v \tag{4.58}$$

Therefore, the evaluation of the DP-GP expected KL value function becomes

$$\begin{aligned}&\varphi_j[\boldsymbol{\phi};\mathbf{m}(k+1) \mid \mathcal{M}_j(k),\mathcal{E}(k),\mathbf{u}(k)]\\ &= \sum_{i=1}^{N} w_{ji} \int_{\mathcal{S}[\mathbf{u}(k+1)]} h[\mathbf{x}_j(k+1^-)]\; p[\mathbf{x}_j(k+1^-)|\mathcal{M}_j(k),\mathcal{E}_i(k)] d\mathbf{x}_j(k{+}1^-),\end{aligned} \tag{4.59}$$

and it is maximized with respect to $\mathbf{u}(k+1)$ to optimize the sensor performance. Unfortunately, the remaining integral of $h_i[\mathbf{x}_j(k+1)]$ over $\mathcal{S}[\mathbf{u}(k+1)]$ does not have a closed form because $\mathbf{x}_j(k+1)$ is contained in the covariance matrix in (4.57). This remaining integral can be evaluated by discretizing the integral domain $\mathcal{S}[\mathbf{u}(k+1)]$. Note that the computational complexity in calculating $h_i[\mathbf{x}_j(k+1)]$ is $O(n_i^3)$, where n_i is the dimensionality of $\mathbf{P}_i(k)$ and is usually large. Therefore, it is infeasible to obtain the optimal control by evaluating (4.59) for all $\mathbf{u}(k+1) \in \mathbb{R}^2$. An alternative approach to reduce the computational burden is to discretize the entire workspace $\mathcal{W}$ and evaluate the integrand in (4.59) at each grid once. Then, for any control $\mathbf{u}(k+1)$, the integral in (4.59) is approximated by multiplying the grid area with the summation of the integrand values at grids inside $\mathcal{S}[\mathbf{u}(k+1)]$.

However, the integrand value at $\mathbf{x}_j(k+1)$ is almost zero when the probability of target j being at $\mathbf{x}_j(k+1)$, i.e. (3.27), goes to 0 and $h_i[\mathbf{x}_j(k+1)]$ is bounded . Then, it is not necessary to calculate the integrand values at locations where (3.27) goes to 0. Therefore, by using the weighted particles, $\mathcal{P}_{ji}(k+1^-)$, Monte Carlo integration

is utilized to evaluate (4.59):

$$\hat{\varphi}_j[\boldsymbol{\phi}; \mathbf{m}(k{+}1) \mid \mathcal{M}_j(k), \mathcal{E}(k), \mathbf{u}(k)] \approx \sum_{i=1}^{N} \sum_{\boldsymbol{\chi}_{jis}(k{+}1^-) \in \mathcal{S}(k{+}1)} h[\boldsymbol{\chi}_{jis}(k{+}1^-)]\ \omega_{jis}(k{+}1^-), \quad (4.60)$$

where $\mathcal{S}[\mathbf{u}(k{+}1)]$ is abbreviated as $\mathcal{S}(k{+}1)$. Then, the total information value from all targets can be approximated as

$$\hat{\varphi}[\boldsymbol{\phi}; \mathbf{m}(k+1)] = \sum_{j=1}^{M} \hat{\varphi}_j[\boldsymbol{\phi}; \mathbf{m}_j(k+1)] \approx \sum_{j=1}^{M} \sum_{i=1}^{N} \frac{w_{ji}}{S} \sum_{\boldsymbol{\chi}_{jis}(k{+}1^-) \in \mathcal{S}(k{+}1)} h_i[\mathbf{x}_j^{(s)}(k+1)]. \quad (4.61)$$

We have the following theorem with respect to (4.61).

Theorem 1. *The Monte Carlo integration (4.60) is an unbiased estimator of the DP-GP expected KL divergence (4.59), and the variance of the error between (4.60) and (4.59) decreases linearly with* $1/S$.

The proof of the above theorem is given for the case when $\boldsymbol{\Sigma}_v = \sigma_v^2 \mathbf{I}_{2\times 2}$, with the help of three lemmas. The first gives the element-wise boundary of matrix $\mathbf{Q}$, the second shows the relationship between $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$, and the third gives the bound of $\mathrm{tr}(\mathbf{R}^T\mathbf{R})$. The proof for the case with varying diagonal entries will be similar.

Lemma 1. $\frac{\sigma_v^2(1+k+\sigma_v^2)}{k+\sigma_v^2}\mathbf{I}_{2\times 2} \preceq \mathbf{Q} \preceq (1+\sigma_v^2)\mathbf{I}_{2\times 2}$, *where* $\preceq$ *denotes element-wise comparison.*

Theorem 1. First, we prove that $\mathbf{Q} \preceq (1+\sigma_v^2)\mathbf{I}_{2\times 2}$. Defining $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{D}$ as

$$\boldsymbol{\Sigma}_2^{-1} = \begin{bmatrix} \mathbf{C}[\mathbf{P}_i(k), \mathbf{P}_i(k)] + \boldsymbol{\Sigma}_v & \mathbf{C}[\mathbf{P}_i(k), \mathbf{x}_j(k+1)] \\ \mathbf{C}[\mathbf{x}_j(k+1), \mathbf{P}_i(k)] & \mathbf{C}[\mathbf{x}_j(k+1), \mathbf{x}_j(k+1)] + \boldsymbol{\Sigma}_v \end{bmatrix}^{-1} \quad (4.62)$$

$$\triangleq \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B} & \mathbf{D} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}\mathbf{Q}^{-1}\mathbf{B}^T\mathbf{A}^{-1} & -\mathbf{A}^{-1}\mathbf{B}\mathbf{Q}^{-1} \\ \mathbf{Q}^{-1}\mathbf{B}^T\mathbf{A}^{-1} & \mathbf{Q}^{-1} \end{bmatrix}, \quad (4.63)$$

it follows that

$$\mathbf{Q} = \mathbf{D} - \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B}. \quad (4.64)$$

The addition of the symmetric positive semi-definite covariance matrix $\mathbf{C}[\mathbf{P}_i(k), \mathbf{P}_i(k)]$, and the symmetric positive definite regulation term $\sigma_v^2\mathbf{I}$, results in the symmetric positive definite matrix $\mathbf{A}$. Therefore, $\mathbf{A}^{-1}$ is well-defined and is also positive definite. Hence, $\mathbf{0}_{2\times 2} \preceq \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B}$ for all $\mathbf{B} \in \mathbb{R}^{k\times 2}$, which shows that $\mathbf{Q} \preceq \mathbf{D}$. Note that the covariance matrix $\mathbf{C}[\mathbf{x}_j(k), \mathbf{x}_j(k)] = \mathbf{I}_{2\times 2}$ for all $\mathbf{x}_j(k+1) \in \mathbb{R}^2$. Then, by assuming the x and y directions of the velocity fields are independent, it is true that $\mathbf{D} = (1+\sigma_v^2)\mathbf{I}_{2\times 2}$ and that

$$\mathbf{Q} \preceq (1+\sigma_v^2)\mathbf{I}_{2\times 2}. \tag{4.65}$$

Second, we prove that $\frac{\sigma_v^2(1+k+\sigma_v^2)}{k+\sigma_v^2}\mathbf{I}_{2\times 2} \preceq \mathbf{Q}$. Because the covariance $\mathbf{C}[\mathbf{P}_i(k), \mathbf{P}_i(k)]$ is real, symmetric, and positive semi-definite, there exists an eigenvalue decomposition, $\mathbf{C}[\mathbf{P}_i(k), \mathbf{P}_i(k)] = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}$, with orthogonal eigenvectors such that $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, where

$$\mathbf{\Lambda} \triangleq \operatorname{diag}[\lambda_1, \cdots, \lambda_k]^T \quad \text{and} \quad \mathbf{U} \triangleq [\mathbf{u}_1 \quad \dots \quad \mathbf{u}_k]^T. \tag{4.66}$$

Because $\mathbf{x}_j(k)$ is the same as the last column of $\mathbf{P}_i(k)$, it is true that

$$\mathbf{B} = \mathbf{X}\mathbf{\Lambda}\mathbf{u}_k. \tag{4.67}$$

where $\mathbf{u}_k = [a_1 \quad \dots \quad a_k]^T$. Substituting (4.67) into (4.64), $\mathbf{Q}$ is rewritten in terms of the eigenvalues as follows:

$$\begin{aligned}
\mathbf{Q} &= (1+\sigma_v^2)\mathbf{I} - (\mathbf{U}\mathbf{\Lambda}\mathbf{u}_k)^T\mathbf{A}^{-1}(\mathbf{U}\mathbf{\Lambda}\mathbf{u}_k) \\
&= (1+\sigma_v^2)\mathbf{I} - \mathbf{u}_k^T\mathbf{\Lambda}\mathbf{U}^T\mathbf{U}(\mathbf{\Lambda}+\sigma_v^2\mathbf{I})^{-1}\mathbf{U}^T\mathbf{U}\mathbf{\Lambda}\mathbf{u}_k \\
&= \left(1+\sigma_v^2 - \sum_{\ell=1}^{k}\frac{\lambda_\ell}{\lambda_\ell+\sigma_v^2}\lambda_\ell a_\ell^2\right)\mathbf{I}.
\end{aligned} \tag{4.68}$$

Because $\mathbf{C}[\mathbf{P}_i(k), \mathbf{P}_i(k)]$ is symmetric and positive semi-definite, all its eigenvalues are greater than or equal to zero: $\lambda_\ell \geqslant 0$, $\ell = 1, \dots, k$. Moreover, the fact that $\mathbf{u}_k^T\mathbf{\Lambda}\mathbf{u}_k = 1$ yields

$$\sum_{\ell=1}^{k}\lambda_\ell a_\ell^2 = 1. \tag{4.69}$$

Substituting (4.69) into (4.68) yields the following inequality,

$$\mathbf{Q} \geq \left[1 + \sigma_v^2 - \max\left(\frac{\lambda_\ell}{\lambda_\ell + \sigma_v^2}\right)\right]\mathbf{I} = \left[1 + \sigma_v^2 - \frac{\max(\lambda_\ell)}{\max(\lambda_\ell) + \sigma_v^2}\right]\mathbf{I}. \tag{4.70}$$

For a real and symmetric matrix, the trace equals the sum of the eigenvalues:

$$\mathrm{tr}\{\mathbf{C}[\mathbf{P}_i(k), \mathbf{P}_i(k)]\} = \sum_{\ell=1}^{k} \lambda_\ell. \tag{4.71}$$

Because the diagonal elements of $\mathbf{C}[\mathbf{P}_i(k), \mathbf{P}_i(k)]$ all equal one, it is true that $\sum_{\ell=1}^{k} \lambda_\ell = k$. Recalling that $\lambda_\ell \geqslant 0$, the maximum value of the eigenvalue is bounded by k: $\max\{\lambda_\ell\} \leqslant k$. Therefore, the entries in matrix $\mathbf{Q}$ are also bounded from below,

$$\mathbf{0} \preceq \left[1 + \frac{\sigma_v^2}{k + \sigma_v^2}\right]\mathbf{I} \preceq \mathbf{Q}, \tag{4.72}$$

which completes the proof of Lemma 1. □

Lemma 2. $0 \prec \mathbf{\Sigma}_2 \preceq \mathbf{\Sigma}_1$.

Proof. Because $\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_2$ are the covariance matrices for two conditional Gaussian distributions obtained from a Gaussian process with a non-zero noise term, they are positive definite. From the block inversion of $\mathbf{\Sigma}_2$, it is true that

$$\mathbf{\Sigma}_1 - \mathbf{\Sigma}_2 = \mathbf{R}\mathbf{Q}^{-1}\mathbf{R}^T. \tag{4.73}$$

From Lemma 1, $\left[1 + \frac{\sigma_v^2}{k+\sigma_v^2}\right]\mathbf{I} \preceq \mathbf{Q}$. Because $\mathbf{Q}$ is a diagonal matrix and the x and y directions of the velocity fields are assumed to be independent, it is sufficient to claim that $\mathbf{Q}$ is positive definite. Therefore, $\mathbf{\Sigma}_2 \preceq \mathbf{\Sigma}_1$. □

Lemma 3. $0 \leqslant \mathrm{tr}(\mathbf{R}\mathbf{R}^T) \leqslant 4k$.

Proof. First, we prove that $0 \leqslant \mathrm{tr}(\mathbf{R}^T\mathbf{R})$. Because $\mathbf{R}^T\mathbf{R}$ is a positive semi-definite matrix, its eigenvalues are either positive or zero. Utilizing the property that the

trace of a positive semi-definite matrix is equal to the summation of its eigenvalues, it can be shown that

$$0 \leqslant \mathrm{tr}(\mathbf{R}\mathbf{R}^T). \tag{4.74}$$

Second, we show the existence of the upper bound. Recalling the definition of $\mathbf{R}$ in (4.57) and that the covariance function (3.23) is non-negative, it is true that

$$\mathbf{R} \preceq \mathbf{C}[\mathbf{X}, \mathbf{x}(k+1)]. \tag{4.75}$$

From the property that the trace of a product can be rewritten as the sum of entry-wise products of elements, it is true that

$$\mathrm{tr}(\mathbf{R}\mathbf{R}^T) = \sum_{ij} (\mathbf{R}^T \circ \mathbf{R}^T)_{ij}, \tag{4.76}$$

where $\circ$ denotes the Hadamard product [116]. Observing that all of the elements in $\mathbf{R}$ are outputs from the covariance function (3.23), it can be claimed that the elements are less than or equal to one. Therefore, $(\mathbf{R}^T \circ \mathbf{R}^T) \preceq \mathbf{I}_{2k\times 2}$, which shows that $\mathrm{tr}(\mathbf{R}\mathbf{R}^T) \leqslant 4k$. □

Proof. From the linearity of the expectation, it is true that

$$\mathbb{E}\left\{ \sum_{j=1}^{M} \sum_{i=1}^{N} \frac{w_{ji}}{S} \sum_{s=1}^{S} h_i\big[\mathbf{x}_j^{(s)}(k+1)\big] \right\} = \sum_{j=1}^{M} \sum_{i=1}^{N} \frac{w_{ji}}{S} \sum_{s=1}^{S} \mathbb{E}\left\{ h_i\big[\mathbf{x}_j^{(s)}(k+1)\big] \right\}. \tag{4.77}$$

Because the samples $\mathbf{x}_j^{(s)}(k+1)$, where $s = 1, \ldots, S$, are drawn identically and independently from the predicted target position distribution (3.27), it is true that

$$\mathbb{E}\left\{ h_i\big[\mathbf{x}_j^{(s)}(k+1)\big] \right\} = \mathbb{E}\left\{ h_i\big[\mathbf{x}_j(k+1)\big] \right\}, \ \forall s = 1, \ldots, S. \tag{4.78}$$

Therefore,

$$\mathbb{E}\left\{ \sum_{j=1}^{M} \sum_{i=1}^{N} \frac{w_{ji}}{S} \sum_{s=1}^{S} h_i\big[\mathbf{x}_j^{(s)}(k+1)\big] \right\} = \hat{\varphi}[\boldsymbol{\phi}; \mathbf{m}(k+1)], \tag{4.79}$$

which means that the estimator (4.61) is unbiased.

Let var$(\cdot)$ denote the variance of a random variable. Then, because the samples are drawn independently and identically, the following statement is true:

$$\text{var}\left\{\sum_{j=1}^{M}\sum_{i=1}^{N}\frac{w_{ji}}{S}\sum_{s=1}^{S}h_i\big[\mathbf{x}_j^{(s)}(k+1)\big]\right\}=\frac{1}{S}\text{var}\left\{\sum_{j=1}^{M}\sum_{i=1}^{N}w_{ji}h_i\big[\mathbf{x}_j(k+1)\big]\right\}. \tag{4.80}$$

Therefore, the variance of the estimator (4.61) decrease linearly with the inverse of the number of samples [14] if the value of $h_i[\mathbf{x}_j(k+1)]$ is bounded for all $\mathbf{x}_j \in \mathbb{R}^2$. The bound of $h_i[\mathbf{x}_j(k+1)]$ is determined by examining the upper bound and lower bound of every term in $h_i[\mathbf{x}_j(k+1)]$ as follows. First, we prove that $h_i[\mathbf{x}_j(k+1)]$ is lower-bounded by zero. Recall that h_i is calculated by integrating the weighted KL divergence by a Gaussian distribution. Because $D \geqslant 0$ and $\mathcal{N}[\boldsymbol{\mu}_j(k+1), \sigma_v^2\mathbf{I}_2] \geqslant 0$, the integral is greater than or equal to zero, and thus

$$h_i[\mathbf{x}_j(k+1)] \geqslant 0. \tag{4.81}$$

With the help of the three lemmas, we can now prove that $h_i[\mathbf{x}_j(k+1)]$ is also bounded from above. From Lemma 2, it is true that

$$\text{tr}(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_2)=\text{tr}[\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{\Sigma}_1-\mathbf{R}\mathbf{Q}^{-1}\mathbf{R}^T)]=\text{tr}(\mathbf{I}_{2L\times 2L}-\boldsymbol{\Sigma}_1\mathbf{R}\mathbf{Q}^{-1}\mathbf{R}^T)\leqslant 2L. \tag{4.82}$$

Also from Lemma 2, it follows that $0 < |\boldsymbol{\Sigma}_1|$ and $0 < |\boldsymbol{\Sigma}_2|$, because the two matrices are both positive definite. Because $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ are Hermitian and $\boldsymbol{\Sigma}_1 > \boldsymbol{\Sigma}_2$, $|\boldsymbol{\Sigma}_1| > |\boldsymbol{\Sigma}_2|$, it also follows that $0 < |\boldsymbol{\Sigma}_2|/|\boldsymbol{\Sigma}_1| < 1$, which means that $0 < -\ln(|\boldsymbol{\Sigma}_2|/|\boldsymbol{\Sigma}_1|) < \infty$.

By adopting the concept of the Frobenius norm (which can be viewed as the Euclidean norm if the matrix is treated as a vector), the Cauchy-Schwarz inequality can be utilized to prove that the trace of the products of positive semi-definite matrices of the same size is less than or equal to the product of traces:

$$\text{tr}[\mathbf{Q}^{-1}\mathbf{R}^T\boldsymbol{\Sigma}_1^{-1}\mathbf{R}\mathbf{Q}^{-1}]\leqslant\text{tr}(\mathbf{Q}^{-1})\text{tr}(\mathbf{R}^T\boldsymbol{\Sigma}_1^{-1}\mathbf{R})\text{tr}(\mathbf{Q}^{-1})=\text{tr}(\mathbf{Q}^{-1})^2\text{tr}(\boldsymbol{\Sigma}_1^{-1}\mathbf{R}\mathbf{R}^T)$$

$$\leqslant \mathrm{tr}(\mathbf{Q}^{-1})^2 \mathrm{tr}(\mathbf{R}\mathbf{R}^T)\mathrm{tr}(\mathbf{\Sigma}_1^{-1}) \leqslant 4k \left[\frac{k+\sigma_v^2}{\sigma_v^2(1+k+\sigma_v^2)}\right]^2 \mathrm{tr}(\mathbf{\Sigma}_1^{-1}) \tag{4.83}$$

Therefore, from (4.81) and (4.83), h_i is shown to be bounded both from below and from above:

$$0 \leqslant h_i[\mathbf{x}_j(k+1)] < 4k \left[\frac{k+\sigma_v^2}{\sigma_v^2(1+k+\sigma_v^2)}\right]^2 \mathrm{tr}(\mathbf{\Sigma}_1^{-1})\mathrm{tr}(\Sigma_v)/2, \tag{4.84}$$

Hence, according to [14], Theorem 1 is true and the decrease rate of the variance of the Monte Carlo estimator in (4.61) is proportional to $1/S$. □

4.5.3 Strategy for Searching the Optimal Coverage

Utilizing all of the weighted particles, the DP-GP expected KL information value function to be maximized at each step can be written as

$$J = \sum_{j=1}^{M}\sum_{i=1}^{N} \sum_{\boldsymbol{\chi}_{jis}(k+1^-)\notin\mathcal{S}(k+1)} h[\boldsymbol{\chi}_{jis}(k+1^-)] w_{jis}(k+1^-) \tag{4.85}$$

where $h[\boldsymbol{\chi}_{jis}(k+1^-)]$ is precalculated for every particle. Given a zoom level, the optimal control $\mathbf{u}^*(k+1)$ is obtained by the reduction to the following geometric covering problem: Given a set of NMS points,

$$\cup_{j=1}^{M} \cup_{i=1}^{N} \cup_{s=1}^{S} \{\boldsymbol{\chi}_{jis}(k+1^-)\},$$

in a two dimensional plane, each associated with a weight,

$$\omega_{jis}(k+1^-) \times h[\boldsymbol{\chi}_{jis}(k+1^-)], \tag{4.86}$$

find the position of an axis-parallel rectangle of given size L_x (horizontal size) and L_y (vertical size) such that J in (4.85) is maximized [5].

