Multi-Robot Resource Constrained Decentralized Active Sensing for Online Environment Monitoring

Kshitij Tiwari

Japan Advanced Institute of Science and Technology

Doctoral Dissertation

Multi-Robot Resource Constrained Decentralized Active Sensing for Online Environment Monitoring

Kshitij Tiwari

Supervisor: Professor Nak Young Chong

School of Information Science
Japan Advanced Institute of Science and Technology

March 2018

Abstract

This thesis addresses the problem of trajectory planning over discrete domains for monitoring an environmental phenomenon that is varying spatially. The most relevant application corresponds to environmental monitoring using an autonomous mobile robot for air, water or land pollution monitoring. Since the dynamics of the phenomenon are not known *a priori*, the planning algorithm needs to satisfy two objectives simultaneously: 1) Learn and predict spatial patterns and, 2) adhere to resource constraints while gathering observations. Subsequently, the thesis brings the following contributions:

Firstly, it formulates a resource constrained information-theoretic path planning scheme called *Resource Constrained Decentralized Active Sensing (RC-DAS)* that can effectively trade-off model performance to resource utilization. Since, these objectives are inherently conflicting, optimizing over both these objectives is rather challenging. However, weighted combination of these objectives into a single objective function is proposed such that the total path length is bounded by the maximum operational range. This path planner is then coupled with a distributed Gaussian Process (DGP) framework to allow the robots to simultaneously infer and predict the dynamics of the environment of interest.

Secondly, optimal weight selection method is proposed wherein the weights of the *RC-DAS* cost function are dynamically updated as a function of residual resources. This extended scheme is referred to as *RC-DAS*† which additionally ensures that the robots return to base station at the end of their respective mission times. This prevents the robots from getting stranded amidst the field and is a first step towards making the architecture fail-proof.

Thirdly, an operational range estimation framework is proposed to interpret the bounds on maximum path length attainable by the robots. This should be used as the limiting condition for terminating the exploration to ensure a safe path to the base station. This framework is then generalized to encompass various classes of robots and is made robust to operate with high accuracy even when subject to natural environmental disturbances like strong wind gusts or uneven terrains.

Fourthly, the *RC-DAS* framework is scaled to multiple robots operating in a fully decentralized fashion in communication devoid environments. Owing to such a setting, multiple inferred models of the environment can be obtained. However, neither all models can be fully trusted nor forthrightly rejected. To solve this dilemma and to obtain one globally consistent model, a pointwise fusion of distributed GP models is introduced and referred to as *FuDGE*.

Keywords: *Active Sensing, Decentralized Multi-robot Teams, Resource Constraints, Map Fusion, Range Estimation.*

Review Panel

“ Feedback is the breakfast of champions. ”

Kenneth H. Blanchard, *2015*

Author	Kshitij Tiwari
Member and Advisor	Nak Young Chong, Professor at School of Info. Sc., JAIST
Member	Tadashi Matsumoto, Professor at School of Info. Sc., JAIST
Member	Tu Bao Ho, Professor at School of Knowledge Sc., JAIST
Member	Han-Lim Choi, Associate Professor at Div. of Aerospace Engg., KAIST
Member	Jongeun Choi, Associate Professor at School of Mech. Engg., Yonsei University

Acknowledgments

> *How I did it, you ask ? I took a calculated risk and chose what side of history I wanted to be on, when the dust settles.*
>
> Kshitij Tiwari, *2017*

The author wishes to express his sincere gratitude to his principal adviser Professor Nak Young Chong of Japan Advanced Institute of Science and Technology (JAIST) for his constant encouragement and kind guidance during this work. The author would also like to thank Professor Fumihiko Asano and Dr. Sungmoon Jeong of JAIST for helpful discussions and suggestions. Aside from the faculty members associated with Robotics Lab @ JAIST, the author would also like to express his gratitude to Professor Xaveir Defago and Assistant Professor Francois Bonnet who, during their stay at JAIST, taught the author about a myriad of aspects, both, academically and otherwise. A special mention for the committee members: Professor Tadashi Matsumoto (JAIST), Professor Tu Bao Ho (JAIST), Dr. Han-Lim Choi (KAIST) and Dr. Jongeun Choi (Yonsei Univeristy) for their valuable comments and constructive criticism that have helped enhance the quality of this work.