Analogous to the method in [5], the approach optimizing $\mathbf{u}^*(k+1)$ consists of five steps. In Step 1, the x coordinates of entire particles are sorted for the sweeping step (Step 4). Step 2 builds a segment tree in $O(MNS \log MNS)$ time for all vertical

segments of length L_y with their bottom edges at the y coordinates of the particles; In Step 3, a value variable is associated with each vertical segment and is initialized at zero; In the 4th step, the sorted x coordinates are swept with a horizontal segment of length L_x and infinite height, and added (deleted) particles whose x coordinates are newly covered (newly uncovered) by this horizontal segment are recorded. Values associated with the retrieved vertical segments that contain y coordinates of added (deleted) particles are added (subtracted) with the weights associated with these particles. In the last step, the maximum value of all of the updated segments is obtained. Steps 1 through 5 are repeated for all zoom-levels in order to obtain the maximum value.

The search for the optimal $\mathbf{u}^*(k+1)$ for each zoom level at each time step can be done in $O(MNS \log MNS + K)$ time, where K is the total number of retrieved segments. It is worth pointing out that, because the measurement noise and the size of the FOV is determined by the zoom level, the pre-calculated quantities stay the same for different zoom levels except for $h[\boldsymbol{\chi}_{jis}(k+1^-)]$. A geometric covering problem is formulated and solved for each zoom level. Finally, the optimal control for each zoom level is compared in time $O(|\mathcal{L}|)$ to obtain the optimal control that maximizes (4.85), where $|\mathcal{L}|$ denotes the number of zoom levels.

4.6 Optimized Visibility Motion Planning

In many cases, the ability to track and localize the target is limited by the absence of a GPS and by the presence of a bounded FOV, which may cause the sensor to lose the target completely. These difficulties are exacerbated by the need for tracking moving targets in complex unstructured environments. Furthermore, because the position and orientation of the sensor FOV is determined by the control inputs, the motion of the robotic sensor must be planned in concert with its measurement sequence for both the sensing and the navigation objective to be optimized. To address the these

issues, an optimized-visibility motion planning approach is proposed using the EKF to simultaneously track the target and localize the robotic sensor [120, 118, 121]. Note that the proposed optimized-visibility approach is applicable to any robot that is equipped with exteroceptive sensors, such as a laser scanner or camera, for tracking and localizing moving targets and with proprioceptive sensors, such as an odometer, for providing ego-motion information.

In GPS-denied environments, the quality of the target state estimates depends on the estimates of the robot state obtained from the EKF algorithm. Therefore, the overall target tracking and localization performance can be represented by the expected power of the error between the true and estimated robot-target joint states. For sensors with a limited FOV, it is possible to obtain an empty target measurement. Therefore, two situations are considered for calculating the joint state error: when the target measurement is available, the posterior estimate of the joint state, $\hat{\mathbf{q}}(k+1|k+1)$, is used; otherwise, the prior estimate, $\hat{\mathbf{q}}(k+1|k)$, is employed. Let P_d denote the probability that the target measurement is available. The error power can be obtained as follows:

$$\begin{aligned} J[\mathbf{u}_r(k)] = \mathbb{E}\left[\mathbf{e}(k+1|k)^T\mathbf{e}(k+1|k)\right] \times (1-P_d) \\ + \mathbb{E}\left[\mathbf{e}(k+1|k+1)^T\mathbf{e}(k+1|k+1)\right] \times P_d, \end{aligned} \tag{4.87}$$

where $\mathbb{E}(\cdot)$ denotes the expectation and

$$\mathbf{e}(k+1|k) = \mathbf{q}(k) - \hat{\mathbf{q}}(k+1|k), \tag{4.88}$$

$$\mathbf{e}(k+1|k+1) = \mathbf{q}(k) - \hat{\mathbf{q}}(k+1|k+1). \tag{4.89}$$

The probability P_d can be obtained by integrating the target state distribution over the robotic sensor's FOV:

$$P_d[\mathbf{q}_r(k)] = \int\limits_{\mathcal{S}[\mathbf{q}_r(k)]} f_t[\mathbf{x}_t(k)]d\mathbf{x}_t(k), \tag{4.90}$$

where $f_t[\mathbf{x}_t(k)]$ is the probability density function of target state distribution. Also, $f_t[\mathbf{x}_t(k)]$ can be approximated by a Gaussian distribution, as follows:

$$\begin{aligned} f_t[\mathbf{x}_t(k)] &= \mathcal{N}[\mathbf{x}_t(k); \hat{\mathbf{x}}_t(k|k-1), \mathbf{P}_t(k|k-1)] \\ &\triangleq \mathcal{N}[\mathbf{x}_t(k); \boldsymbol{\mu}_t(k), \boldsymbol{\Sigma}_t(k)], \end{aligned} \tag{4.91}$$

where $\boldsymbol{\mu}_t(k)$ and $\boldsymbol{\Sigma}_t(k)$ are introduced to simplify notations. Then (4.87) can be rewritten as

$$J[\mathbf{u}_r(k)] = \mathrm{tr}[\mathbf{P}(k+1|k)] - P_d\big\{\mathrm{tr}[\mathbf{P}(k+1|k)] - \mathrm{tr}[\mathbf{P}(k+1|k+1)]\big\},$$

where $\mathrm{tr}(\cdot)$ denotes the trace of a matrix. Because the propagation step of EKF only produces a prior estimate of the joint state, here we focus on controlling the robot to obtain the most informative measurements to reduce the uncertainty of the joint state. Therefore, it is assumed that the prior estimates are optimal with respect to the robot control. In addition, for the priori and posteriori state estimates in the EKF, it is true that

$$\mathrm{tr}[\mathbf{P}(k+1|k)] - \mathrm{tr}[\mathbf{P}(k+1|k+1)] \geqslant 0. \tag{4.92}$$

As a result, minimizing the error of the joint state (4.87) can be achieved by maximizing the probability of detection (4.90), and the robot controller can be obtained by solving the following constrained optimization problem in $\mathbf{u}_r(k)$:

$$\max_{\mathbf{u}_r(k)} \; P_d[\mathbf{q}_r(k+1)] \tag{4.93}$$

$$\text{s.t.} \;\; \mathbf{q}_r(k+1) = \mathbf{q}_r(k) + \mathbf{B}_r(k)\mathbf{u}_r(k)\delta t \tag{4.94}$$

4.6.1 EKF for Tracking and Localization

In this type of problem, the robot and the target state must be simultaneously estimated from data. Therefore, consider an augmented state vector containing both

the robot and the target state and the augmented control vector, respectively, given by

$$\mathbf{q}(k) = [\mathbf{q}_r^T(k) \quad \mathbf{q}_t^T(k)]^T, \tag{4.95}$$

and

$$\mathbf{u}(k) = [\mathbf{u}_r^T(k) \quad \mathbf{0}]^T. \tag{4.96}$$

Then, based on the robot state propagation equation (2.11) and the target state propagation equation (2.10), the joint state propagation of the robot and the target is

$$\mathbf{q}(k+1) = \mathbf{f}[\mathbf{q}(k), \mathbf{u}(k), k] = \begin{bmatrix} \mathbf{f}_r[\mathbf{q}_r(k), \mathbf{u}_r(k), k] \\ \mathbf{f}_t[\mathbf{q}_t(k)] \end{bmatrix}, \tag{4.97}$$

and the Jacobian matrix of the state transition function for the joint state is

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{\Phi}_r(k) & \mathbf{0} \\ \mathbf{0} & \mathbf{\Phi}_t \end{bmatrix}, \tag{4.98}$$

where $\mathbf{\Phi}_t$ is the target state transition matrix and

$$\mathbf{\Phi}_r(k) \triangleq \frac{\partial}{\partial \mathbf{q}_r(k)}\{\mathbf{f}_r[\mathbf{q}_r(k), \mathbf{u}_r(k), k]\} = \begin{bmatrix} 1 & 0 & -\sin\theta_r(k) v_r(k)\delta t \\ 0 & 1 & \cos\theta_r(k) v_r(k)\delta t \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.99}$$

Similar to the classical Kalman filer, the EKF posterior estimates are taken from [108]:

$$\tilde{\mathbf{y}}(k+1) = \begin{bmatrix} \mathbf{z}_r^T(k+1) \\ \mathbf{z}_l^T(k+1) \end{bmatrix} - \mathbf{h}[\hat{\mathbf{q}}(k+1|k)], \tag{4.100}$$

$$\mathbf{S}(k+1) = \mathbf{H}(k+1)\mathbf{P}(k+1|k)\mathbf{H}(k+1)^T + \begin{bmatrix} \mathbf{R}_r & \mathbf{0} \\ \mathbf{0} & \mathrm{diag}([\mathbf{R}_t \quad \dots \quad \mathbf{R}_t]) \end{bmatrix}, \tag{4.101}$$

$$\mathbf{K}(k+1) = \mathbf{P}(k+1|k)\mathbf{H}(k+1)^T\mathbf{S}^{-1}(k+1), \tag{4.102}$$

$$\hat{\mathbf{q}}(k+1|k+1) = \hat{\mathbf{q}}(k+1|k) + \mathbf{K}(k+1)\tilde{\mathbf{y}}(k+1), \tag{4.103}$$

$$\mathbf{P}(k+1|k+1) = [\mathbf{I} - \mathbf{K}(k+1)\mathbf{H}(k+1)]\mathbf{P}(k+1|k). \tag{4.104}$$

Here, diag$(\cdot)$ denotes the square diagonal matrix with the blocks of matrices on the main diagonal. In the above EKF, the prediction of the joint state and its covariance before the measurements is given by

$$\hat{\mathbf{q}}(k+1|k) = \mathbf{f}[\hat{\mathbf{q}}(k|k), \mathbf{u}(k), k], \tag{4.105}$$

$$\mathbf{P}(k+1|k) = \mathbf{\Phi}\mathbf{P}(k|k)\mathbf{\Phi}^T + \begin{bmatrix} \mathbf{B}_r(k)\mathbf{Q}_r\mathbf{B}_r^T(k) & \mathbf{0} \\ \mathbf{0} & \mathbf{G}\mathbf{Q}_t\mathbf{G}^T \end{bmatrix}, \tag{4.106}$$

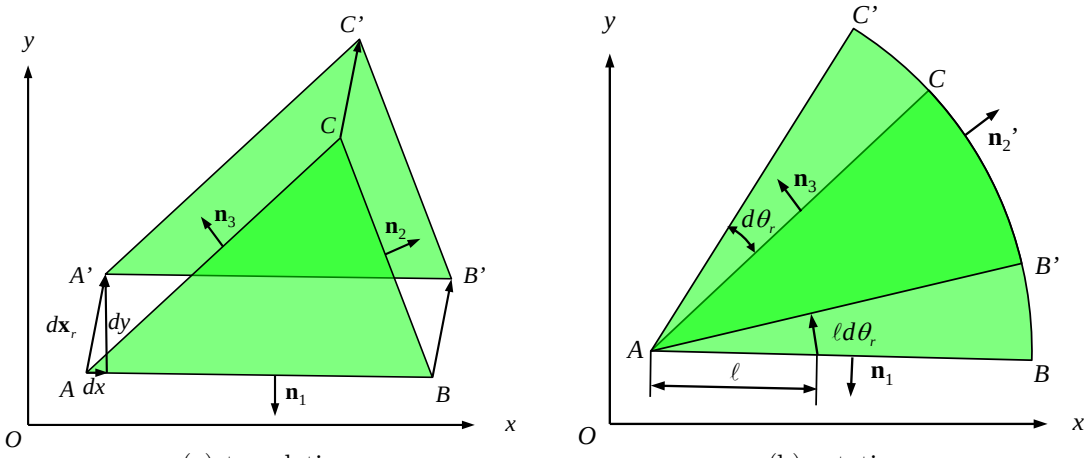
and the Jacobian matrix of the measurement function $\mathbf{h} \triangleq [\mathbf{h}_t^T \quad \mathbf{h}_l^T]^T$ is given by [120]:

$$\mathbf{H}(k) = \begin{bmatrix} \frac{2(x_r-x_t)}{\|\mathbf{x}_r-\mathbf{x}_t\|} & \frac{2(y_r-y_t)}{\|\mathbf{x}_r-\mathbf{x}_t\|} & 0 & \frac{2(x_t-x_r)}{\|\mathbf{x}_r-\mathbf{x}_t\|} & \frac{2(y_t-y_r)}{\|\mathbf{x}_r-\mathbf{x}_t\|} & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ \frac{2(x_r-x_1)}{\|\mathbf{x}_r-\mathbf{x}_{l_1}\|} & \frac{2(y_r-y_1)}{\|\mathbf{x}_r-\mathbf{x}_{l_1}\|} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ & & \vdots & & & \\ \frac{2(x_r-x_L)}{\|\mathbf{x}_r-\mathbf{x}_{l_L}\|} & \frac{2(y_r-y_L)}{\|\mathbf{x}_r-\mathbf{x}_{l_L}\|} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}. \tag{4.107}$$

4.6.2 Robot Motion Planning

The robot controller considers the tracking and localization performance while maintaining a low computational complexity; thus, realtime implementation can be obtained as follows. At any discrete time step k, the robot control inputs are computed to maximize the probability of detection at the next time step, $(k+1)$, subject to the robot kinematics. Similar to potential methods, the solution of the constrained optimization problem (4.93-4.94) can be obtained by moving in the direction of the adjoined gradient, which can be obtained analytically, thus providing the controller in closed form. As a first step, the Jacobian for (4.93) can be written as

$$\frac{\partial P_d[\mathbf{q}_r(k+1)]}{\partial \mathbf{u}_r(k)} = \frac{\partial P_d[\mathbf{q}_r(k+1)]}{\partial \mathbf{q}_r(k+1)} \frac{\partial \mathbf{q}_r(k+1)}{\partial \mathbf{u}_r(k)}, \tag{4.108}$$



(a) translation (b) rotation

FIGURE 4.6: Illustration of the changes of the robot FOV due to the translation and rotation of the robot.

where

$$\frac{\partial P_d[\mathbf{q}_r(k+1)]}{\partial \mathbf{q}_r(k+1)} = \begin{bmatrix} \frac{\partial}{\partial x_r}\{P_d[\mathbf{q}_r(k+1)]\} \\ \frac{\partial}{\partial y_r}\{P_d[\mathbf{q}_r(k+1)]\} \\ \frac{\partial}{\partial \theta_r}\{P_d[\mathbf{q}_r(k+1)]\} \end{bmatrix}. \tag{4.109}$$

Each term of the above expression (4.109) is computed by derivatives of integrals, as shown in Fig. 4.6.

The first entry of (4.109) can be calculated as follows:

$$\frac{\partial}{\partial x_r} P_d[\mathbf{q}_r(k+1)] = \frac{\partial}{\partial x_r} \int_{\mathcal{S}[\mathbf{q}_r(k+1)]} f_t(\mathbf{x}_t) d\mathbf{x}_t = \int_{\mathcal{S}[\mathbf{q}_r(k+1)]} \frac{\partial}{\partial x_r} f_t(\mathbf{x}_t) d\mathbf{x}_t + \oint_{\partial\mathcal{S}[\mathbf{q}_r(k+1)]} (\mathbf{v}_x \mathbf{n}) f_t(\mathbf{x}_t) d\mathbf{x}_t, \tag{4.110}$$

where $\partial\mathcal{S}$ denotes the boundary of $\mathcal{S}$, $\mathbf{v}_x = [1\ 0]^T$ is the velocity of the robot FOV, and $\mathbf{n}$ represents the outward-pointing unit normal vector along the boundary of the robot FOV. The outward-pointing unit normal vectors can be obtained through the heading of the robot and the opening angle of the robot FOV, as shown in Fig. 4.6

(a). Thus, we have

$$\begin{aligned}
\mathbf{n}_1 &= \mathbf{C}(\pi/2 + \alpha/2)[\cos\theta_r \quad \sin\theta_r]^T, \\
\mathbf{n}_2 &= [\cos\theta_r \quad \sin\theta_r]^T, \\
\mathbf{n}_3 &= \mathbf{C}(-\pi/2 - \alpha/2)[\cos\theta_r \quad \sin\theta_r]^T,
\end{aligned} \tag{4.111}$$

where $\mathbf{C}(\cdot)$ is the 2×2 rotation matrix. Moreover, because $f_t(\mathbf{x}_t)$ is not a function of $\mathbf{q}_r$, we have

$$\frac{\partial}{\partial \mathbf{q}_r} f_t(\mathbf{x}_t) = 0. \tag{4.112}$$

In addition, (4.91) contains the analytical form of f_t. Then, the partial derivative in (4.110) can be simplified as follows:

$$\begin{aligned}
&\frac{\partial}{\partial x_r} P_d[\mathbf{q}_r(k+1)] \\
&= \int_{AB} (\mathbf{v}_x \cdot \mathbf{n}_1) \exp[-\frac{1}{2}(\mathbf{x}_t - \boldsymbol{\mu}_t)^T \boldsymbol{\Sigma}_t^{-1} (\mathbf{x}_t - \boldsymbol{\mu}_t)] d\mathbf{x}_t \\
&+ \int_{BC} (\mathbf{v}_x \cdot \mathbf{n}_2) \exp[-\frac{1}{2}(\mathbf{x}_t - \boldsymbol{\mu}_t)^T \boldsymbol{\Sigma}_t^{-1} (\mathbf{x}_t - \boldsymbol{\mu}_t)] d\mathbf{x}_t \\
&+ \int_{CA} (\mathbf{v}_x \cdot \mathbf{n}_3) \exp[-\frac{1}{2}(\mathbf{x}_t - \boldsymbol{\mu}_t)^T \boldsymbol{\Sigma}_t^{-1} (\mathbf{x}_t - \boldsymbol{\mu}_t)] d\mathbf{x}_t.
\end{aligned} \tag{4.113}$$

Similarly, the second entry of (4.109) can be obtained in the process, except that

$\mathbf{v}_y = [0\ 1]^T$:

$$\begin{aligned}
&\frac{\partial}{\partial y_r} P_d[\mathbf{q}_r(k+1)] \\
&= \int_{AB} (\mathbf{v}_y \cdot \mathbf{n}_1) \exp[-\frac{1}{2}(\mathbf{x}_t - \boldsymbol{\mu}_t)^T \boldsymbol{\Sigma}_t^{-1}(\mathbf{x}_t - \boldsymbol{\mu}_t)] d\mathbf{x}_t \\
&+ \int_{BC} (\mathbf{v}_y \cdot \mathbf{n}_2) \exp[-\frac{1}{2}(\mathbf{x}_t - \boldsymbol{\mu}_t)^T \boldsymbol{\Sigma}_t^{-1}(\mathbf{x}_t - \boldsymbol{\mu}_t)] d\mathbf{x}_t \\
&+ \int_{CA} (\mathbf{v}_y \cdot \mathbf{n}_3) \exp[-\frac{1}{2}(\mathbf{x}_t - \boldsymbol{\mu}_t)^T \boldsymbol{\Sigma}_t^{-1}(\mathbf{x}_t - \boldsymbol{\mu}_t)] d\mathbf{x}_t.
\end{aligned} \tag{4.114}$$

Let sign$(\cdot)$ denote the sign function and ℓ denote the distance from a point on the boundary of the FOV to the point A. Then, the third entry of (4.109) can also be calculated analytically as follows:

$$\begin{aligned}
&\frac{\partial}{\partial \theta_r} P_d[\mathbf{x}_r(k+1)] \\
&= \int_{AB} -\text{sign}(d\theta_r)\ell \exp[-\frac{1}{2}(\mathbf{x}_t - \boldsymbol{\mu}_t)^T \boldsymbol{\Sigma}_t^{-1}(\mathbf{x}_t - \boldsymbol{\mu}_t)] d\mathbf{x}_t \\
&+ \int_{CA} \text{sign}(d\theta_r)\ell \exp[-\frac{1}{2}(\mathbf{x}_t - \boldsymbol{\mu}_t)^T \boldsymbol{\Sigma}_t^{-1}(\mathbf{x}_t - \boldsymbol{\mu}_t)] d\mathbf{x}_t.
\end{aligned} \tag{4.115}$$

The second term of (4.108) is essentially the robot motion model, i.e.,

$$\frac{\partial \mathbf{q}_r(k+1)}{\partial \mathbf{u}_r(k)} = \mathbf{B}_r(k)\delta t. \tag{4.116}$$

Thus, we have computed the Jacobian (4.113)-(4.116). The optimized-visibility method is summarized in Algorithm A.1, where η is the learning rate and ϵ is a predefined threshold [36]. Notice that Algorithm A.1 handles the situation when the target is out of the FOV for multiple steps. It is worth pointing out that, when the sensor FOV is far away from the estimated position of the target, the gradient

obtained from (4.108) is usually relatively small. A potential field method based on the distance between the sensor and the target could be used to provide extra force to drive the sensor towards the target.

5

Simulations and Results

This chapter summarizes results generated using the approaches introduced in Chapter 4 to solve the problems in Chapter 2. The information potential method for integrated control and navigation was tested in the first problem, where the sensor is mounted on a mobile platform and is deployed in a workspace populated with position-fixed targets and obstacles. Then, the optimized coverage planning approach was demonstrated through the second problem, which is referred to as the camera intruder problem. In this problem, a position-fixed camera is deployed to monitor multiple moving targets to learn unknown target kinematics, with a FOV that can only cover a portion of the entire workspace at any given time. After that, the optimized visibility planning approach was tested in the third problem, in which one mobile robotic sensor with a bounded FOV is used to track a moving target, where the GPS signal is unavailable.

5.1 Information Potential Approach for Robotic Sensor Classifying Targets

The information potential (IP) method is demonstrated through a simulation environment developed using MATLAB®. All sensors are characterized by the geometric objects in Fig. 2.3, and their motion is simulated by integrating the unicycle dynamics (2.4) with control inputs provided by the switched controller (4.22)-(4.27). The closed-loop dynamics are integrated using a fourth-order Runge-Kutta integration method [56] over a time interval $[t_0,\ t_f]$ with $t_0 = 0$ (s) and $t_f = 20$ (s). Additionally, the following bounds on the control and state are imposed to make the robot kinematics more realistic: $|a| < 5$ (m/s^2), $|v| < 2$(m/s), and $|w| < \pi/10$ (rad/s^2).

The probabilistic model of sensor measurements (2.2) is a Bayesian Network (BN) model of a ground-penetrating radar (GPR) taken from [32]. The non-observable target classification variable $\boldsymbol{\theta}_i$ has two mutually-exclusive values $\boldsymbol{\Theta} = \{\boldsymbol{\zeta}_1,\ \boldsymbol{\zeta}_2\}$. The prior target PDF, $p(\boldsymbol{\theta}_i)$, is given by prior measurements obtained by a simulated airborne Agema Thermovision 900 infrared (IR) sensor [32, 112]. The prior $p(\mathbf{e}_i)$ of the environment condition is assumed to be uniform, and $\mathbf{e}_i$ is assumed to be known for all $i = 1, \ldots, M$. When the sensor FOV, $\mathcal{S}$, intersects a target, $\mathcal{T}_i$, the noisy measurement value, $\mathbf{z}_i$, is obtained, and $\mathbf{x}_i$ is inferred from $\boldsymbol{\theta}_i$ using Bayes' rule in (2.3). Let N_{IR} denote the number of targets that are correctly classified by the IR sensor prior to deploying the robotic GPR sensor, and let N_{GPR} denote the number of targets that are correctly classified after the GPR measurements have been obtained. Then,

$$N_c = N_{GPR} - N_{IR} \tag{5.1}$$

represents the *classification performance* of the robotic sensor. The overall robotic sensor efficiency, defined as the correct classification rate per unit distance, can then

be computed as follows:

$$\eta = \frac{N_c}{D} \times 100, \tag{5.2}$$

where

$$D = \int_{t_0}^{t_f} [\dot{\mathbf{q}}(t)^T \ \dot{\mathbf{q}}(t)]^{1/2} dt \tag{5.3}$$

is the distance traveled by a sensor. The performance metrics N_c, D, and η are used to evaluate the effectiveness of the information potential method.

The simulation results first show that the method is capable of controlling the robotic sensor in narrow passages without oscillations. Then, the closed-loop stability and properties of the IP method are verified in simulation with multiple targets, multiple obstacles, and bounds on the control. Finally, the IP method is shown to outperform both the RRT method discussed above [66] and a classical potential field (PF) method.

IP Multi-Sensor Path Planning and Control in Narrow Passages

A workspace composed of a narrow passage and two targets $\mathcal{T}_1$ and $\mathcal{T}_2$ (Fig. 5.1) is used to illustrate that collisions with other robotic sensors can be avoided in a narrow passage. In this example, the two targets are manually assigned to two robotic sensors for the purpose of forcing them to approach each other inside the narrow passage. The results show that IP allows the two robots to measure their assigned target while avoiding a collision with one another when the two sensors are in the same narrow passage. Therefore, the repulsive force defined in (4.15) is effective at preventing collisions between moving obstacles online and the IP approach is capable of controlling the robotic sensor in narrow passages.

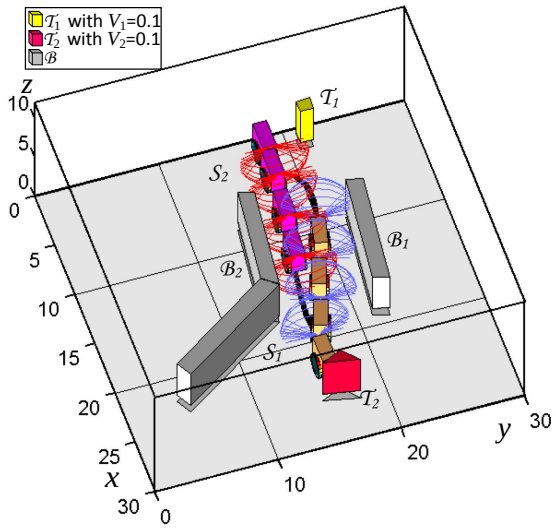


FIGURE 5.1: Simulation results for two robotic sensors in a workspace with two targets, two obstacles, and one narrow passage.

Stability of Information Potential-Switched Control Law

In Section 4.4.4, the IP-switched control law (4.22)-(4.27) was proven to be closed-loop stable, under proper simplifying assumptions. Extensive numerical simulations showed that the IP-switched controller remains stable even when these assumptions are violated by imposing bounds on the linear acceleration a, linear velocity v, and angular velocity w. The time histories of the state and control inputs are plotted in Fig. 5.2, where t_ϵ denotes the time at which $||\boldsymbol{\xi}_i - \mathbf{x}|| = \epsilon$, i.e., when the controller is switched. These results confirm that, with the IP-switched controller, ρ_i, defined in (4.21), goes to zero at t_ϵ and subsequently, for $t > t_\epsilon$, θ converges to β_i and thus a measurement is obtained. As can be seen, $||\boldsymbol{\xi}_i - \mathbf{x}||$ decreases at all times, so the sensor remains inside the cylinder illustrated in Fig. 4.3 after t_ϵ.

Information Roadmap Method for Escaping Local Minima

The information roadmap method was demonstrated through the example in Fig. 5.3, which contained one concave obstacle and two targets: $\mathcal{T}_1$ with information value $V_1 = 0.2$ and $\mathcal{T}_2$ with information $V_2 = 0.1$. The two targets are forced to

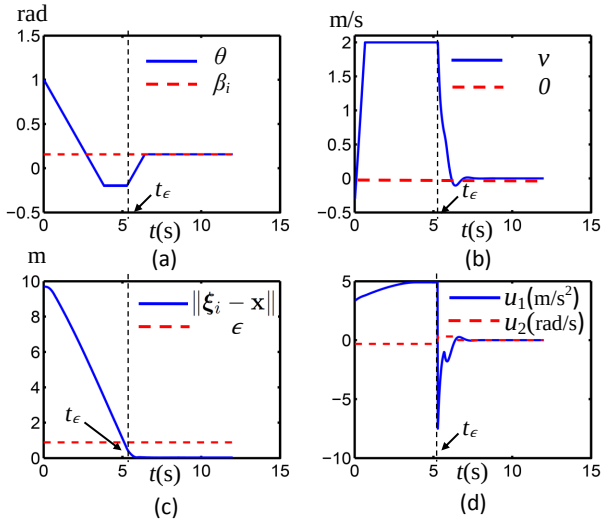


FIGURE 5.2: Time histories of sensor orientation (a), linear velocity (b), distance from C-target (c), and control inputs (d)

be symmetrically positioned above and below the obstacle and to have the same geometry. The concave shape of the obstacle is chosen to create a local minimum in the potential function, encouraging the robot to become trapped at $\tilde{\mathbf{q}}_i$. As shown in Fig. 5.3, configurations in $\mathcal{CT}_1$ have a higher probability of being sampled than configurations in $\mathcal{CT}_2$. From the set M of the sampled configurations, a roadmap G containing a collision-free path from $\tilde{\mathbf{q}}_i$ to $\mathcal{CT}_1$ is constructed (Fig. 5.3) such that the robot can successfully escape $\tilde{\mathbf{q}}_i$ and obtain measurements from the most valuable target $\mathcal{T}_1$.

Performance Comparison

In this subsection, IP is compared to an information-driven RRT method [66] and to a classical potential field (PF) method. Average values of the classification performance (5.1), distance traveled (5.3), and sensor efficiency (5.2) are obtained by computing several paths and controllers for three sensors that are simultaneously deployed in five workspaces. For every simulation, the positions and geometries of the targets and obstacles are generated randomly, as are the initial configurations

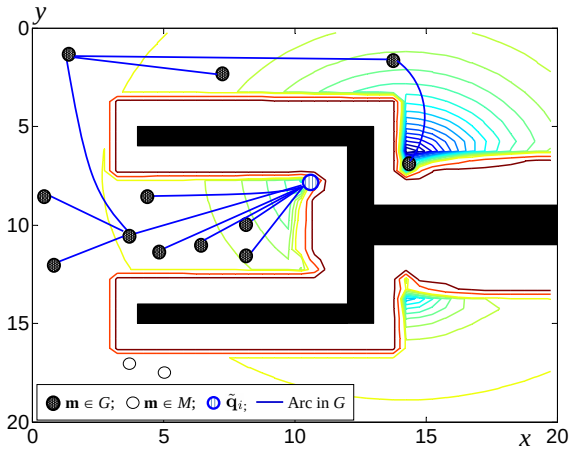


FIGURE 5.3: Potential field contour and information roadmap generated to escape local minima.

of the robotic sensors and the information value of the targets. In simulations, the workspaces are obtained by first considering a medium obstacle density, coupled with a low ($M = 15$), medium ($M = 27$), or high ($M = 40$) target density. Then, a medium target density is considered, coupled with a low ($N = 10$), medium ($N = 17$), or high ($N = 24$) obstacle density.

The average sensor efficiency obtained by IP is summarized in Table 5.1. As can be expected, for the same target density, the sensor performance increases when the obstacle density decreases because the sensor can travel a shorter distance to reach the same targets. Table 5.2 compares the performance of IP with the performance of classical PF and RRT, showing that sensors controlled by IP outperform sensors controlled by PF and RRT not only in average sensor efficiency ($\bar{\eta}$) but also in the number of targets correctly classified (N_c). Also, as can be expected, the distance traveled (D) by IP-controlled sensors may be higher than that by sensors controlled by PF, as PF-controlled sensors are often trapped in local minima, as shown in Fig. 5.4. Additionally, IP produces lower values of D than RRT because it allows the robotic sensors to avoid unnecessary routes and produces smoother paths. However, when the obstacle density is high, D may be higher for IP because the sensors move

to obtain measurements from approximately 75% of their assigned targets, while RRT typically measures only 50% of the targets.

Table 5.1: Average efficiency of IP method

Obstacle density	Target density	$\bar{\eta}$
Medium	Low	0.0223
Medium	Medium	0.0240
Medium	High	0.0482
Low	Medium	0.0439
High	Medium	0.0148

Table 5.2: Average performance comparison for $M = 27$

Obstacle density	Performance	IP	RRT	PF
Low	$\bar{\eta}$	.0439	.0143	.0126
	$\bar{D}$	296.6	420	318
	$\bar{N}_c$	13	6	4
Medium	$\bar{\eta}$	.024	.0139	.0132
	$\bar{D}$	513	397	190
	$\bar{N}_c$	12.3	5.5	2.5
High	$\bar{\eta}$	.0148	.0043	.0055
	$\bar{D}$	575	460	109.5
	$\bar{N}_c$	8.5	2	0.6

Finally, it can be seen from Fig. 5.4 that IP allows the sensor to favor targets with higher information values. For example, in Fig. 5.4, the IP-controlled robotic sensor measures $\mathcal{T}_2$ instead of $\mathcal{T}_1$ and $\mathcal{T}_4$ instead of $\mathcal{T}_3$, despite each pair being a similar distance from the sensor path. On the other hand, as shown in Fig 5.4, the PF-controlled sensor measures $\mathcal{T}_2$ instead of $\mathcal{T}_1$, despite $\mathcal{T}_1$ having a highest information value.

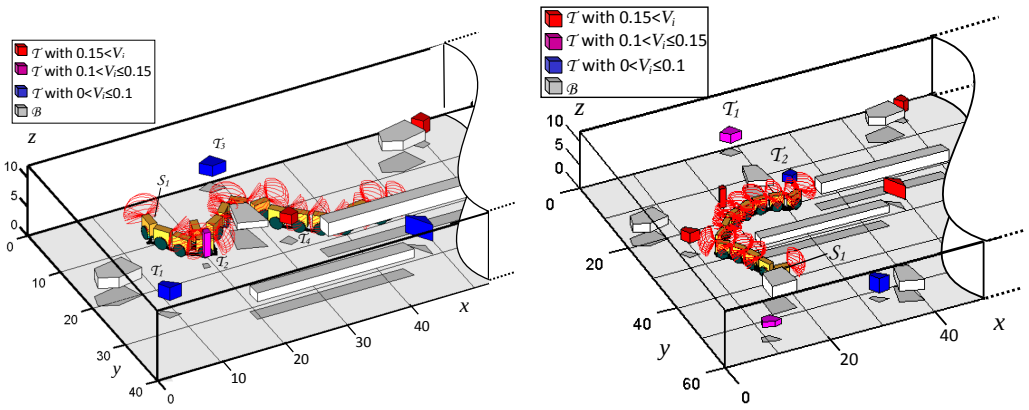


FIGURE 5.4: Details of sensor path obtained by IP method (left) and by classic PF method (right)

5.2 Optimized Coverage Planning for a Camera Monitoring Moving Targets

The performances of four algorithms in solving the camera intruder problem are compared in Chapter 2. The first algorithm is the optimized coverage planning approach that obtains the sensor control by maximizing the DP-GP expected KL divergence at each time step, and its result is labeled as "DP-GP EKL". The second algorithm is a greedy approach maximizing the mutual information of the target position estimation and a future measurement, and its performance is labeled as "MI". The third algorithm is a heuristic that determines the position of the FOV by tracking the mean of the position distribution for the nearest target that is not observed at the last time step, and its result is labeled as "Heuristic". The last algorithm randomly chooses the FOV position and its result is referred to as "Random".

The sensor problem is simulated by designing four velocity fields, which are utilized to specify the target motions. Examples of simulated target trajectories with respect to each velocity field are shown in Fig. 5.5, where the red dots in the trajectory figures are examples of targets' initial positions. In the simulations, the details and number of the velocity fields are hidden from the sensors. The points of interest,

i.e., $\mathbf{X}$, are known to the sensors and are the same for all simulations, indicated by yellow dots in Fig. 5.7. During simulations, at most four targets are allowed to travel simultaneously in the workspace. Every target uniformly chooses one velocity field from the set at random. The DP-GP model is updated once the sensor collects 5 new target trajectories. The Markov Chain Monte Carlo (MCMC) sampling algorithm is adopted to estimate the DP-GP model from measurements, where the number of burn-ins is 200, the number of samples is 40, and the sampling interval is set to be every 5 samples. A snapshot of a simulation is shown in Fig. 5.6.

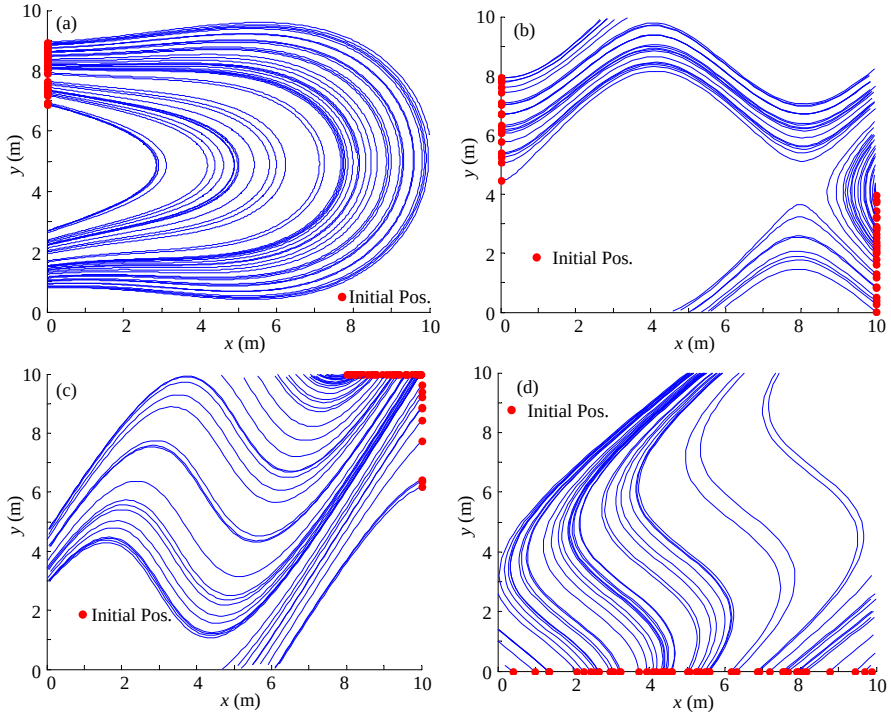


FIGURE 5.5: Examples of target trajectories following the first velocity field; plots of the velocity vectors on a regular grid. (a) $\mathbf{f}_1$, (b) $\mathbf{f}_2$, (c) $\mathbf{f}_3$, and (d) $\mathbf{f}_4$.