Besides the academic council mentioned above, the author would also like to thank his father, Dr. Anupam Tiwari who kept the author's spirit up when times were tough. The author benefited a lot from the teachings of his father and always sought counsel when the progress seems to have bogged down. The author is thankful to his family away from home *viz.,* Dr. A.A.R. Newaz, Valentin Honoré, Alessandro Carfi, Quyên Dang and other colleagues for their constant emotional support that helped him tread through the road less traveled and attain the goal.

Dedication

> *Life is like a sledge hammer, out there to make you or break you. It is upto you to either be like glass and get shattered or be like steel and get forged.*

Kshitij Tiwari, *2017*

In loving memory of my Mother. Losing you was the toughest hurdle, thus far, that life threw at me but I kept going on to live up to your expectations. I hope you are feeling proud. May you Rest in Peace.

Preface

> *I transformed my wounds into wisdom, criticism into motivation, pain into power and defeats into stepping stones to proceed with even more vigor compared to yesterday. The only easier day was yesterday and all this is a part and parcel of life.*

Kshitij Tiwari, *2017*

Kshitij Tiwari was born in a small town called Khetri located in Jhunjhunu district at the foothills of the Aravalli range of Rajasthan, India as the only child of Dr. Anupam Tiwari and Nisha Tiwari. He obtained his Bachelors Degree from the University of Hong Kong, Hong Kong S.A.R. of China in Electronics and Communication Engineering (EComE) in 2013 from the Faculty of Electrical and Electronics Engineering. Soon after, the author enrolled at the University of Edinburgh, Scotland from where he obtained his Masters in Artificial Intelligence in 2014 with a special focus on Intelligent Robotics. Since 2015, he has been pursuing his Ph.D. under the supervision of Professor Nak Young Chong @ Robotics Lab, School of Information Science, Japan Advanced Institute of Science and Technology (JAIST).

This thesis is the final work and submitted as a partial fulfillment for the degree of Doctor of Philosophy (Ph.D.) from JAIST. This work is aimed to reach out to research scholars who have prior experience with state-of-the-art machine learning approaches and real robot hardware. Irrespective of the skill-set of the readers, this work is designed to be self-sufficient as all the necessary details are enclosed herewith. As a part of this venture, the author got the chance to setup several international collaborations, mentor peers and work with researchers from different domains. Some of these collaborations were even transformed into publications.

Kshitij Tiwari | Date

Declaration

> *Create your own style so that you are known for who you are and not for who you resemble.*

Kshitij Tiwari, *2017*

I hereby declare that this submission is my own work and that, to the best of my comprehension, it does not significantly overlap with materials published by others as a part of another degree or diploma awarded by other institutions of higher education, except whereby acknolwedged explicitly. Most of the research contributions appearing in this dissertation are either already published or in several stages of publication at various venues and their appropriate details are provided herewith.

Kshitij Tiwari | Date

Contents

List of Figures

List of Tables

List of Algorithms

Nomenclature

Acronyms

AE	Ancillary Energy
CEO	Central Estimation Officer
CRLB	Cramer Rao Lower Bound
DDD	Dirty, Dangerous and Dull
FC	Fusion Center
FIM	Fisher Information Matrix
FuDGE	Fusion of distributed GP Experts
full-DAS	Entropy Maximization
GP	Gaussian Process
gPoE	Gaussian Product of Experts
GTGP	Ground Truth GP
HSIC	Hilbert-Schmidt Independence Criterion
LML	Log Marginal Likelihood
ME	Maneuvering Energy
MLE	Maximum Likelihood Estimate
MoE	Mixture of Experts
MR-GP	Map-Reduce GP
NN	Nearest Neighbors

OGM	Occupancy Grid Map
RBF	Radial Basis Function
RC-DAS	Resource Constrained Decentralized Active Sensing
RC-DAS†	RC-DAS + Homing
RMSE	Root Mean Squared Error
TE	Traversal Energy
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
UMV	Unmanned Marine Vehicle
WRMSE	Weighted RMSE