The algorithm performance is evaluated using the root mean square (RMS) error, denoted by ε, between the estimated velocity from the DP-GP model and the actual underlying velocity fields. The relative RMS error of velocity, denoted by ξ, is the RMS error, ε, normalized by the velocity $\dot{\mathbf{x}}_j(k)$ at each point. To obtain ε,

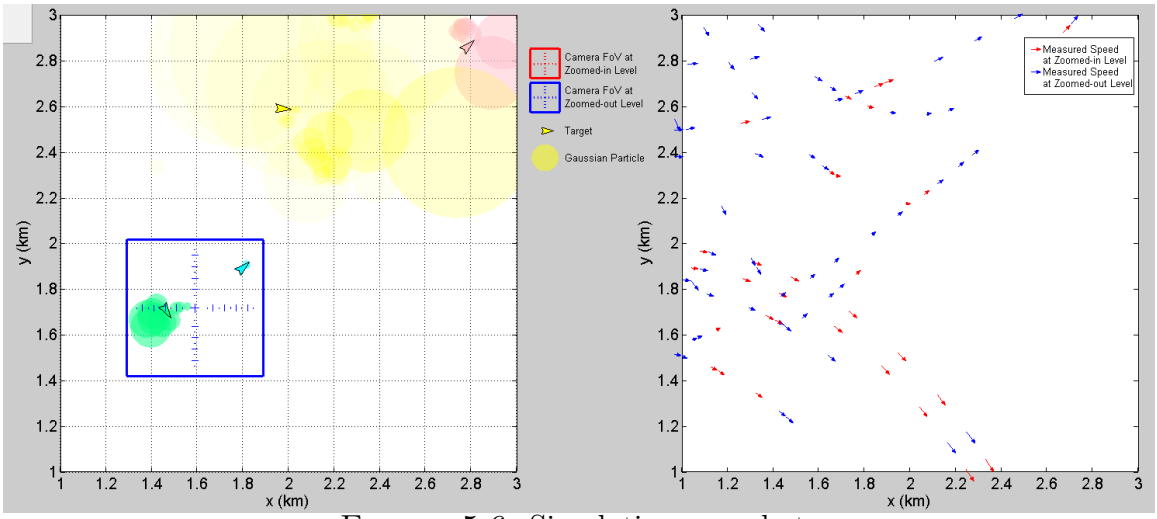


FIGURE 5.6: Simulation snapshot

$N_A = 500$ new test trajectories (distinct from those observed by the camera) $\{\mathcal{T}_j\}$, $j = 1, \ldots, N_A$, are generated according to the motion patterns, where $\mathcal{T}_j = \{\mathbf{x}_i(k), \dot{\mathbf{x}}_j(k)\}$, $k = 1, \ldots, N_{T_j}$, represents the jth new trajectory and N_{T_j} is the length of the jth trajectory. These trajectories are compared with the evolving DP-GP model. By utilizing $\boldsymbol{\mu}_{ji}[\mathbf{x}_j(k)]$ to denote the mean speed at $\mathbf{x}_j(k)$ by the ith Gaussian process component in the DP-GP model, ξ can be expressed as follows:

$$\varepsilon = \frac{1}{N_A} \sum_{j=1}^{N_A} \sum_{i=1}^{M} w_{ji} \sqrt{\frac{1}{N_{T_j}} \sum_{k=1}^{N_{T_j}} \|\dot{\mathbf{x}}_j(k) - \boldsymbol{\mu}_{ji}(\mathbf{x}_j(k))\|_2^2} \tag{5.4}$$

where M is the estimated number of Gaussian process components in the DP-GP model and w_{ji} is updated according to (3.34). The performance is evaluated once the DP-GP model is updated in order to determine the algorithm performance as a function of time. Multiple simulations are conducted in order to obtain statistics of the results.

Three scenarios with different the prior information about the velocity fields are used to examine four strategies. The first scenario is referred to as "more informative prior" (MIP), where a large number (15) of sampled trajectories from the first velocity field and a small number (3 or 4) of sampled trajectories from the remaining velocity

fields are utilized to train the prior DP-GP model. The second scenario is referred to as "intermediate informative prior" (IIP), where a few $(2-4)$ trajectories from each velocity field are used to train the prior DP-GP model. The third scenario is referred to as "less informative prior" (LIP), where no sampled trajectories from the first velocity are utilized to obtain the prior knowledge. Each scenario is tested 50 times with each algorithm.

More Informative Prior Scenario

In the first scenario, the trained DP-GP model provides an estimation of the first velocity field with low uncertainty and an estimation of the remaining velocity fields with high uncertainty. To further illustrate the prior DP-GP model, the prior trajectories and the DP-GP expected KL divergence for each possible position of the future measurement in the entire workspace at $k = 1$ is plotted in Fig. 5.7. The absolute RMS error of the velocity obtained by the four algorithms versus time is shown in Fig. 5.8. As can be seen, the "DP-GP EKL" algorithm outperforms the other algorithms as the error decreases the fastest and reaches the lowest value at the end of the simulation. In addition, the smaller error bar by the "DP-GP EKL" algorithm indicates that its performance is more stable compared to the other methods. The "DP-GP EKL" algorithm is able to follow a target with a motion pattern with higher uncertainty in the current DP-GP model, which explains the faster reduction in error. Figure 5.9 shows that "DP-GP EKL" is able to obtain fewer observations of the targets following the first type of velocity field, as these observations are less informative to the DP-GP model than other observations.

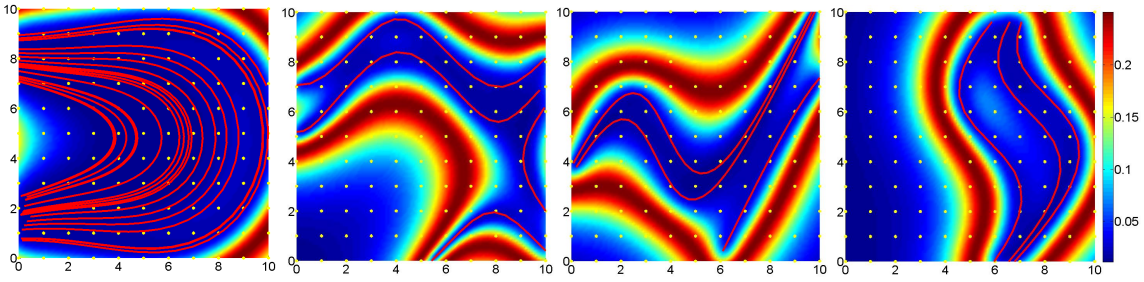


FIGURE 5.7: DP-GP expected KL divergence against each possible position of the future measurement in the workspace at initial time. Red curves: the training trajectories for obtaining MIP; Yellow dots: points of interest.

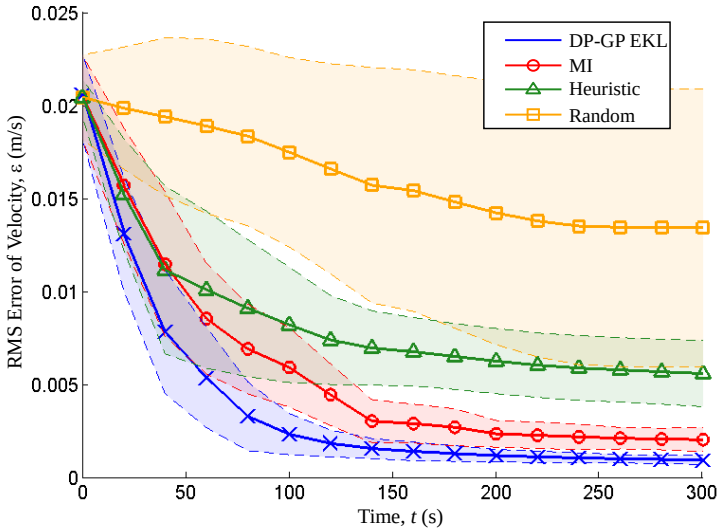


FIGURE 5.8: The mean and variance of the RMS error of thevelocity, ε, obtained by "DP-GP EKL" (blue, cross line), by "MI" (red, circle line), by "Heuristic" (green, triangle line), and by "Random" (yellow, square line), given MIP.

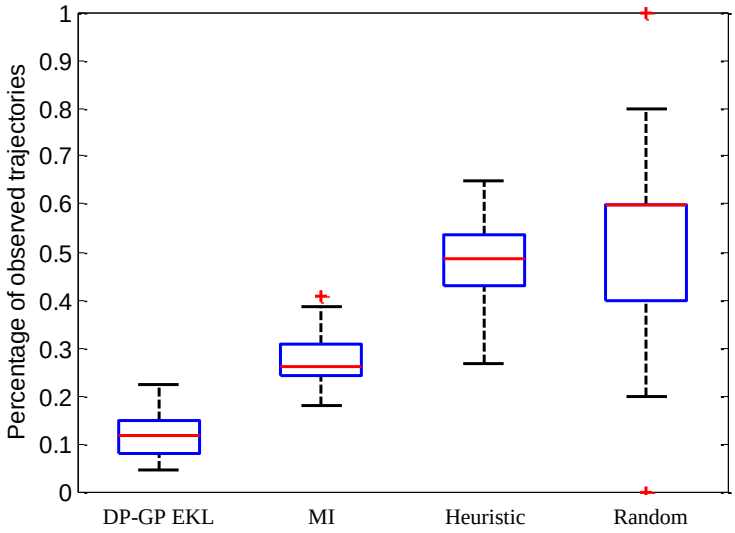


FIGURE 5.9: The percentage of trajectories belonging to the first velocity type observed by the sensor during the simulation given MIP.

Intermediate and Less Informative Prior Scenarios

The second scenario is the "intermediate informative prior" (IIP) simulation. Thus, the trained DP-GP model has an estimation of all of the velocity fields with high uncertainty, while in the "less informative prior" (LIP) scenario, no sampled trajectory from the first velocity is utilized to obtain the prior DP-GP model. As a result, the trained DP-GP model has no knowledge of the first velocity and has only an estimation of the other three velocity fields with high uncertainty.

From Fig. 5.10, we can see that the "DP-GP EKL" algorithm outperforms the other algorithms, as the error decreases the fastest and reaches the lowest value at the end of the simulation. In addition, the smaller error bar by the "DP-GP EKL" algorithm indicates that its performance is more stable compared to other methods. The "DP-GP EKL" algorithm is able to follow a target displaying a motion pattern with higher uncertainty in the current DP-GP model, which explains the faster error decrease rate. Figure 5.11 shows that the "DP-GP EKL" is able to obtain fewer observations of the targets following the first type of velocity field in the "MIP"

scenario, as these observations are less informative to the DP-GP model than other observations. While in the "LIP" scenario, the "DP-GP EKL" 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, leading to a better performance.

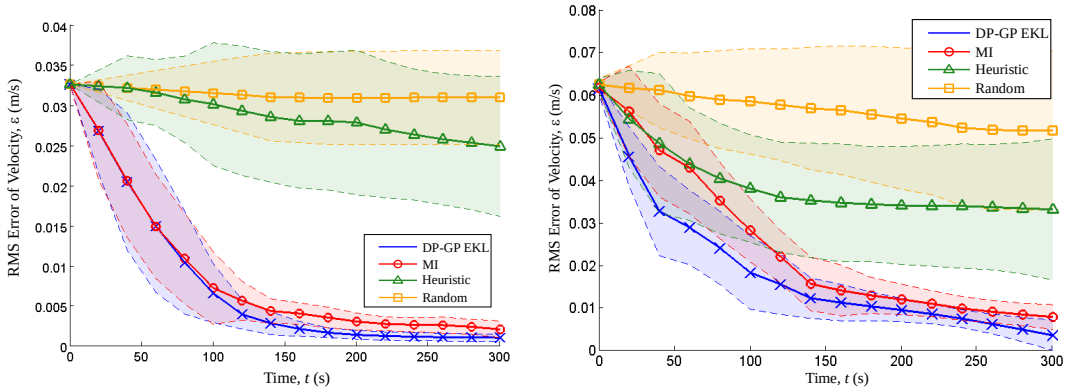


FIGURE 5.10: The mean and variance of RMS error of velocity, ε, obtained by "DP-GP EKL" (blue, cross line), by "MI" (red, circle line), by "Heuristic" (green, triangle line), and by "Random" (yellow, square line), given IIP (left) and LIP (right).

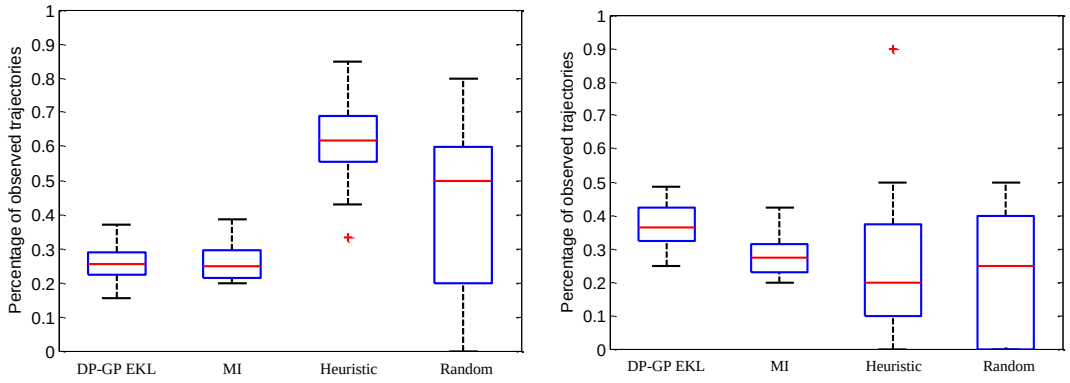


FIGURE 5.11: The percentage of trajectories belonging to the first velocity type observed by the sensor during the simulation Given IIP (left) and LIP (right).

By examining all of the results from the three scenarios shown in Fig. 5.8 and 5.10, we can see that, for all different priors, the "DP-GP EKL" algorithm is more effective at evaluating the expected utility of a future measurement and thus leads to

more informative measurements and a more accurate target model estimation than the "MI", "Heuristic", and "Random" algorithms.

5.3 Optimized Visibility Motion Planning for Robotic Sensor Tracking and Localizing Targets

To validate the effectiveness of the proposed approach, we conduct various simulations under different conditions and compare the performance to that of a state-of-the-art potential field approach. Specifically, the potential approach first calculates a force,

$$\mathbf{f}_p(k) = c_p[\mathbf{x}_p(k) - \boldsymbol{\mu}_t(k)], \tag{5.5}$$

proportional to the distance between the center of the inscribed circle of the FOV, $\mathbf{x}_p(k)$, and the estimated mean of the target position distribution, $\boldsymbol{\mu}_t(k)$, where c_p is a constant. Then, the potential approach projects the force along the robot heading. Let $\theta_p(k)$ denote the angle between the robot heading and the direction from $\mathbf{x}_p(k)$ to $\boldsymbol{\mu}_t(k)$. The control is determined as a linear function of the projections:

$$v_r(k) = a_p \|\mathbf{f}_p(k)\| \cos\theta_p(k) \tag{5.6}$$

$$\omega_r(k) = b_p \|\mathbf{f}_p(k)\| \sin\theta_p(k) \tag{5.7}$$

where a_p and b_p are constants.

In the simulations, the robot and the target are assumed to move in a workspace of $\mathcal{W} = [-50, 50] \times [-50, 50]$ m^2. The sensor's FOV is assumed to have a radius of $\gamma = 2.5$ m and an opening angle of $\alpha = \pi/6$ rad. This choice of parameters results in a relatively small sensor FOV as compared to the workspace, so the target may easily be outside of the FOV. The sampling time, δt, is assumed to be 0.2 sec; thus, the robot makes both proprioceptive measurements, $\mathbf{z}_r(k)$, and exteroceptive measurements, $\mathbf{z}_t(k)$, every 0.2 second. For the proprioceptive measurements, the noise is 2% of the maximum speed of the robot and $\pi/180$ rad/sec for the angular speed measurement.

In all of the tests, it is assumed that the maximum speed that the robot is able to achieve is 3 m/sec and the maximum angular speed for the robot is 0.5 rad/sec. As a result, the proprioceptive noise covariance is $\mathbf{R}_r \approx \text{diag}([36 \quad 3]) \times 10^{-4}$. Note that we did not restrict the robot to travel forward, which means that the robot can travel backward at a maximum speed of 3 m/sec. For the exteroceptive measurements, the noise level is 3% of the maximum detection radius of the FOV for the range measurement and $\pi/36$ rad for the bearing measurement and the noise covariance is $\mathbf{R}_t \approx \text{diag}([81 \quad 76]) \times 10^{-4}$.

Additionally, the moving target follows a constant velocity model:

$$\mathbf{\Phi}_t = \begin{bmatrix} 1 & 0 & \delta t & 0 \\ 0 & 1 & 0 & \delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5.8}$$

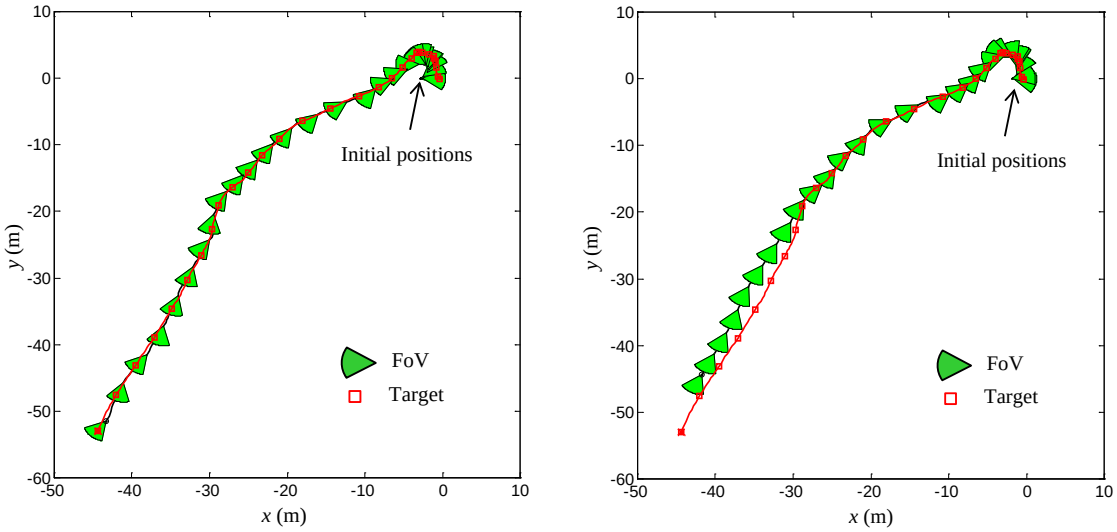
The noise in the target state propagation equation (2.10) causes the target move randomly in the workspace. It is assumed that $\mathbf{G} = \mathbf{I}$ and the noise of the target position is correlated with its speed. The noise matrix is assumed to be

$$\mathbf{Q} = \begin{bmatrix} \delta t^3\sigma^2/3 & 0 & \delta t^2\sigma^2/2 & 0 \\ 0 & \delta t^3\sigma^2/3 & 0 & \delta t^2\sigma^2/2 \\ \delta t^2\sigma^2/2 & 0 & \delta t\sigma^2 & 0 \\ 0 & \delta t^2\sigma^2/2 & 0 & \delta t\sigma^2 \end{bmatrix}, \tag{5.9}$$

where σ is chosen to be 0.5 m/sec, which is large enough to prevent the target from moving in a straight line. The initial state of the target is assumed to be $\mathbf{q}_t(0) = [0 \quad 0 \quad 0 \quad 0]$, which enables the target to move in every direction with the same probability.

Figure 5.12(a) shows the tracking performance of the proposed gradient descent approach for one particular realization with $\eta = 1$ and $\epsilon = 10^{-3}$, from which it is clear that the robot is able to track the target throughout the simulation. The tracking result of the potential method with an identical setup is shown in Fig. 5.12(b). As

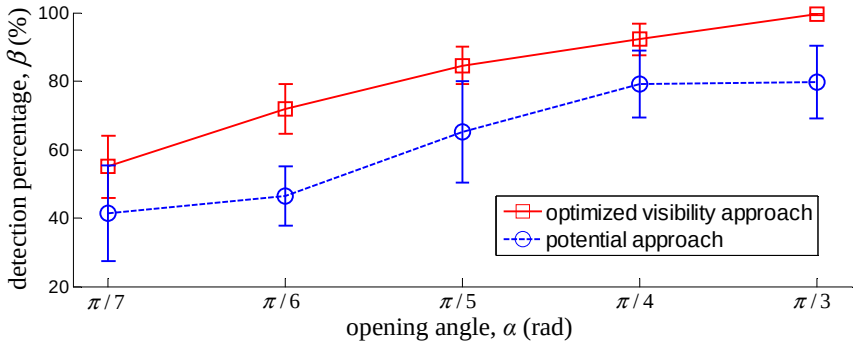
evident, the robot lost the target at time step $k = 170$, while the proposed optimized visibility approach reliably tracks the target (see Fig. 5.12(a)).



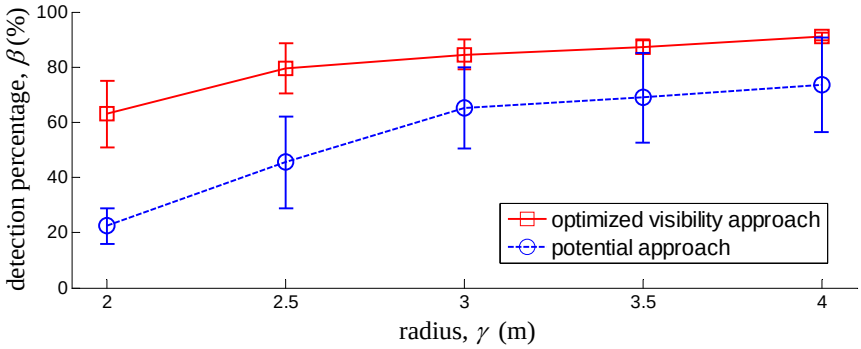
(a) optimized-visibility approach (b) potential approach

FIGURE 5.12: An example of the simulation result where the visibility-optimized approach enables the robot to keep the target in its FOV at all times while the potential field method loses the target around the 200th time step, for a FOV with $\alpha = \pi/6$ rad and $\gamma = 2.5$ m.