Notation Conventions

$(\cdot)^T$	Transpose
$\#(A)$	Set cardinality
$\bigodot^{[0:end]}$	Value of the quantity $\bigodot$ from $t = 0$ until end of mission
$\bigodot^{[0:t]}$	Value of the quantity $\bigodot$ from $t = 0$ to $t = t$
$\bigodot^{[t:end]}$	Value of the quantity $\bigodot$ from $t = t$ until end of mission
$\mathbf{x}$	multi-dimensional variable or Vector
$\overline{[*]}$	Average quantity $*$
$\mathrm{tr}(A)$	Trace of a matrix
$\widehat{[*]}$	Estimated quantity $*$
A^{-1}	Matrix inversion
A	Matrix
x	scalar

Gaussian Process

$(\mathbf{s}, t)$	Spatiotemporal input with spatial component $\mathbf{s}$ and temporal component t
$\boldsymbol{\mu}(\mathbf{x}^*)$	set of all predictions over inputs $\mathbf{x}^*$
$\boldsymbol{\mu}_{\mathbf{gt}}(\mathbf{x})$	set of all predictions over inputs $\mathbf{x}$ from the GTGP.
$\boldsymbol{\theta}$	set of true hyper-parameters
$\hat{\boldsymbol{\theta}}$	set of estimated hyper-parameters
$\mathbf{x}$	Training data (inputs)
$\mathbf{x}^*$	Test data (inputs)
$\mathbf{y}$	Training observations (targets)
$\mathbf{y}^*$	Test observations (targets)
$\mathcal{L}$	Log Marginal Likelihood
σ_n	Noise Variance
σ_{sig}	Signal Variance
$K(\cdot, \cdot)$	Noise free spatial covariance (kernel) function
$K(x, x^*)$	Covariance between inputs X and X^*
$K^{ST}(\cdot, \cdot)$	Noise free spatiotemporal covariance (kernel) function
$K_\epsilon^{ST}(\cdot, \cdot)$	Noisy spatiotemporal covariance (kernel) function
$K_\epsilon(\cdot, \cdot)$	Noisy spatial covariance (kernel) function
O_{global}	Union of all observed sets
U_{global}	Intersection of all unobserved sets
$x >$	Best Location

Range Estimation

η_1	Battery charge storage loss
η_2	Drive motor losses

η_3	Mechanical losses
η_4	Ancillary losses
Γ	Figure of merit for propellers
γ	Average terrain elevation
Ω	Overall system efficiency
θ	Average terrain smoothness
ζ	Number of charge-discharge cycles of power source
C_{rr}	Coefficient of rolling resistance
D	Duty cycle
d	Operational range
$E_O, \tilde{E}$	Original and reduce power source energy
f_s	Sampling frequency for sensors
f_{comm}	Data transmission frequency
g	Gravitational acceleration
m_R	Mass of the robot
N	Normal
N_R	Number of propellers
P_C	Power for wireless communication and computations
P_{anc}	Power consumed by ancillary branch
r_p	Radius of propellers
t	Age of power source
T_M	Net mission time
T_S	Net stopping time
v	Operational velocity

$W_{R_\parallel}, W_{R_\perp}$	Components of weight of robot
${}^r\Omega_{anc}$	Efficiency of ancillary branch
${}^r\Omega_{Man}$	Efficiency of maneuvering branch

Publications

Conference Articles

[1] **K. Tiwari**, V. Honoré, S. Jeong, N.Y. Chong and M.P. Deisenroth: "Resource-Constrained Decentralized Active Sensing for Multi-Robot Systems using Distributed Gaussian Processes". In Proc. of 16th International Conference on Control, Automation, and Systems (ICCAS), pp. 13-18 (Oct. 2016). Received **Best Student Paper Award**.

[2] **K. Tiwari**, S. Jeong and N.Y. Chong: "Multi-UAV Resource Constrained Online Monitoring of Large-scale Spatio-temporal Environment with Homing Guarantee". In Proc. of 43rd Annual Conference IECON of the IEEE Industrial Electronics Society (IES), pp. 5893-5900 (Oct. 2017).