To further justify the conclusion drawn from Fig. 5.12, we have performed various simulations with different parameters. In particular, we studied the impact of the FOV opening angle α and the radius γ on the efficiency of the potential and the proposed optimized visibility methods. In order to evaluate the tracking performance, the percentage of target detection, β, is defined as the number of successful target detections divided by the total number of simulation steps. The parameter η is set to one for all of the simulations and ϵ is 10^{-3}. Ten simulations are conducted for each scenario. The mean and one standard variance are summarized in Fig. 5.13, which show that the optimized visibility approach outperforms the potential method with a higher detection percentage.



(a) Opening angle



(b) Edge length

FIGURE 5.13: Percentage of detections obtained by the proposed optimized visibility and the potential approaches for various opening angles and edge lengths

6

Hybrid ADP for Switched Systems

Because active sensors with multiple modes can be modeled as a switched hierarchical system, the sensor path planning problem can be viewed as a hybrid optimal control problem involving both discrete and continuous state and control variables. For example, several authors have shown that a sensor with multiple modalities is a switched hybrid system that can be modeled by a hierarchical control architecture with components of mission planning, trajectory planning, and robot control. This architecture can be modeled by a well-known three-layer hybrid framework with tractable computational complexity. This framework typically involves both discrete state (e.g., the sensor mode) and continuous state (e.g. position and orientation of the robot platform). Additionally, this framework also consists of discrete control (decision on the sensor mode) and continuous control (force, acceleration, or angular speed acting on the robot platform). Such hybrid systems are described by both time-driven and event-driven kinematics. Event-driven kinematics are described by discrete states and controls that are expressed by finite alphabets, while time-driven dynamics (differential or difference equations) are used to represent systems with continuous states and controls in a Euclidean space.

The optimal control of switched hybrid systems seeks to determine an optimal discrete controller that decides the system mode and multiple optimal continuous controllers that regulate the system motion given the system mode, such that a scalar objective function of the hybrid system state and control is minimized over a period of time [12]. The sensor performance can be represented by two Lagrangian functions, one function of the discrete state and control variables, and one function of the continuous state and control variables. Because information value functions are typically nonlinear, this dissertation also presents an adaptive dynamic programming approach for the model-free control of nonlinear switched systems (hybrid ADP), which is capable of learning the optimal continuous and discrete controllers online. The hybrid ADP approach is based on new recursive relationships derived in this dissertation and is proven to converge to the solution of the hybrid optimal control problem. Simulation results show that the hybrid ADP approach is capable of converging to the optimal controllers by minimizing the cost-to-go online based on a fully observable state vector.

6.1 Optimal Control Problem of Switched Systems

The optimal control of switched hybrid systems arises in a wide variety of fields, such as mobile manipulator systems, unmanned robotic sensor planning, and autonomous assemble lines. In these applications, both the discrete and the continuous control are crucial to system performance. The switched system considered in this dissertation has E discrete modes, and its mode at time k is denoted by $\xi(k) \in \mathcal{E}$, where $\mathcal{E} = \{1, \ldots, E\}$, and it is known *a priori*. The discrete control at time k is denoted by $\nu(k) \in \mathcal{E}$. The system continuous state is denoted by $\mathbf{x}(k) \in \mathcal{W} \subset \mathbb{R}^n$, while the continuous control for the system under the discrete control $\nu(k)$ is denoted by $\mathbf{u}_\nu(k) \in \mathcal{U}_\nu \subset \mathbb{R}^{m_\nu}$. Let $\mathbf{c}_\nu[\mathbf{x}(k), k]$ and $a[\mathbf{x}(k), \xi(k), k]$ denote the continuous and discrete controllers, respectively. In the remainder of this dissertation, the continuous

controller $\mathbf{c}_\nu$ and the discrete controller a are referred to as a policy, which is defined by a tuple defined as $\pi = \{a, \mathbf{c}_1, \cdots, \mathbf{c}_E\}$, and the controller approximations are referred to as actor networks.

The system starts at initial state $\mathbf{x}_0$ and at the initial system mode ξ_0, and the (fixed) final time index N is assumed known *a priori*. The objective function of the optimal control problem is given by

$$J \triangleq \phi[\mathbf{x}(N)] + \sum_{j=0}^{N-1} \mathscr{L}_{\nu(j)}[\mathbf{x}(j), \mathbf{u}_{\nu(j)}(j)], \tag{6.1}$$

where $\mathscr{L}_\nu : \mathbb{R}^n \times \mathcal{U}_\nu \to \mathbb{R}$ is the Lagrangian of the system. The objective function is to be minimized with respect to the continuous control $\mathbf{u}_\nu$ and the discrete control ν, subject to the system kinematics, given by

$$\mathbf{x}(k+1) = \mathbf{f}_{\nu(k)}[\mathbf{x}(k), \mathbf{u}_\nu(k)], \quad \xi(k+1) = \nu(k) \in \mathcal{E}. \tag{6.2}$$

Here, $\mathbf{f}_\nu$ is the nonlinear kinematic equation of the switched system under mode ν, $\mathbf{u}_\nu(k) = \mathbf{c}_\nu[\mathbf{x}(k), k]$, and $\nu(k) = a[\mathbf{x}(k), \xi(k), k]$. Additionally, it is assumed that (i) the switching between modes can occur at any time, and it is fully controlled by the discrete control $\nu(k)$, where the cost of switching is zero and switching only affects the system mode; (ii) the system state $\mathbf{x}$ is fully observable and error-free.

6.2 Hybrid ADP Approach

This section presents the proposed Hybrid ADP approach derived from the Bellman Equations. At any time k, the value function for the switched hybrid system is defined as

$$V[\mathbf{x}(k), \xi(k), k] \triangleq \phi[\mathbf{x}(N)] + \sum_{k=0}^{N-1} \mathscr{L}_{\nu(k)}[\mathbf{x}(k), \mathbf{u}_{\nu(k)}(k)]. \tag{6.3}$$

The value function is also referred to as the "cost-to-go", as it sums the instantaneous cost (the Lagrangian) from the current time k to the final time N. Note that at $k = 0$,

given the initial conditions $\mathbf{x}_0$ and ξ_0, the value function $V[\mathbf{x}_0, \xi_0, 0]$ is equal to the cost function J in (6.1). Let the optimal switching mode sequence be denoted by $\{\xi_0, \ldots, \xi^*_{N-1}\}$ and the optimal continuous state be denoted by $\mathbf{x}^*(k)$. Then, the optimal value function at any k has the following recursive form:

$$V^*[\mathbf{x}^*(k), \xi^*(k), k] = V^*[\mathbf{x}^*(k+1), \xi^*(k+1), k+1] + \mathscr{L}_{\nu^*(k)}[\mathbf{x}^*(k), \mathbf{u}^*_{\nu(k)}(k)], \tag{6.4}$$

where $\xi^*(k+1) = \nu^*(k)$. This recursive value function is called the Bellman equation [42].

The optimal continuous controller can be obtained by setting the derivative of the value function (6.4) with respect to (w.r.t.) $\mathbf{u}^*_{\nu^*}(k)$ equal to zero:

$$\frac{\partial V^*[\mathbf{x}^*(k+1), \xi^*(k+1), k+1]}{\partial \mathbf{x}^*(k+1)} \frac{\partial \mathbf{x}^*(k+1)}{\partial \mathbf{u}^*_{\nu^*}(k)} + \frac{\partial \mathscr{L}_{\nu^*(k)}[\mathbf{x}^*(k), \mathbf{u}^*_{\nu^*}(k)]}{\partial \mathbf{u}^*_{\nu^*}(k)} = 0. \tag{6.5}$$

Here, the Hessian of the value function at $\mathbf{u}^*_{\nu^*}(k)$ must be positive definite.

Note that the gradient of the value function w.r.t. the continuous state is required to evaluate the optimality condition (6.5) in order to obtain the optimal $\mathbf{u}^*_\nu(k)$. Let

$$\partial_{\mathbf{x}^*(k+1)} V^*[\mathbf{x}^*(k+1), \xi^*(k+1), k+1] \triangleq \boldsymbol{\lambda}^*[\mathbf{x}^*(k+1), \xi^*(k+1), k+1], \tag{6.6}$$

where $\boldsymbol{\lambda}^*$ is approximated by a neural network, referred to as a critic network. Note that $\boldsymbol{\lambda}^*$ is also known as the costate or adjoint vector in the Hamilton-Jacobi-Bellman (HJB) equation. The critic network is adapted by the critic recurrence relationship, which is derived by taking the derivative of both sides of (6.4) w.r.t. the continuous state, as follows:

$$\begin{aligned}
\boldsymbol{\lambda}^*[\mathbf{x}^*(k), \xi^*(k), k] &= \frac{\partial V^*[\mathbf{x}^*(k), \xi^*(k), k]}{\partial \mathbf{x}^*(k)} \\
&= \frac{\partial \mathscr{L}_{\nu^*}[\mathbf{x}^*(k), \mathbf{u}^*_{\nu^*}(k)]}{\partial \mathbf{x}^*(k)} + \frac{\partial V^*[\mathbf{x}^*(k+1), \xi^*(k+1), k+1]}{\partial \mathbf{x}^*(k)} \\
&= \frac{\partial \mathscr{L}_{\nu^*}[\mathbf{x}^*(k), \mathbf{u}^*_{\nu^*}(k)]}{\partial \mathbf{x}^*(k)} + \boldsymbol{\lambda}^*[\mathbf{x}^*(k+1), \xi^*(k+1), k+1] \frac{\partial \mathbf{x}^*(k+1)}{\partial \mathbf{u}^*_{\nu^*}(k)} \frac{\partial \mathbf{c}^*_{\nu^*}[\mathbf{x}(k), k]}{\partial \mathbf{x}^*(k)}
\end{aligned}$$

$$+ \boldsymbol{\lambda}^*[\mathbf{x}^*(k+1), \xi^*(k+1), k+1] \frac{\partial \mathbf{x}^*(k+1)}{\partial \mathbf{x}^*(k)} + \frac{\partial \mathscr{L}_{\nu*}[\mathbf{x}^*(k), \mathbf{u}^*_{\nu*}(k)]}{\partial \mathbf{u}^*_{\nu*}(k)} \frac{\partial \mathbf{c}^*_{\nu*}[\mathbf{x}(k), k]}{\partial \mathbf{x}^*(k)}. \tag{6.7}$$

where $\mathbf{u}_{\nu*}$ is a function of $\mathbf{x}^*(k)$. The boundary condition for $\boldsymbol{\lambda}^*$ at the end time N is given by

$$\boldsymbol{\lambda}^*[\mathbf{x}^*(N), \xi^*(N), N] = \partial_{\mathbf{x}} \phi[\mathbf{x}(N)], \tag{6.8}$$

where ϕ is the terminal cost function in (6.1).

The objective function (6.1) can be written as

$$\begin{aligned} J = \sum_{j=0}^{k} \mathscr{L}_{\nu^*(j)}[\mathbf{x}^*(j), \mathbf{u}^*_{\nu^*(j)}(j)] + V[\mathbf{x}^*(k-1), \xi^*(k-1), k-1] \\ - V[\mathbf{x}^*(k), \xi^*(k), k] + V[\mathbf{x}^*(k), \xi^*(k+1), k+1], \end{aligned} \tag{6.9}$$

because $V[\mathbf{x}^*(k-1), \xi^*(k-1), k-1] - V[\mathbf{x}^*(k), \xi^*(k), k]$ is equal to $\mathscr{L}_{\nu^*(k-1)}[\mathbf{x}^*(k-1), \mathbf{u}^*_{\nu^*(k-1)}(k-1)]$. Because $V[\mathbf{x}^*(k), \xi^*(k), k]$ and $V[\mathbf{x}^*(k), \xi^*(k+1), k+1]$ are differentiable w.r.t. $\mathbf{x}^*(k)$, J is also differentiable w.r.t. $\mathbf{x}^*(k)$. Therefore, at the optimal state $\mathbf{x}^*(k)$, $\frac{\partial J}{\partial \mathbf{x}^*(k)}$ is zero (otherwise, $\mathbf{x}^*(k)$ is not optimal). Then, the transversality condition of $\boldsymbol{\lambda}^*$,

$$\boldsymbol{\lambda}^*[\mathbf{x}^*(k), \xi^*(k), k] = \boldsymbol{\lambda}^*[\mathbf{x}^*(k), \xi^*(k+1), k], \tag{6.10}$$

can be obtained from taking the derivative of both sides of the above equation with respect to $\mathbf{x}^*(k)$ and setting it to zero.

The optimal discrete control is obtained by minimizing the Hamiltonian, following the discrete minimum principle [78]:

$$\nu^*(k) = \underset{\nu}{\operatorname{argmin}}\, H_\nu[\mathbf{x}^*(k), \mathbf{u}_{\nu*}, \boldsymbol{\lambda}_*, k], \tag{6.11}$$

where the Hamiltonian is given by

$$H_\nu[\mathbf{x}^*(k), \mathbf{u}^*_\nu, \boldsymbol{\lambda}, k] = \mathscr{L}_\nu[\mathbf{x}^*(k), \mathbf{u}^*_\nu(k)] + \boldsymbol{\lambda}^*[\mathbf{x}^*(k+1), \nu, k+1] \mathbf{f}_\nu[\mathbf{x}^*(k), \mathbf{u}^*_\nu(k)]. \tag{6.12}$$

The optimality conditions (6.5), (6.7), (6.10), and (6.11) are used to adapt $\boldsymbol{\lambda}$, a, and $\{\mathbf{c}_1, \cdots, \mathbf{c}_E\}$ online. Each of these functions is approximated by a neural network (NN). Then, $\mathrm{NN}_{\lambda}^{\xi}(\mathbf{w}_{\boldsymbol{\lambda}}^{\xi}) \approx \boldsymbol{\lambda}^*[\mathbf{x}, \xi, k]$ is called the critic network, $\mathrm{NN}_{c}^{\nu}(\mathbf{w}_{c}^{\nu}) \approx \mathbf{c}_{\nu}^*[\mathbf{x}(k), k]$ is called the actor network for each mode, and $\mathrm{NN}_a(\mathbf{w}_a) \approx a[\mathbf{x}(k), \xi, k]$ is also called an actor network. The adjustable parameters of NN_{c}^{ν}, $\mathrm{NN}_{\lambda}^{\xi}$, and NN_a are denoted by $\mathbf{w}_{\boldsymbol{\lambda}}^{\xi}$, $\mathbf{w}_{c}^{\nu}$, and $\mathbf{w}_a$, respectively. As schematized in Fig. 6.1, the hybrid ADP approach cycles between critic network adaptation and actor network adaptation, as summarized in Appendices B.1-B.2. Each cycle contains T iterations of the critic network adaptation and M iterations of the actor network adaptation, where iterations are indexed by k.

At each cycle of the hybrid ADP algorithm, indexed by l, a new improved policy $\pi^l = \{a^l, \mathbf{c}_1^l, \cdots, \mathbf{c}_E^l\}$ is obtained by holding the critic parameters fixed and by adapting the actor parameters as follows:

$$\Delta \mathbf{w}_{c}^{\nu} = -\epsilon \left\{ \frac{\partial \mathbf{x}^{\nu}(k+1)}{\partial \mathbf{u}_{\nu}(k)} \boldsymbol{\lambda}[\mathbf{x}^{\nu}(k+1), \xi(k+1), k+1] - \frac{\partial \mathscr{L}_{\nu}[\mathbf{x}(k), \mathbf{u}_{\nu}(k)]}{\partial \mathbf{u}_{\nu}(k)} \right\} \frac{\mathbf{c}_{\nu}[\mathbf{x}(k), k]}{\partial \mathbf{w}_{c}^{\nu}}, \tag{6.13}$$

where ϵ is a positive, user-defined learning rate. The actor parameters of the discrete controller at time k are trained through supervised learning with training examples. Each example is a pair consisting of an input vector ($[\mathbf{x}(k), \xi(k), k]$) and a desired discrete control value ($\nu(k)$) obtained from (6.11), as shown in Appendix B.2.

Then, holding the actor parameters fixed, a new improved critic network $\boldsymbol{\lambda}^l$ is obtained by adapting its parameters. From the critic recurrence relationship (6.7), at each time step k, the critic parameters can be updated according to the learning rule,

$$\Delta \mathbf{w}_{\lambda}^{\xi} = -\eta \left\{ \boldsymbol{\lambda}[\mathbf{x}(k), \xi(k), k] - \frac{\partial \mathscr{L}_{\nu}[\mathbf{x}(k), \mathbf{u}_{\nu}(k)]}{\partial \mathbf{u}_{\nu}(k)} \frac{\partial \mathbf{c}_{\nu}(k)}{\partial \mathbf{x}(k)} - \frac{\partial \mathscr{L}_{\xi}[\mathbf{x}(k), \mathbf{u}_{\nu}(k)]}{\partial \mathbf{x}(k)} \right.$$

$$- \boldsymbol{\lambda}[\mathbf{x}(k+1), \xi(k+1), k+1]\Big[\frac{\partial \mathbf{x}(k+1)}{\partial \mathbf{x}(k)} - \frac{\partial \mathbf{x}(k+1)}{\partial \mathbf{u}_\nu(k)}\frac{\partial \mathbf{c}_\nu(k)}{\partial \mathbf{x}(k)}\Big]\Big\}\frac{\boldsymbol{\lambda}[\mathbf{x}(k), \xi(k), k]}{\partial \mathbf{w}_\lambda^\xi}, \tag{6.14}$$

where $\mathbf{u}_\nu(k) = \mathbf{c}_\nu[\mathbf{x}(k), k]$ and η is a positive user-defined learning rate.

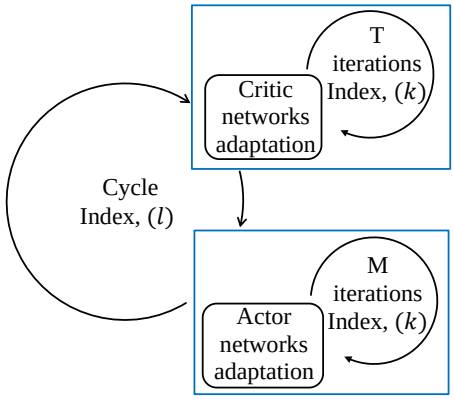


FIGURE 6.1: Critic and actor network adaptation in hybrid ADP.

The number of hidden neurons (N_h) in each critic network depends on the size of training samples (N_s), which is further determined by the shape of $\boldsymbol{\lambda}$ and $\mathbf{c}_\nu$. With the assumption that $\boldsymbol{\lambda}$ and $\mathbf{c}_\nu$ have Lipschitz-continuous gradients with respect to $\mathbf{x}$ with modulus L, the sufficient sample size is $\text{vol}(\mathcal{W})L/(\varepsilon)^n$ [10], where ε is a targeted approximation error, $\text{vol}(\mathcal{W})$ is the volume of $\mathcal{W}$, and n is the number of workspace dimensions. An empirical number of hidden neurons is given by $N_h = (n + \sqrt{N_s})/N_L$[48], where N_L is the number of hidden layers. Because two neural networks are adopted to approximate the controller and critic for each mode, the total number of neural networks is $2E$. Then the computational complexity of each updating step (6.14) or (6.13) is $O(N_h^2 + nN_h)$, and thus the total complexity of each cycle is $O[E(N_h^L + nN_h)(M + N)]$.

6.3 Hybrid ADP for Optimal Control Problem of Linear Switched Systems

The hybrid ADP algorithm is demonstrated on a hybrid linear-quadratic (LQ) optimal control problem for which an exact solution can be obtained by solving a switched differential Riccati equation (SDRE) numerically, as was first shown in [78]. The hybrid system considered in this study consists of a power system with two modes, one gasoline-driven and one electric-driven. Each can be represented by a linear time-invariant (LTI) system with a continuous state vector $\mathbf{x} = [x\ \dot{x}]^T$, where $x \in \mathcal{X} \subset \mathbb{R}$. It is assumed that the state $\mathbf{x}$ is fully observable and that the measurements are error-free. It is also assumed that the system mode can switch to any of the two power systems at any time from a discrete time index set $\{0, 1, \cdots, N\}$, where N is given, and that the two power systems are independent and supplied with sufficient fuel.

The mode of the power system is represented by a discrete, binary state variable $\xi \in \mathcal{E}$, where $\mathcal{E} = \{1, 2\}$; $\xi = 1$ denotes the gasoline-driven model, and $\xi = 2$ denotes the electric-driven mode. The system dynamics can be modeled by a set of different LTI subsystems:

$$\mathbf{x}(k+1) = \begin{cases} \mathbf{A}_1\mathbf{x}(k) + \mathbf{B}_1\mathbf{u}(k), & \text{for } \nu(k) = 1 \\ \mathbf{A}_2\mathbf{x}(k) + \mathbf{B}_2\mathbf{u}(k), & \text{for } \nu(k) = 2 \end{cases}, \tag{6.15}$$

where $\mathbf{u} \in \mathbb{R}^2$ is the continuous control input, and the initial continuous state, $\mathbf{x}(0) = \mathbf{x}_0$, is given. The system matrixes are given as follows:

$$\mathbf{A}_1 = \begin{pmatrix} 1 & 0.05 \\ -0.05 & 0.95 \end{pmatrix}, \mathbf{A}_2 = \begin{pmatrix} 1 & 0.05 \\ -0.05 & 0.975 \end{pmatrix}, \mathbf{B}_1 = [0 \ \ 0.05]^T, \mathbf{B}_2 = [0 \ \ 0.04]^T. \tag{6.16}$$

At any time $k \in \{0, \ldots, N-1\}$ $(N = 100)$, the system mode ξ can be fully controlled at no cost by a switching signal $\nu \in \mathcal{E}$ provided by the discrete controller.

Thus, the objective function to be minimized is represented by a switched integral cost,

$$J = \mathbf{x}^T(N)\mathbf{P}_f\mathbf{x}(N) + \sum_{j=0}^{N-1} \mathbf{x}^T(j)\mathbf{Q}_\nu\mathbf{x}(j) + \mathbf{u}_\nu^T(j)\mathbf{R}_\nu\mathbf{u}_\nu(j), \tag{6.17}$$

where $\mathbf{R}_1 = 400$, $\mathbf{R}_2 = 50$,

$$\mathbf{Q}_1 = \begin{pmatrix} 100 & 0 \\ 0 & 200 \end{pmatrix}, \mathbf{Q}_2 = \begin{pmatrix} 250 & 0 \\ 0 & 200 \end{pmatrix}, \quad \text{and} \quad \mathbf{P}_f = \begin{pmatrix} 1500 & -1500 \\ -1500 & 3000 \end{pmatrix}. \tag{6.18}$$

Then, from [78], the switched differential Riccati equation is given by

$$\mathbf{P}(k-1) = \mathbf{Q}_\nu + \mathbf{A}_\nu^T\Big(\mathbf{P}(k) - \mathbf{P}(k)\mathbf{B}_\nu(\mathbf{R}_\nu + \mathbf{B}_\nu^T\mathbf{P}(k)\mathbf{B}_\nu)^{-1}\mathbf{B}_\nu^T\mathbf{P}(k)\Big)\mathbf{A}_\nu, \tag{6.19}$$

where the discrete controller is obtained by

$$\nu(k) = \underset{\nu}{\operatorname{argmin}}\{H_\nu[\mathbf{P}(k), \mathbf{x}(k), \xi(k), \mathbf{u}(k)]\}. \tag{6.20}$$

With an initial condition $\mathbf{x}(0) = [0.5596\ -0.6387]^T$ and a final condition $\mathbf{x}(N) = [0.01\ 0]^T$, the solution from solving the SDRE numerically is plotted in Fig. 6.2, where the gasoline-driven mode is shown by red dashed lines with square markers and the electric-driven mode is shown by blue dashed lines with point markers. The switching mode and instants can be identified by the change in color, curve style, and "+" along the trajectory.

The hybrid ADP algorithm is applied to the same optimal control problem. The critic network is initialized to satisfy the terminal condition on the costate vector $\boldsymbol{\lambda}(N) = P_f\mathbf{x}(N) = [0\ 0]^T$, while the actor network is trained to satisfy

$$\mathbf{u}_\nu(k) = -\Big(R_\nu + B_\nu^T B_\nu\Big)^{-1}\Big[B_\nu^T(I + A_\nu)\mathbf{x}(k)\Big], \tag{6.21}$$

such that (6.5) holds. Subsequently, the hybrid ADP recurrence relationships presented in Section 6.2 are used to adapt the critic and the actor networks online, while the resultant actor networks are used to control the power system. Unlike the

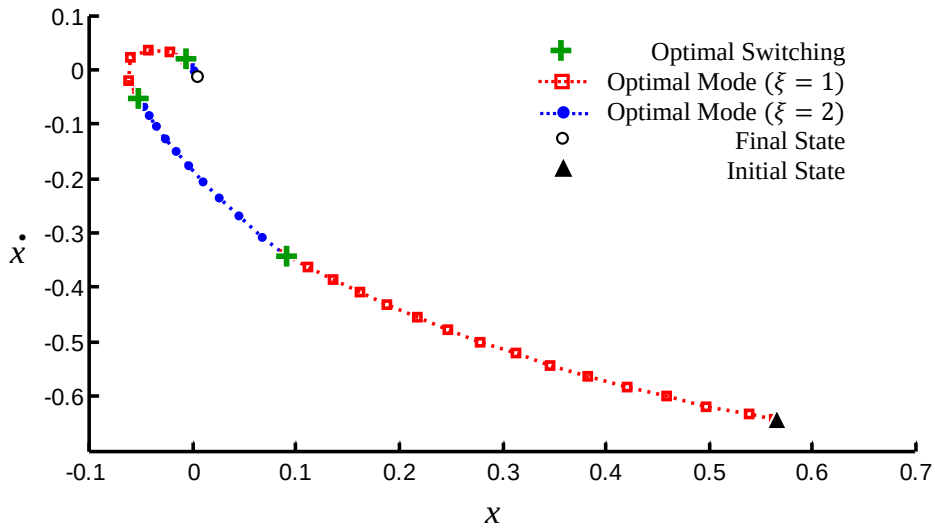


FIGURE 6.2: Optimal state trajectory obtained from SDRE solution.

SDRE approach, hybrid ADP only uses online evaluations of the state and immediate reward that could be obtained by a simulation or real system.

In this example, the critic (actor) neural networks have two hidden layers with 30 (10) hyperbolic tangent (or sigmoidal) functions, and the learning rates η and ϵ are both chosen to be equal to 5×10^{-2}. The learning steps are $T = 400$ and $M = 100$. The value of the cost function (6.17) is evaluated at every cycle and plotted in Fig. 6.3, where it is shown to converge to the optimal cost known from the SDRE solution (dashed line). In this simulation, the learning rates η and ϵ are deliberatively chosen to be greater than $1/(L_1 L_2 + L_3)L_4$ in order to accelerate convergence; therefore, the cost function does not decrease at every cycle of the algorithm.

The optimal trajectory and the state trajectories obtained by the hybrid ADP algorithm are indicated by the dashed and solid lines in Fig. 6.4 for five cycles, respectively. When $l = 5$, the state trajectories converge to the optimal state trajectory obtained from the SDRE solution in Fig. 6.2. For the trajectory obtained by ADP, the switching mode and instants can be identified by the change in color, curve style, and "×" along the trajectory. The gasoline-driven mode is indicated by

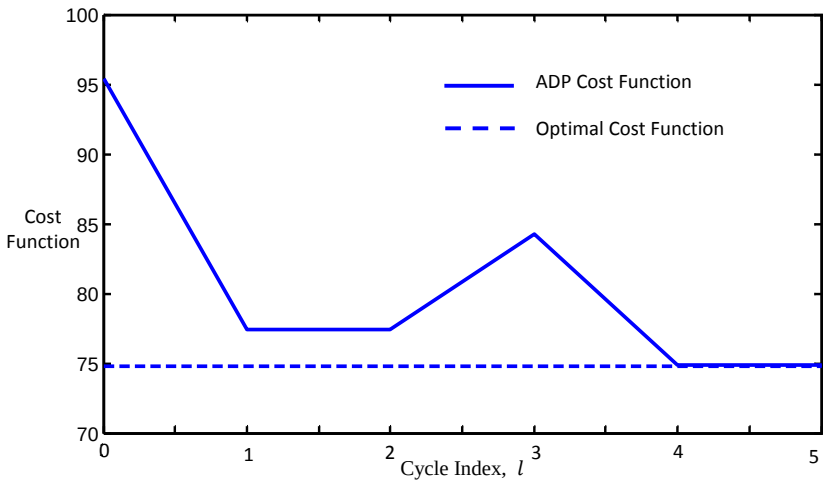


FIGURE 6.3: Hybrid ADP cost function convergence to optimal solution obtained by SDRE.

red solid lines with diamond markers, and the electric-driven mode is indicated by blue solid lines with circle markers.

These results demonstrate that the hybrid ADP algorithm is capable of learning the optimal controller for a switched LQ problem online, using state observations obtained over time from a simulation of the system. Because the algorithm does not rely on the LQ structure of the system dynamics and cost function, hybrid ADP can be similarly applied to nonlinear (and/or time-varying) switched systems, for which SDER solutions are not typically available.

6.4 Convergence Analysis

The hybrid ADP algorithm can be guaranteed to converge to a global or local optimal solution under the following assumption:

Assumption 1. *Assume* $V(\mathbf{x}, \xi, k)$, $\mathbf{x}(k+1)$, $\mathscr{L}_\nu[\mathbf{x}, \mathbf{u}_\nu]$, *and* $NN_c^\nu(\mathbf{w}_c^\nu)$ *each have Lipschitz-continuous gradients with respect to* $\mathbf{x}$, $\mathbf{u}_\nu(k)$, $\mathbf{u}_\nu(k)$, *and* $\mathbf{w}_c^\nu$, *with modulus* L_1, L_2, L_3, *and* L_4, *respectively.*

Four lemmas are presented prior to presenting the convergence proof. The first

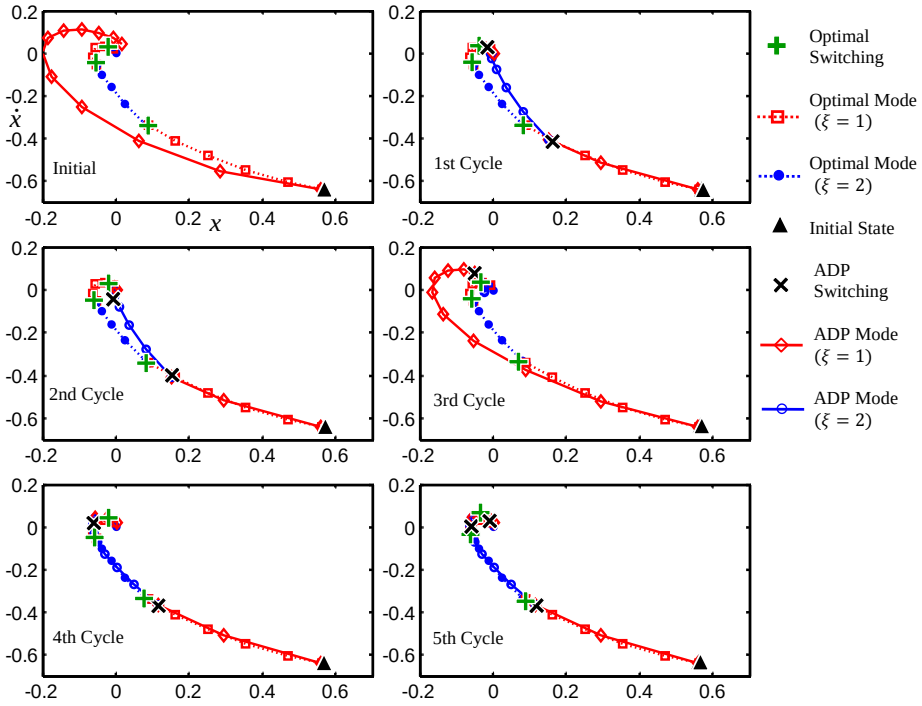


FIGURE 6.4: State trajectory optimization for five cycles of the hybrid ADP algorithm and convergence to optimal solution obtained by SDRE.

two lemmas build connections among the recurrence relationship, the value function, and the updating rules for the weights (6.14). Then, the last two lemmas establish the progression of the policy and value function updates at consecutive iterations. This is schematized in Fig. 6.5, where critic or actor network adaptation is denoted by an arrow, such that, for example, $\boldsymbol{\lambda}^l \rightarrow \pi^{l+1}$ denotes a network π^{l+1} that is adapted by holding $\boldsymbol{\lambda}^l$ fixed.

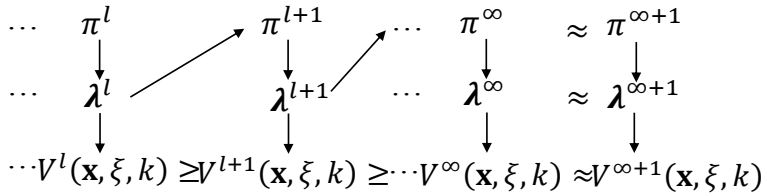


FIGURE 6.5: Iterations between actor network adaptations and critic network adaptations.

Lemma 1. *Let π^l denote a control policy obtained by the update in (6.14) while fixing $\boldsymbol{\lambda}^{l-1}[\mathbf{x},\xi,k]$. Then, for $T \gg 1$, the critic costate vector $\boldsymbol{\lambda}^l[\mathbf{x},\xi,k]$ obtained from the recurrence relation (6.7), while fixing π^l, satisfies*

$$\boldsymbol{\lambda}^l[\mathbf{x}^l(k),\xi^l(k),k] = \frac{\partial \mathscr{L}_{\nu^l}[\mathbf{x}(k),\mathbf{u}^l_\nu(k)]}{\partial \mathbf{x}(k)} + \boldsymbol{\lambda}^l[\mathbf{x}^l(k+1),\xi^l(k+1),k+1]\frac{\partial \mathbf{x}^l(k+1)}{\partial \mathbf{u}^l_\nu(k)}\frac{\partial \mathbf{c}^l_\nu[\mathbf{x}(k),k]}{\partial \mathbf{x}(k)} + \frac{\partial \mathscr{L}_{\nu^l}[\mathbf{x}(k),\mathbf{u}^l_\nu(k)]}{\partial \mathbf{u}^l_\nu(k)}\frac{\partial \mathbf{c}^l_\nu[\mathbf{x}(k),k]}{\partial \mathbf{x}(k)} + \frac{\partial \mathbf{x}^l(k+1)}{\partial \mathbf{x}(k)}\boldsymbol{\lambda}^l[\mathbf{x}^l(k+1),\xi^l(k+1),k+1], \tag{6.22}$$

where $\mathbf{x}^l(k+1)$ and $\xi^l(k+1)$ denote the state of the switched system at the $(k+1)$th time step, obtained by implementing the control policy π^l at time step k.

Proof of Lemma 1. Given $\mathbf{x}(0) = \mathbf{x}_0$, the trajectory $\mathbf{x}^l(k)$, $k = 1,\ldots,N$, obtained by the control policy π^l is fixed. Then, at time step k, the coefficient of $\boldsymbol{\lambda}^l(k+1)$ and the remaining term in (6.7) are all constant matrices or vectors evaluated at $\mathbf{x}^l(k)$ and can be denoted by $\mathbf{A}(k)$ and $\mathbf{b}(k)$, respectively. Hence, the recurrence relationship (6.7) can be expressed as

$$\boldsymbol{\lambda}^l(k) = \mathbf{A}(k)\boldsymbol{\lambda}^l(k+1) + \mathbf{b}(k), \tag{6.23}$$

which is the kth equation in a linear system of equations. The Nth equation in this system of equations is $\boldsymbol{\lambda}^l(N) = \partial_x \phi[\mathbf{x}^l(N)]$. Thus, (6.14) follows the successive over-relaxation (SOR) method with relaxation factor η [81]. The eigenvalue of the iteration matrix for this linear system is $1-\eta$, the absolute value of which is less than unity. Therefore, when $T \gg 1$, the solution to (6.23) can be computed by SOR [95]. Therefore, $\boldsymbol{\lambda}^l[\mathbf{x},\xi,k]$ can be obtained from (6.14), satisfying (6.22). □

Remark 1. *The boundary condition of $\boldsymbol{\lambda}^l[\mathbf{x}(N),\xi(N),N]$ is also satisfied when $\boldsymbol{\lambda}^l[\mathbf{x},\xi,k]$ is computed from (6.14), such that $\boldsymbol{\lambda}^l[\mathbf{x}(N),\xi(N),N] = \partial_x \phi[\mathbf{x}^l(N)]$.*

Lemma 2. *Obtained on the lth cycle of the hybrid ADP algorithm, given π^l, the critic network, $\boldsymbol{\lambda}^l[\mathbf{x},\xi,k]$, and its corresponding value function, $V^l(\mathbf{x},\xi,k)$, have the*

following properties.

$$V^l[\mathbf{x}^l(k), \xi^l(k), k] = \mathscr{L}_{\nu^l}[\mathbf{x}^l(k), \mathbf{u}_\nu^l(k)] + V^l[\mathbf{x}^l(k+1), \xi^l(k+1), k+1],$$

$$V^l[\mathbf{x}^l(N), \xi^l(N), N] = \phi[\mathbf{x}(N)],$$

$$\frac{\partial V^l[\mathbf{x}^l(k), \xi^l(k), k]}{\partial \mathbf{x}^l(k)} = \boldsymbol{\lambda}^l[\mathbf{x}^l(k), \xi^l(k), k], \tag{6.24}$$

for all $\mathbf{x}^l(k)$ *and* $\xi^l(k)$ *and at any time step* k*. The proof is neglected here.*

Lemma 3. *Given* $\boldsymbol{\lambda}^l(\mathbf{x}, \xi, k)$*,* $\mathbf{c}_\nu^{l+1}(\mathbf{x}, k)$ *updated by (6.13) with a learning rate* $\epsilon = 1/(L_1L_2 + L_3)L_4$*, we have*

$$V^l[\mathbf{x}^{l+1}(k+1), \xi(k+1) = \nu(k), k+1] + \mathscr{L}_\nu[\mathbf{x}(k), \mathbf{u}_\nu^{l+1}(k)]$$

$$\leqslant V^l[\mathbf{x}^l(k+1), \xi(k+1) = \nu(k), k+1] + \mathscr{L}_\nu[\mathbf{x}(k), \mathbf{u}_\nu^l(k)], \ \forall \nu(k) \in \mathcal{E}, \tag{6.25}$$

where $\mathbf{u}_\nu^l(k) = \mathbf{c}_\nu^l[\mathbf{x}(k)]$ *and* $\mathbf{u}_\nu^{l+1}(k) = \mathbf{c}_\nu^{l+1}[\mathbf{x}(k)]$*.*

Proof of Lemma 3. When $\mathbf{x}(k)$ and $\nu(k)$ are given, the control input $\mathbf{u}_\nu^l(k)$ is a function of the actor network weights $\mathbf{w}_c^\nu(l)$. Thus, the system state at the next time step, $\mathbf{x}^l(k+1)$, is a function of $\mathbf{w}_c^\nu(l)$ and, for fixed $V^l(\mathbf{x}, \xi, k)$, so is $V^l[\mathbf{x}^l(k+1), \nu(k), k]$. It follows that the expression $\{V^l[\mathbf{x}^l(k+1), \nu(k), k] + \mathscr{L}_\nu[\mathbf{x}(k), \mathbf{u}^l(k)]\} \triangleq G^l[\mathbf{w}_c^\nu(l)]$ is a function of $\mathbf{w}_c^\nu(l)$ and that $G^l[\mathbf{w}_c^\nu(l+1)]$ is only a function of the actor network weights at the next iteration cycle, i.e., $\mathbf{w}_c^\nu(l+1)$.