[3] **K. Tiwari**, S. Jeong and N.Y. Chong: "Map-Reduce Gaussian Process (MR-GP) for Multi-UAV based Environment Monitoring with with Limited Battery". In Proc. of the 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), pp. 760-763 (Sept. 2017).

[4] **K. Tiwari**, X. Xiao and N.Y. Chong: "Estimating Achievable Range of Ground Robots Operating on Single Battery Discharge for Operational Efficacy Amelioration". In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3991-3998 (Oct. 2018).

Journal Articles

[5] **K. Tiwari**, S. Jeong and N.Y. Chong: "Point-wise Fusion of Distributed Gaussian Process Experts (*FuDGE*) using a Fully Decentralized Robot Team operating in Communication Devoid Environments". In IEEE Transaction on Robotics (T-RO), 34 (3), pp. 820-828 (June 2018).

[6] **K. Tiwari**, X. Xiao, A. Malik and N.Y. Chong: "A Unified Framework for Operational Range Estimation of Mobile Robots Operating on a Single Discharge to Avoid Complete Immobilization". In Mechatronics, In Press.

Part I

Foundation: The Basics

Chapter 1

Introduction

> *Everything is related to everything else, but near things are more related than distant things.*

Waldo Tobler, *1st Law of Geography*, 1970

Environmental issues like water quality monitoring [1–4], ambient air quality monitoring [5–7], climate change [8–10] and renewable resource depletion [11] are increasingly becoming important to the research community. This owes to the fact that drastic changes in the environment as they are witnessed today, *e.g.,* the Tsunami that was triggered by an earthquake in the Indian Ocean in 2004 that devastated $230,000$ human lives and anything else that stood in its wake or the melting of glaciers owing to global warming, are all intimately correlated with the actions of the human kind over long periods of time. Since the past cannot be changed and the future is yet to come, the best bet is to handle the existing environmental issues and efficiently decipher the underlying dynamics to ensure sustainability in the future. To assist with this mission, both *robotics* and *artificial intelligence* would go a long way. Recent decade has seen significant developments in deployment of robots for environmental monitoring [12–16]. In light of the recent developments, this work aims to present novel path planning approaches for decentralized multi-robot teams endowing them with the capabilities to decipher the most *significant* areas of the environment that needs to be monitored while simultaneously learning a model of complex variations of environment being monitored.

1.1 Motivation

As Waldo Tobler said, it is a known fact that *correlation* $\propto$ *proximity* but the underlying dynamics of the environmental phenomenon are highly complex. So much so that they cannot be represented by ordinary [17] or even stochastic [18] differential equations for non-linear systems. An alternative solution would then be to use machine

learning techniques which require gathering of the training data. Traditionally, this was done using static sensors but when the domain has a large spatial expanse and the resolution desired is high, this translates into positioning a very large array of sensors across the domain of interest [19]. Besides, the position of the sensors must also be pre-determined. To remedy these drawbacks, the optimal solution would be to deploy autonomous robot or a team thereof, each equipped with apt sensors along with on-board computational resources to simultaneously develop a model of the environment. The advantages of doing so are:

- Robots can utilize befitting path planners endowing them to choose the locations to sample the information from.
- A few robots can cover more locations from which the information needs to sampled in lesser time as compared to an array of static sensors.
- When considering temporal evolution of the environment phenomenon, the position of the array of sensors needs to be manually adjusted through time while the robots can adjust on-the-fly.
- Sometimes the environment being monitored can be dirty, dangerous or dull (DDD) [20] thereby, making human intervention infeasible but robots can sustain without any hassle.
- Increased dependence on robots reduces the chances of human error owing to stress during the mission.

Every coin has a flip-side and this setting too has disadvantages. Some of them are:

- Although recent developments have expanded the hardware capabilities of the robot significantly but the machine learning approaches utilized are data-hungry. This means that the model requires hoards of data to be acquired but the resources (*e.g., battery capacity, flight time, payload capability etc.,*) all limit the duration of the mission. In other words, the robot resources are always at conflict with the model requirements.
- The robot needs to be self-sustaining *i.e.,* it should be able to actively select the locations which must be sampled whilst iteratively updating the model being inferred. This incurs computational costs and all the of computational hardware also draws power from the same source that allows the robot to navigate; thereby, affecting the maximum attainable range.
- Deploying robots in real world requires accurate localization and obstacle avoidance for both dynamic and static obstacles.
- Real-time performance guarantees need to be provided for navigation within and monitoring of the environmental phenomenon of interest.