Based on Assumption (1), the values of G^l evaluated at $\mathbf{w}_c^\nu(l)$ and $\mathbf{w}_c^\nu(l+1)$ obey the inequality

$$|G^l[\mathbf{w}_c^\nu(l)] - G^l[\mathbf{w}_c^\nu(l+1)]|$$

$$= |V[\mathbf{x}^l(k+1), \nu(k), k] + \mathscr{L}_\nu[\mathbf{x}(k), \mathbf{u}^l(k)] - V[\mathbf{x}^{l+1}(k+1), \nu(k), k] - \mathscr{L}_\nu[\mathbf{x}(k), \mathbf{u}^{l+1}(k)]|$$

$$\leqslant |V[\mathbf{x}^l(k+1), \nu, k] - V[\mathbf{x}^{l+1}(k+1), \nu, k]| + |\mathscr{L}_\nu[\mathbf{x}(k), \mathbf{u}^l(k)] - \mathscr{L}_\nu[\mathbf{x}(k), \mathbf{u}^{l+1}(k)]|$$

$$\leqslant L_1|\mathbf{x}^l(k+1) - \mathbf{x}^{l+1}(k+1)| + L_3|\mathbf{u}^l(k) - \mathbf{u}^{l+1}(k)|$$

$$\leqslant (L_1L_2 + L_3)|\mathbf{u}^l(k) - \mathbf{u}^{l+1}(k)|$$

$$\leqslant (L_1L_2 + L_3)L_4|\mathbf{w}_c^\nu(l+1) - \mathbf{w}_c^\nu(l)|. \tag{6.26}$$

Here, $|\cdot|$ denotes the Euclidean norm and G^l is a function of $\mathbf{w}_c^\nu$ with the Lipschitz-continuous gradient modulus $(L_1L_2 + L_3)L_4$. Then, the following inequality holds:

$$\begin{aligned}
&G^l[\mathbf{w}_c^\nu(l+1)] \\
&\leqslant G^l[\mathbf{w}_c^\nu(l)] + < \nabla G^l[\mathbf{w}_c^\nu(l)], \mathbf{w}_c^\nu(l+1) - \mathbf{w}_c^\nu(l) > + \frac{(L_1L_2+L_3)L_4}{2}|\mathbf{w}_c^\nu(l+1) - \mathbf{w}_c^\nu(l)| \\
&= G^l[\mathbf{w}_c^\nu(l)] - \epsilon|\nabla G^l[\mathbf{w}_c^\nu(l)]|^2 + (L_1L_2 + L_3)L_4\epsilon^2/2|\nabla G^l[\mathbf{w}_c^\nu(l)]|^2,
\end{aligned} \tag{6.27}$$

from the properties of functions with Lipschitz-continuous gradients [6] and the controller update (6.13). Then, letting $\epsilon = 1/(L_1L_2 + L_3)L_4$, the following inequality holds:

$$G^l[\mathbf{w}_c^\nu(l+1)] \leqslant G^l[\mathbf{w}_c^\nu(l)]. \tag{6.28}$$

By using $\mathbf{c}_\nu^{l+1}(\mathbf{x}, k)$, we have

$$V^l[\mathbf{x}^{l+1}(k+1), \nu(k), k+1] + \mathscr{L}_\nu[\mathbf{x}(k), \mathbf{u}_\nu^{l+1}(k)] \leqslant V^l[\mathbf{x}^l(k+1), \nu(k), k+1] + \mathscr{L}_\nu[\mathbf{x}(k), \mathbf{u}_\nu^l(k)]. \tag{6.29}$$

□

Remark 2. *The equality in (6.25) holds iff* $|\nabla G^l[\mathbf{w}_c^\nu(l)]| = 0$*, i.e., iff the optimality condition (6.5) is satisfied. When* $\epsilon \leqslant 2/(L_1L_2 + L_3)L_4$*, (6.28) holds.*

Lemma 4. *Let* $\pi^{l+1} = \{a^{l+1}, \mathbf{c}_1^l, \cdots, \mathbf{c}_E^l\}$ *denote the policy obtained in the* $(l+1)$*th cycle of the hybrid ADP algorithm. Then, given* $\boldsymbol{\lambda}^l[\mathbf{x}, \xi, k]$*, it follows from the discrete and continuous controller updates (6.11) and (6.13) that*

$$\begin{aligned}
&V^l[\mathbf{x}^{l+1}(k+1), \nu^{l+1}(k), k+1] + \mathscr{L}_{\nu^{l+1}}[\mathbf{x}(k), \mathbf{u}_{\nu^{l+1}}^{l+1}(k)] \\
&\leqslant V^l[\mathbf{x}^l(k+1), \nu^l(k), k+1] + \mathscr{L}_{\nu^l}[\mathbf{x}(k), \mathbf{u}_{\nu^l}^l(k)],
\end{aligned} \tag{6.30}$$

where the superscripts l *and* $l+1$ *denote the use of policies* π^l *and* π^{l+1} *in the* l*th and* $l+1$*th cycles, respectively.*

Proof of Lemma 4. The discrete control at time k obtained from (6.11) minimizes $V^l[\mathbf{x}(k), \boldsymbol{\xi}(k), k]$ because

$$V^l[\mathbf{x}(k), \boldsymbol{\xi}(k), k] = \mathbf{x}(k)\boldsymbol{\lambda}(k) + \phi[\mathbf{x}(N)] - \mathbf{x}(N)\boldsymbol{\lambda}(N) + \sum_{\tau=k}^{N-1} H_\nu[\mathbf{x}(\tau), \mathbf{u}_\nu, \boldsymbol{\lambda}, \tau] - \mathbf{x}(\tau)\boldsymbol{\lambda}(\tau), \tag{6.31}$$

from [13], has only one term, $H_\nu[\mathbf{x}(k), \mathbf{u}_\nu, \boldsymbol{\lambda}, k]$, that is a function of $\nu(k)$. According to Lemma 2, for any $\nu \in \mathcal{E}$, $\mathbf{u}_\nu^{l+1}(k)$ has the following property:

$$V^l[\mathbf{x}^{l+1}(k+1), \nu, k+1] + \mathscr{L}_\nu[\mathbf{x}(k), \mathbf{u}_\nu^{l+1}] \leqslant V^l[\mathbf{x}^l(k+1), \nu, k+1] + \mathscr{L}_\nu[\mathbf{x}(k), \mathbf{u}_\nu^l(k)]. \tag{6.32}$$

Therefore, the updated policy π^{l+1} results in a lower cost, i.e.,

$$\begin{aligned} &V^l[\mathbf{x}_{\nu^{l+1}}^{l+1}(k+1), \nu^{l+1}(k), k+1] + \mathscr{L}_{\nu^{l+1}(k)}[\mathbf{x}(k), \mathbf{u}_{\nu^{l+1}}^{l+1}(k)] \\ &\leqslant V^l[\mathbf{x}^l(k+1), \nu^l(k), k+1] + \mathscr{L}_{\nu^{l+1}(k)}[\mathbf{x}(k), \mathbf{u}_{\nu^l}^l]. \end{aligned} \tag{6.33}$$

□

The four Lemmas derived above are used to prove the following result on the convergence of hybrid ADP:

Theorem 1 (Convergence)**.** *Under Assumptions 1, and $T \gg 1$, the policies obtained by the hybrid ADP algorithm 6.11 and (6.14)-(6.13) at every cycle l, $l = 1, 2, \ldots$, are characterized by $V^{l+1}[\mathbf{x}(k), \xi(k), k] \leqslant V^l[\mathbf{x}(k), \xi(k), k]$, for all $\mathbf{x}, \xi, k$. As $l \to \infty$, they converge to their stationary counterparts, i.e., $V^l \to V^\infty$, $\mathbf{u}^l \to \mathbf{u}^\infty$, $\nu^l \to \nu^\infty$, and $\pi^l \to \pi^\infty$. When the Hamiltonian of the system is a convex function in $\mathbf{x}$ and $\mathbf{u}$, the stationary policy is an optimal solution.*

Proof of Convergence. According to Lemma (2), the value function at the lth cycle is

$$V^l[\mathbf{x}(k), \xi(k), k] = \mathscr{L}_{\nu^l(k)}[\mathbf{x}(k), \mathbf{u}_{\nu^l}^l(k)] + V^l[\mathbf{x}^l(k+1), \xi^l(k+1), k+1], \tag{6.34}$$

and the value function at the $(l+1)$th cycle is

$$V^{l+1}[\mathbf{x}(k),\xi(k),k] = \mathscr{L}_{\nu^{l+1}(k)}[\mathbf{x}(k),\mathbf{u}^{l+1}_{\nu^{l+1}}(k)] + V^{l+1}[\mathbf{x}^{l+1}(k+1),\xi^{l+1}(k+1),k+1]. \tag{6.35}$$

Then, the change in the value function during one cycle can be obtained by subtracting (6.34) from (6.35):

$$\begin{aligned} &V^{l+1}[\mathbf{x}(k),\xi(k),k] - V^{l}[\mathbf{x}(k),\xi(k),k] \\ &= \mathscr{L}_{\nu^{l+1}(k)}[\mathbf{x}(k),\mathbf{u}^{l+1}_{\nu^{l+1}}(k)] - V^{l}[\mathbf{x}^{l}(k+1),\xi^{l}(k+1),k+1] \\ &\quad + V^{l+1}[\mathbf{x}^{l+1}(k+1),\xi^{l+1}(k+1),k+1] - \mathscr{L}_{\nu^{l}(k)}[\mathbf{x}(k),\mathbf{u}^{l}_{\nu^{l}}(k)]. \end{aligned} \tag{6.36}$$

From Lemma (4), the change in the value function during one cycle can be shown to have the following upper bound:

$$\begin{aligned} &V^{l+1}[\mathbf{x}(k),\xi(k),k] - V^{l}[\mathbf{x}(k),\xi(k),k] \\ &\leqslant \mathscr{L}_{\nu^{l+1}(k)}[\mathbf{x}(k),\mathbf{u}^{l+1}_{\nu^{l+1}}(k)] - V^{l}[\mathbf{x}^{l+1}(k+1),\xi^{l+1}(k+1),k+1] \\ &\quad + V^{l+1}[\mathbf{x}^{l+1}(k+1),\xi^{l+1}(k+1),k+1] - \mathscr{L}_{\nu^{l+1}(k)}[\mathbf{x}(k),\mathbf{u}^{l+1}_{\nu^{l+1}}(k)] \\ &= V^{l+1}[\mathbf{x}^{l+1}(k+1),\xi^{l+1}(k+1),k+1] - V^{l}[\mathbf{x}^{l+1}(k+1),\xi^{l+1}(k+1),k+1]. \end{aligned} \tag{6.37}$$

We have

$$V^{l}[\mathbf{x}^{l+1}(N),\xi^{l+1}(N),N] = V^{l+1}[\mathbf{x}^{l+1}(N),\xi^{l+1}(N),N] = \phi(\mathbf{x}^{l+1}(N)). \tag{6.38}$$

from the boundary condition (6.8). Thus, from (6.37) it can be concluded that

$$\begin{aligned} &V^{l+1}[\mathbf{x}(k),\xi(k),k] - V^{l}[\mathbf{x}(k),\xi(k),k] \\ &\leqslant V^{l+1}[\mathbf{x}^{l+1}(N),\xi^{l+1}(N),N] - V^{l}[\mathbf{x}^{l+1}(N),\xi^{l+1}(N),N] = 0, \end{aligned} \tag{6.39}$$

and $V^{l+1}[\mathbf{x}(k),\xi(k),k] \leqslant V^{l}[\mathbf{x}(k),\xi(k),k]$ for all $\mathbf{x}(k)$ and all $\xi(k)$. Therefore, the value function obtained during the $(l+1)$th cycle is lower in value than that obtained during the lth cycle and is said to be improved. According to the definition of the value function (6.3), $V^{l}[\mathbf{x}(k),\xi(k),k]$ is non-negative for all $\mathbf{x}(k)$ and all $\xi(k)$.

Because $V^l[\mathbf{x}(k), \xi(k), k]$ is non-negative and $\{V^l\}$ is decreasing as $l \to \infty$, a stationary policy is obtained and is further denoted by π^∞. According to the discrete minimum principle, when $H_\nu[\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, k]$ is convex in $\mathbf{x}$, the stationary solution is an optimal solution *iff* π^∞ minimizes $H_\nu[\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, k]$, given that conditions (6.2) and (6.7) are satisfied. Based on Remark (1), when π^∞ is stationary, the equality in (6.25) must hold, and thus this stationary policy satisfies the optimality condition (6.5). Therefore, for each discrete action, $\mathbf{c}_\nu^\infty$ minimizes $H_\nu[\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, k]$ if $H_\nu[\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, k]$ is also convex in $\mathbf{u}$.

From the above discussion, if $H_\nu[\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, k]$ is convex for each mode in $\mathbf{x}$ and $\mathbf{u}$, π^∞ is an optimal solution. In all other cases, when $H_\nu[\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, k]$ is non-convex, π^∞ is a local optimal solution. For this type of problem, multiple proper initializations are usually provided by [71] or [30] for searching a better local optimal solution. The algorithm terminates when the change in the critic networks (6.36) is below a predefined tolerance. □

Remark 3. *Then, following the above analysis, if* $\epsilon \leqslant 2/(L_1L_2 + L_3)L_4$ *as* $l \to \infty$, $\mathbf{u}^l \to \mathbf{u}^\infty$, *and* $\nu^l \to \nu^\infty$, *the proposed approach provides a local or global optimal policy* π^∞. *When* $\epsilon = 1/(L_1L_2+L_3)L_4$, *the maximum convergence rate is achieved, according to Remark 2, while the convergence cannot be guaranteed if* $\epsilon > 2/(L_1L_2+L_3)L_4$. *The algorithm terminates when the changes in (6.13) and (6.14) are below a predefined tolerance.*

7

Conclusions

This dissertation develops a general and systematic approach for deriving information-driven path planning and control methods that maximize the expected utility of the sensor measurements subject to the vehicle kinodynamic constraints. This approach is used to develop three path planning and control methods: the IP method for integrated path planning and control, the optimized coverage planning based on the DP-GP expected KL divergence, and the optimized visibility planning for simultaneous target tracking and localization.

The IP method is based on a potential function defined from conditional mutual information that is used to design a switched feedback control law, as well as to generate a local PRM for escaping local minima, while obtaining valuable sensor measurements. Theoretical analysis shows that the closed-loop robotic system is asymptotically stable and that an escaping path can be found when the robotic sensor is trapped in a local minimum. Numerical simulation results show that this method outperforms rapidly-exploring random trees and classical potential methods.

The optimized coverage planning method maximizes the DP-GP expected KL divergence approximated by Monte Carlo integration. The variance of the KL ap-

proximation error is proven to decrease linearly with the inverse of the number of samples. Numerical simulations as well as physical experiments show that the optimized coverage planning approach outperforms other applicable algorithms, such as methods based on mutual information, rule-based systems, and randomized planning.

The optimized visibility motion planning approach uses the output of an EKF algorithm to optimize the simultaneous tracking and localization performance of a robot equipped with proprioceptive and exteroceptive sensors, in a GPS denied environment. Numerical simulation results demonstrate that the optimized visibility motion planning approach outperforms the state-of-the-art potential approach.

Because active sensors with multiple modes can be modeled as a switched hierarchical system, the sensor path planning problem can be viewed as a hybrid optimal control problem involving both discrete and continuous state and control variables, where the sensor performance can be represented by two nonlinear Lagrangian functions. This dissertation also presents new recurrence relationships, proof of convergence, and computational complexity for a hybrid ADP approach applicable to switched hybrid systems that are possibly nonlinear. Simulation and theoretic results show that the hybrid ADP approach is capable of converging to the optimal controllers by minimizing the cost-to-go online based on a fully observable state vector.

Appendix A

Algorithms for Optimized Visibility Planning

A.1 Robot Controller

Require: $f_t(\mathbf{x}_t)$, $\mathcal{U}$, and ϵ

$\mathbf{u}_r = \mathbf{u}_0$
while 1 **do**
    $\mathbf{u}'_r \leftarrow \mathbf{u}_r + \eta \frac{\partial}{\partial \mathbf{u}_r(k)} \{P_d[\mathbf{q}_r(k+1)]\}$
    if $\mathbf{u}'_r \notin \mathcal{U}$ **then** Break
    else if $\|\mathbf{u}'_r - \mathbf{u}_r\| \leqslant \epsilon$ **then** Break
    else
        $\mathbf{u}_r \leftarrow \mathbf{u}'_r$
    end if
end while
Return $\mathbf{u}_r$

Appendix B

Algorithms for Hybrid ADP

B.1 Hybrid ADP Critic Network Adaptation

1: Critic Networks Adaptation (T Iterations)
Require: Fix Controller and Actor Networks π^l
Require: $j = 1$, $\mathbf{x}(k)$,$\xi(k)$,k
2: **while** $j \leqslant T$ **do**
3: $\quad \boldsymbol{\lambda}[\mathbf{x}(k), \xi(k), k] = \mathrm{NN}_{\boldsymbol{\lambda}}^{\xi(k)}[\mathbf{x}(k), k, \mathbf{w}_{\boldsymbol{\lambda}}^{\xi(k)}]$
4: $\quad \nu^l(k) = \mathrm{NN}_a^{\xi(k)}[\mathbf{x}(k), k, \mathbf{w}_a^{\xi(k)}]$
5: $\quad \mathbf{u}_{\nu^l}^l(k) = \mathrm{NN}_c^{\nu^l}[\mathbf{x}(k), k, \mathbf{w}_c^{\nu^l(k)}]$
6: $\quad \mathbf{x}^l(k+1) = \mathbf{f}_{\nu^l}[\mathbf{x}(k), \mathbf{u}_{\nu^l}(k)]$
7: $\quad \xi^l(k+1) = \nu^l(k)$
8: $\quad$ **if** $k \leqslant N-1$ **then** $\boldsymbol{\lambda}[\mathbf{x}^l(k+1), \xi^l(k+1), k+1]$
9: $\quad\quad = \mathrm{NN}_{\boldsymbol{\lambda}}^{\xi^l(k+1)}(\mathbf{x}^l(k+1), k, \mathbf{w}_{\boldsymbol{\lambda}}^{\xi^l(k+1)})$
10: $\quad$ **else** $k = 0$
11: $\quad\quad \boldsymbol{\lambda}[\mathbf{x}^l(k+1), \xi^l(k+1), k+1] = \partial_{\mathbf{x}}\phi[\mathbf{x}^l(k+1)]$
12: $\quad$ **end if**
13: $\quad$ Update $\mathbf{w}_{\boldsymbol{\lambda}}^{\xi(k)}$ according (6.14)
14: $\quad j = j+1$, $k = k+1$
15: **end while**

B.2 Hybrid ADP Actor Network Adaptation

Actor Networks Adaptation (M Iterations)
Require: Fix Critic Networks $\boldsymbol{\lambda}^l$
Require: $j = 1$, $\mathbf{x}(k)$,$\xi(k)$,k
while $j \leqslant M$ **do**
  for all $\nu^l \in \mathcal{E}$ **do**
    $\mathbf{u}_\nu^l(k) = \mathrm{NN}_c^{\nu^l}[\mathbf{x}(k), k, \mathbf{w}_c^{\nu^l}]$
    $\xi^l(k+1) = \nu^l(k)$
    $\mathbf{x}^l(k+1) = \mathbf{f}_{\nu^l}[\mathbf{x}^l(k+1), \mathbf{u}_\nu^l(k)]$
    $\boldsymbol{\lambda}[\mathbf{x}^l(k+1), \xi^l(k+1), k+1] = \mathrm{NN}_{\boldsymbol{\lambda}}^{\xi^l(k+1)}[\mathbf{x}^l(k+1), k, \mathbf{w}_{\boldsymbol{\lambda}}^{\xi(k+1)}]$
    Update $\mathbf{w}_c^\nu$ according to (6.13)
  end for
  Obtain $\nu(k)$ and update $\mathbf{w}_a^\nu$ according to (6.11)
  $\mathbf{u}_\nu(k) = \mathrm{NN}_c^{\nu(k)}[\mathbf{x}(k), k, \mathbf{w}_c^\nu]$
  $\mathbf{x}(k+1) = \mathbf{f}_\nu[\mathbf{x}(k), \mathbf{u}_\nu(k)]$
  $\xi(k+1) = \nu(k)$
  $j = j + 1$
end while

Bibliography

[1] A. Al-Tamimi, F. Lewis, and M. Abu-Khalaf. Discrete-time nonlinear hjb solution using approximate dynamic programming: Convergence proof. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 38(4):943–949, aug. 2008.

[2] N. M. Amato, O. B. Bayazit, L. K. Dale, C. Jones, and D. Vallejo. Choosing good distance metrics and local planners for probabilistic roadmap methods. *IEEE Transactions on Robotics and Automation*, 16(4):442–447, 2000.

[3] S. Balakrishnan and V. Biega. Adaptive-critic-based neural networks for aircraft optimal control. *Journal of Guidance, Control, and Dynamics*, 19(4):893–898, 1996.