- Having a multi-robot team in a large sensing area also calls for selecting the nature of communication architecture *i.e., disconnected,* where no robot talks to its peers or base station, *partially-connected* where the robot talks only to its nearest neighbor(s) and *connected* wherein an all-to-all communication overhead is incurred. In cases where communication is deemed necessary, either for *partially-connected* or *connected* architecture, further lacunae caused due to data losses, latency, signal interference *etc.,* need to be addressed. The choice of these communication mechanisms are largely dependent on the nature of the environment in which the robot team is being deployed and the preferences set by the supervisor.

- In a multi-robot team, besides communication protocols, data being acquired needs to be processed for which either a *fully-decentralized* approach can be taken wherein each robot processes the data locally acquired, or, *partially-decentralized* approach where data from several peers are assimilated before processing on a local base node, or, *centralized* approach wherein all the data from all the robots is streamed wirelessly to a central base node.

All-in-all, the long-term perspective is to be able to deploy a fully decentralized and disconnected team of robots in a very large scale unknown environment which are capable of developing accurate models of the target phenomenon. These models can later be utilized by humans for ensuring sustainability and making the living environment amicable to live-stock. Having said this, it must be pointed out that this goal is rather far fetched and as a stepping stone towards this aim, this work presents a novel path planning approach suitable to some real-world scenarios like those shown in Fig. 1.1. In all the scenarios like algal bloom monitoring, oil spill analysis, forest-fire containment, or nuclear radiation fallout analysis, the common aspect that binds them together is that the area to be monitored is large, unknown and mostly dangerous for humans to intervene, while the robot being used to aid the process has limited resources. Additionally, in a decentralized setting, multiple robots generate multiple models but the supervisor cannot perform a *many-to-one* correlation analysis to deduce the overall dynamics. To solve this issue, the author also proposes a novel map fusion approach where a point-wise weighted fusion of maps is performed to eventually deduce overall density map.

1.2 Related Works

Researchers in different domains have made several attempts at solving some subsets of the problem that the author is interested in solving. Be it the domain of combinatorics, robotics, or information theory, several related works exist that can be considered as starting point to address the problem at hand. They are briefly categorized below as path planning related or resource optimization related works and finally their limitations are highlighted thereby emphasizing the novel contributions and significance of this work for research community.

Figure 1.1: Environmental Monitoring Scenarios. The top row shows algal bloom [21] on the left which can be monitored using surface vehicles as shown on the right [22]. The second row shows the oil spill off the coast of Gulf of Mexico [23] which can be monitored using underwater gliders [24] and the third row shows forest fires [25] which can be monitored using multi-rotor UAVs [26]. The bottom row shows the fukushima daichii nuclear disaster [27] and one of the robots that were used to assess the damage [28].

1.2.1 Path Planning for Reward Maximization

This section aims at addressing some categories of path planning literature where the objective is to plan tours across spatially dispersed nodes such that the reward accrued is maximal. For some, the objective is to visit all nodes whilst for others the objective is to maximize reward under path length constraints.

1.2.1.1 Orienteering Problem (OP)

Orienteering Problem or OP is a well researched problem that has been around for over a decade. Originating in 1996 from the works of [29], it refers to the research problem where given a collection of spatially distributed nodes, a subset of the nodes must be selected and a shortest path must be calculated to visit the selected nodes while maximizing the accruable reward [30]. Whilst the main objective is node selection coupled with reward maximization, the optimization routine is further constrained by "budget" *e.g.,* time limitations *etc.* This renders an exhaustive visit to all nodes infeasible. Furthermore, works like that of [31] solve the Correlated Orienteering Problem in which the rewards are spatially correlated across nodes as shown in Fig. 1.2 while others like [32] solve the spatiotemporal correlated orienteering problem or alternatively known as the Vehicle Routing Problem [33] where the rewards are varying with time across the spatial domain as shown in Fig. 1.3. The OP has been extended to a Team Orienteering Problem or TOP wherein the problem is extended to multiple agents. Works like [34], use the ant-colony based optimization approaches to solve this by constraining the path length of each agent and maximizing the reward accrued.