[4] A. Bellini, W. Lu, R. Naldi, and S. Ferrari. Information driven path planning and control for collaborative aerial robotic sensors using artificial potential functions. In *American Control Conference*, page in press, Portland, OR, 2014.

[5] M. Berg, M. Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry*. Springer, 2000.

[6] D. P. Bertsekas. *Convex Analysis and Optimization*. Athena Scientific, Belmont, MA, 2003.

[7] D. P. Bertsekas and S. E. Shreve. *Stochastic optimal control: The discrete time case*, volume 139. Academic Press New York, 1978.

[8] D. P. Bertsekas and J. N. Tsitsiklis. *Introduction to Probability*. Athena Scientific, Nashua, NH, U.S.A, 2008.

[9] C. Bishop and N. Nasrabadi. *Pattern recognition and machine learning*, volume 1. springer New York, 2006.

[10] J. Boissonnat and S. Oudot. Provably good sampling and meshing of surfaces. *Graphical Models*, 67(5):405–451, 2005.

[11] V. Boor, M. H. Overmars, and A. F.van der Stappen. The Gaussian sampling strategy for probabilistic roadmap planners. volume 2, pages 1018–1023, 1999.

[12] M. Branicky, V. Borkar, and S. Mitter. A unified framework for hybrid control: model and optimal control theory. *IEEE Transactions on Automatic Control*, 43(1):31–45, Jan 1998.

[13] A. Bryson and Y. Ho. *Applied Optimal Control*. Blaisdell Pub. Co Waltham, Mass, 1969.

[14] R. Caflisch. Monte carlo and quasi-monte carlo methods. *Acta Numerica*, 7:1–49, 1998.

[15] C. Cai and S. Ferrari. On the development of an intelligent computer player for CLUE®: a case study on preposterior decision analysis. In *American Control Conference*, pages 4350– 4355, Minneapolis, MN, 2006.

[16] 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.

[17] C. Cai, S. Ferrari, and Q. Ming. Bayesian network modeling of acoustic sensor measurements. In *Proceedings of IEEE Sensors Conference*, pages 345–348, Atlanta, GA, 2007.

[18] G. Casella and R. Berger. *Satistical Inference*. Duxbury Press, 2001.

[19] M. Chu, H. Haussecker, and F. Zhao. Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks. *International Journal of High Performance Computing Applications*, 16(3):293–313, 2002.

[20] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley and Sons, Inc., 1991.

[21] P. Cruz, R. Fierro, W. Lu, and S. Ferrari. Maintaining robust connectivity in heterogeneous robotic networks. In *SPIE Defense, Security, and Sensing*, pages 87410N–87410N. International Society for Optics and Photonics, 2013.

[22] D. Culler, D. Estrin, and M. Srivastava. Overview of sensor networks. *Computer*, 37(8):41–49, 2004.

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

[24] R. Enns and J. Si. Helicopter trimming and tracking control using direct neural dynamic programming. *IEEE Transactions on Neural Networks*, 14(4):929–939, 2003.

[25] T. Ferguson. A bayesian analysis of some nonparametric problems. *The Annals of Statistics*, pages 209–230, 1973.

[26] S. Ferrari, M. Anderson, R. Fierro, and W. Lu. Cooperative navigation for heterogeneous autonomous vehicles via approximate dynamic programming. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 121–127. IEEE, 2011.

[27] S. Ferrari and C. Cai. Information-driven search strategies in the board game of CLUE®. *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, 39(2), 2009.

[28] S. Ferrari, C. Cai, R. Fierro, and B. Perteet. A multi-objective optimization approach to detecting and tracking dynamic targets in pursuit-evasion games. In *Proceedings of American Control Conference*, pages 5316–5321, New York, NY, 2007.

[29] S. Ferrari, R. Fierro, and D. Tolic. A geometric optimization approach to tracking maneuvering targets using a heterogeneous mobile sensor network. In *Proceedings of Decision and Control Conference*, pages 1080–1087, Cancun, MX, Dec 2009.

[30] S. Ferrari and R. Stengel. *Algebraic and Adaptive Learning in Neural Control Systems*. Princeton University, 2002.

[31] S. Ferrari and R.F. Stengel. On-line adaptive critic flight control. *Journal of Guidance, Control, and Dynamics*, 27(5):777–786, 2004.

[32] S. Ferrari and A. Vaghi. Demining sensor modeling and feature-level fusion by bayesian networks. *IEEE Sensors*, 6:471–483, 2006.

[33] S. Ferrari, G. Zhang, and T. A. Wettergren. Probabilistic track coverage in cooperative sensor networks. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 40(6):1492–1504, 2010.

[34] R. Fierro and F. Lewis. A framework for hybrid control design. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 27(6):765–773, nov 1997.

[35] S. Ge and Y. Cui. New potential functions for mobile robot path planning. *IEEE Transactions on Robotics and Automation*, 16(5):615–620, 2000.

[36] C. Gerald and P. Wheatley. *Numerical analysis*. Addison Wesley, 2003.

[37] N. Gordon, B. Ristic, and S. Arulampalam. Beyond the Kalman filter: Particle filters for tracking applications. *Artech House, London*, 2004.

[38] C. Guestrin, A. Krause, and A. Singh. Near-optimal sensor placements in Gaussian processes. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, pages 265–272, New York, NY, USA, 2005. ACM.

[39] G. D. Hager. *Task-Directed Sensor Fusion and Planning: A Computational Approach*. Kluwer Inc, Boston, 1990.

[40] G. D. Hager and M. Mintz. Computational methods for task-directed sensor data fusion and sensor planning. *International Journal of Robotics Research*, 10:285–313, 1991.

[41] H. He, Z. Ni, and J. Fu. A three-network architecture for on-line learning and optimization based on adaptive dynamic programming. *Neurocomputing*, 78(1):3–13, 2012.

[42] R. A. Howard. Information value theory. *IEEE Transactions on Systems Science and Cybernetics*, 2:22–26, 1966.

[43] J. Joseph, F. Doshi-Velez, A. Huang, and N. Roy. A bayesian nonparametric approach to modeling motion patterns. *Autonomous Robots*, 31(4):383–400, 2011.

[44] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein. Energy efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. 2002.

[45] K. Kastella. Discrimination gain to optimize detection and classification. *IEEE Transactions on Systems, Man, and Cybernetics - Part A*, 27(1):112–116, 1997.

[46] L. E. Kavraki, P. Svetska, J. C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration space. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996.

[47] M. Kazemi, M. Mehrandezh, and K. Gupta. Sensor-based robot path planning using harmonic function-based probabilistic roadmaps. *Proceedings of International Conference on Advanced Robotics*, pages 84–89, 2005.

[48] J. Ke and X. Liu. Empirical analysis of optimal hidden neurons in neural network modeling for stock prediction. In *Computational Intelligence and Industrial Application, 2008. PACIIA '08. Pacific-Asia Workshop on*, volume 2, pages 828–832, Dec 2008.

[49] H. Khalil. *Nonlinear Systems*. Prentice Hall, Upper Saddle River, NJ, 2002.

[50] Y. Koren and J. Borenstein. Potential field methods and their inherent limitations for mobile robot navigation. In *Proceedings of IEEE Conference on Robotics and Automation*, pages 1398–1404, Sacramento, CA, 1991.

[51] A. Krause and C. Guestrin. Nonmyopic active learning of Gaussian processes: an exploration-exploitation approach. In *Proceedings of the 24th International Conference on Machine Learning*, pages 449–456. ACM, 2007.

[52] 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.

[53] X. C. Lai, S. S. Ge, and A. Al-Mamun. Hierarchical incremental path planning and situation-dependent optimized dynamic motion planning considering accelerations. *IEEE Transactions on Systems, Man, and Cybernetics- Part A*, 37(6):1541–1554, 2007.

[54] F. Lamiraux and J. P. Laumond. On the expected complexity of random path planning. In *IEEE Int. Conf. Robot. & Autom*, pages 3306–3311, 1996.

[55] J. C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[56] S. LaValle and J. Kuffner. Randomized kinodynamic planning. *The International Journal of Robotics Research*, 20(5):378–400, 2001.

[57] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006. Available at http://planning.cs.uiuc.edu/.

[58] G. F. Lawler. *Introduction to Stochastic Processes*. Chapman & Hall/CRC, Boca Raton, FL, 2006.

[59] F. Lewis and D. Liu. *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*. John Wiley and Sons, 2012.

[60] A. Lingas. The power of non-rectilinear holes. *Proceedings of Colloquium on Automata, Languages and Programming*, pages 369–383, 1982.

[61] D. Liu, H. Li, and D. Wang. Neural-network-based zero-sum game for discrete-time nonlinear systems via iterative adaptive dynamic programming algorithm. *Neurocomputing*, 110(0):92–100, 2013.

[62] D. Liu, D. Wang, and X. Yang. An iterative adaptive dynamic programming algorithm for optimal control of unknown discrete-time nonlinear systems with constrained inputs. *Information Sciences*, 220(0):331–342, 2012.

[63] W. Lu and S. Ferrari. An approximate dynamic programming approach for model-free control of switched systems. In *CDC*, pages 3837–3844, 2013.

[64] W. Lu, S. Ferrari, R. Fierro, and T. A. Wettergren. Approximate dynamic programming recurrence relations for a hybrid optimal control problem. In *proceedings of the Society of Photographic Instrumentation Engineers, Unmanned Systems Technology XIV*, 2012.

[65] W. Lu, H. Wei, and S. Ferrari. A Kalman-particle filter for estimating the number and state of multiple targets. In *Proceedings of International Conference on Management Sciences and Information Technology*, Changsha, China, 2012.

[66] W. Lu, G. Zhang, and S. Ferrari. A randomized hybrid system approach to coordinated robotic sensor planning. In *IEEE Conference on Decision and Control*, pages 3857–3864, 2010.

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

[68] W. Lu, G. Zhang, and S. Ferrari. An information potential approach to integrated sensor path planning and control. *IEEE Transactions on Robotics*, in press.

[69] W. Lu, G. Zhang, S. Ferrari, M. Anderson, and R. Fierro. A particle-filter information potential method for tracking and monitoring maneuvering targets using a mobile sensor agent. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, 11(1):47–58, 2014.

[70] 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.

[71] D. Nguyen and B. Widrow. Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. In *Neural Networks, 1990., 1990 IJCNN International Joint Conference on*, pages 21–26. IEEE, 1990.

[72] Z. Ni and H. He. Adaptive learning in tracking control based on the dual critic network design. *IEEE Transactions on Neural Networks and Learning Systems*, 24(6):913–928, june 2013.

[73] R. Padhi, N. Unnikrishnan, X. Wang, and S. N. Balakrishnan. Optimal control synthesis of a class of nonlinear systems using single network adaptive critics. *Neural Networks*, 19(1):1648–1660, 2006.

[74] J. Park, R. Harley, and G. Venayagamoorthy. Adaptive-critic-based optimal neurocontrol for synchronous generators in a power system using mlp/rbf neural networks. *IEEE Transactions on Industry Applications*, 39(5):1529–1540, 2003.

[75] R. E. Parr. *Hierarchical Control and Learning for Markov Decision Processes*. Ph.D. Thesis, University of California, Berkeley, CA, 1998.

[76] K. Pathak and S. Agrawal. An integrated path-planning and control approach for nonholonomic unicycles using switched local potentials. *IEEE Transactions on Robotics*, 21(6):1201–1208, 2005.

[77] N. Rao. Robot navigation in unknown generalized polygonal terrains using vision sensors. *IEEE Transactions on System, Man, and Cybernetics*, 25(6):947–962, 1995.

[78] P. Riedinger, F. Kratz, C. Iung, and C. Zane. Linear quadratic optimization for hybrid systems. In *Proceedings of the Conference on Decision and Control*, volume 3, pages 3059–3064, Phoenix, AZ, 1999.

[79] L.I. Rozonoer. Ls pontryagin maximum principle in the theory of optimum systems. i, ii, iii. *Automatic Remote Control*, 20:1288–1302, 1959.

[80] S. Russell and P. Norvig. *Artificial Intelligence A Modern Approach*. Prentice Hall, Upper Saddle River, NJ, 2003.

[81] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2nd edition, 2003.

[82] W. Schmaedeke. Information based sensor management. In *Proc. of SPIE Signal Processing, Sensor Fusion, and Target Recognition II*, volume 1955, pages 156–164, Orlando, FL, 1993.

[83] C. Seatzu, D. Corona, A. Giua, and A. Bemporad. Optimal control of continuous-time switched affine systems. *IEEE Transactions on Automatic Control*, 51(5):726–741, may 2006.

[84] J. Si, A. Barto, and W. Powell. *Learning and Approximate Dynamic Programming*. John Wiley and Sons, 2004.

[85] R. Siegel. Land mine detection. *IEEE Instrumentation and Measurement Magazine*, 5(4):22–28, 2002.

[86] A. Singh, A. Krause, C. Guestrin, W. Kaiser, and M. Batalin. Efficient planning of informative paths for multiple robots. In *IJCAI*, volume 7, pages 2204–2211, 2007.

[87] K. Song and C. C. Chang. Reactive navigation in dynamic environment using a multisensor predictor. *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, 29(6):870–880, 1999.

[88] R. F. Stengel. *Optimal Control and Estimation*. Dover Publications, Inc., 1986.

[89] G. W. Stewart. Gershgorin theory for the generalized eigenvalue problem $ax = \lambda bx$. *Mathematics of Computation*, 29(130), 1975.

[90] Z. Sun, D. Hsu, T. Jiang, H. Kurniawati, and J. H. Reif. Narrow passage sampling for probabilistic roadmap planning. *IEEE Transactions on Robotics*, 21(6):1105–1115, 2005.

[91] Z. Sun and J.H. Reif. On robotic optimal path planning in polygonal regions with pseudo-eucledian metrics. *IEEE Transactions on Systems, Man, and Cybernetics - Part A*, 37(4):925–936, 2007.

[92] Y. Teh, M. Jordan, M. Beal, and D. Blei. Hierarchical dirichlet processes. *Journal of the american statistical association*, 101(476), 2006.

[93] P. Trautman and A. Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In *Intelligent Robots and Systems, IEEE International Conference on*, pages 797–803. IEEE, 2010.

[94] K. Vamvoudakis and F. Lewis. Multi-player non-zero-sum games: Online adaptive learning solution of coupled hamilton-jacobi equations. *Automatica*, 47(8):1556–1569, 2011.

[95] R.S. Varga. *Matrix Iterative Analysis*. Springer Series in Computational Mathematics. Springer, 2009.

[96] D. Vrabie and F. Lewis. Adaptive dynamic programming for online solution of a zero-sum differential game. *Journal of Control Theory and Applications*, 9(3):353–360, 2011.

[97] D. Wang, D. Liu, and Q. Wei. Finite-horizon neuro-optimal tracking control for a class of discrete-time nonlinear systems using adaptive dynamic programming approach. *Neurocomputing*, 78(1):14–22, 2012.

[98] F. Wang, N. Jin, D. Liu, and Q. Wei. Adaptive dynamic programming for finite-horizon optimal control of discrete-time nonlinear systems with ϵ error bound. *Neural Networks, IEEE Transactions on*, 22(1):24–36, 2011.

[99] F. Wang and G. Saridis. On successive approximation of optimal control of stochastic dynamic systems. *International Series in Operations Research and Management Science*, pages 333–358, 2002.

[100] H. Wei and S. Ferrari. A geometric transversals approach to analyzing the probability of track detection for maneuvering targets. *Computers, IEEE Transactions on*, PP(99):1–1, 2013.

[101] H. Wei, W. Lu, and S. Ferrari. An information value function for nonparametric Gaussian processes. In *Proceedings of Neural Information Processing Systems Conference*, Lake Tahoe, NV, 2012. NIPS.

[102] H. Wei, W. Lu, and S. Ferrari. An information value function for nonparametric Gaussian processes. In *Proceedings of Neural Information Processing Systems Conference*, Lake Tahoe, NV, 2012. NIPS.

[103] H. Wei, W. Lu, S. Ferrari, Robert H Klein, Shayegan Omidshafiei, and Jonathan How. Camera control for learning nonlinear target dynamics via bayesian non-parametric Dirichlet-process Gaussian-process (DP-GP) models. In *IROS*, page accepted, Chicago, IL, USA, 2014.

[104] H. Wei, W. Lu, P. Zhu, S. Ferrari, Guoquan Huang, and John Leonard. Visibility-based motion planning for active target tracking and localization. In *IROS*, page accepted, Chicago, IL, USA, 2014.

[105] 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 (CDC), IEEE Annual Conference on*, pages 6778–6785, Florence, Italy, Dec 2013.

[106] Q. Wei and D. Liu. Finite horizon optimal control of discrete-time nonlinear systems with unfixed initial state using adaptive dynamic programming. *Journal of Control Theory and Applications*, 9(3):381–390, 2011.

[107] Q. Wei and D. Liu. Stable iterative adaptive dynamic programming algorithm with approximation errors for discrete-time nonlinear systems. *Neural Computing and Applications*, pages 1–13, 2013.

[108] G. Welch and G. Bishop. An introduction to the Kalman filter. 1997.

[109] X. Xu and P. Antsaklis. Optimal control of switched systems based on parameterization of the switching instants. *Automatic Control, IEEE Transactions on*, 49(1):2–16, Jan. 2004.

[110] M. Zavlanos and G. Pappas. Distributed hybrid control for multiple-pursuer multiple-evader games. In *Hybrid Systems: Computation and Control*, volume 4416, pages 787–789. Lecture Notes in Computer Science, 2007.

[111] 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.

[112] G. Zhang, S. Ferrari, and M. Qian. An information roadmap method for robotic sensor path planning. *Journal of Intelligent and Robotic Systems*, 56:69–98, 2009.

[113] G. Zhang, S. Ferrari, and M. Qian. Information roadmap method for robotic sensor path planning. *Journal of Intelligent and Robotic Systems*, in press, 2009.

[114] H. Zhang, Y. Luo, and D. Liu. Neural-network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraints. *IEEE Transactions on Neural Networks*, 20:1490–1503, 2009.

[115] H. Zhang, Q. Wei, and D. Liu. An iterative adaptive dynamic programming method for solving a class of nonlinear zero-sum differential games. *Automatica*, 47(1):207–214, 2011.

[116] Y. Zhang, S. Pang, and Y. Wang. Orthogonal arrays obtained by generalized hadamard product. *Discrete Mathematics*, 238(1):151–170, 2001.

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

[118] K. Zhou and S. Roumeliotis. Multirobot active target tracking with combinations of relative observations. *Robotics, IEEE Transactions on*, 27(4):678–695, 2011.

[119] D. Zhu and J. C. Latombe. New heuristic algorithms for efficient hierarchical path planning. *Robotics and Automation, IEEE Transactions on*, 7(1):9–20, Feb 1991.

[120] P. Zhu, B. Chen, and J. Príncipe. Extended Kalman filter using a kernel recursive least squares observer. In *Proc. of Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 1402–1408, San Jose, California, USA, Aug 2011.

[121] P. Zhu, B. Chen, and J. Príncipe. A novel extended kernel recursive least squares algorithm. *Neural Networks*, 32(0):349 – 357, 2012.

[122] D. Zielinski, R. Kopper, R. McMahan, W. Lu, and S. Ferrari. Intercept tags: Enhancing intercept-based systems. In *Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology*, VRST '13, pages 263–266, New York, NY, USA, 2013. ACM.

[123] D. Zielinski, R. McMahan, W. Lu, and S. Ferrari. Ml2vr: providing matlab users an easy transition to virtual reality and immersive interactivity. In *Virtual Reality (VR), 2013 IEEE*, pages 83–84. IEEE, 2013.

Biography

Wenjie Lu born in Yuyao, China, on Match 8th 1986, received a Ph.D and a M.S. degree in mechanical engineering from the Duke University, USA in 2014, 2011, and B.S. degree in mechatronic engineering from Zhejiang University, China, 2009. His research focuses on intelligent mobile sensor agent that can adapt to heterogeneous environmental conditions, to achieve the optimal performance, via approximate dynamic programming, information value functions, and non-parametric Bayesian models.

He has published [68], [67], [69], [66], and [4] on information driven sensor path planning, [103] and [102] on information value functions for Bayesian Non-Parametric models, [63], [21], [64], and [26] on approximate dynamic programming for hybrid systems, [70], [104], and [65] on moving target tracking, [123] and [122] on providing MATLAB users an easy transition to virtual reality and immersive interactivity.