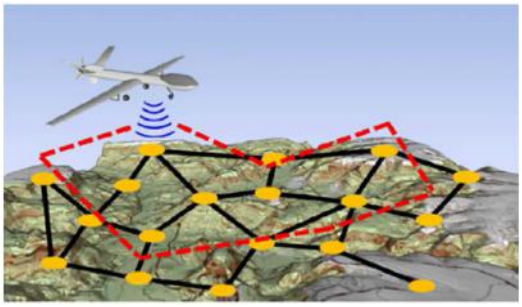


Figure 1.2: Correlated Orienteering [31]

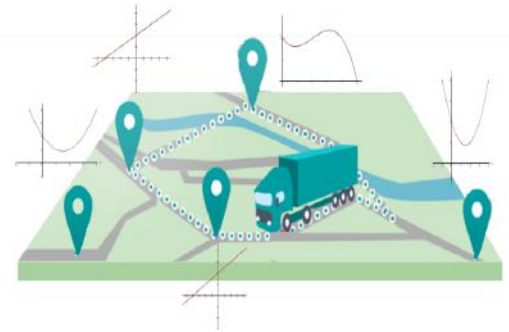


Figure 1.3: Spatiotemp. Orienteering [32]

1.2.1.2 Knapsack Problem

Knapsack problem [35] or the Rucksack problem refers to the problem of maximizing the reward given a collection of items each associated with their respective weights. *E.g.,* while preparing the backpack for a hiking trip, only the absolutely necessary items must be packed. The heavier the rucksack, the harder it will be to climb. Each item packed in the rucksack represents a tuple $< weight, reward >$ where the objective is

to maximize the reward while maintaining the net weight below the load-bearing limit of the rucksack. This problem again is analogous to the OP, albeit, being indirectly related to path planning *i.e.,* the resources that a hiker packs with him/her gets depleted as the time progress and efficient utilization of resources ensures a convenient hike. Additional applications of the knapsack problem include a file storage system with limited storage space at the server or a store robbery with a limited payload that can be carried by the thief as shown in Fig. 1.4 and Fig. 1.5 respectively. In the file storage case, the objective is to store majority of the important files locally at the server to reduce latency while for that of the robbery example, the object is to maximize the monetary reward that can be accrued by selling off stolen merchandize. A natural generalization to multiple knapsack problem was discussed in the works of [36].

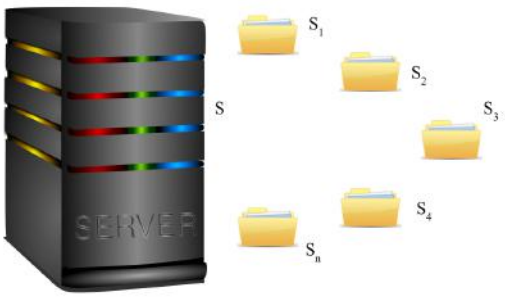


Figure 1.4: File storage system

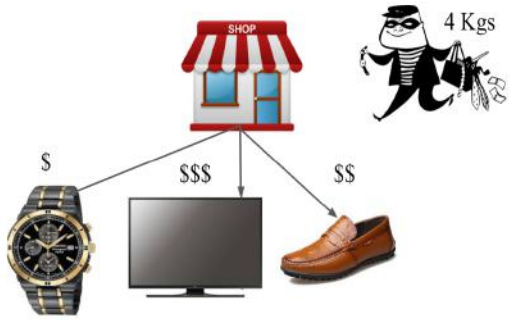


Figure 1.5: Shop robbery

1.2.1.3 Traveling Salesman Problem (TSP)

The Travelling Salesman Problem or TSP as it is more commonly known was initially designed to solve the problems for salesman who had to visit door-to-door to advertise and sell their products. Given, a set of cities (nodes) and the pairwise distances between them, the problem here is to find the shortest possible path visiting each city exactly once and terminating at the start location. The readers are hereby cautioned that this problem is slightly different from the Hamiltonian path planning problem [37] wherein the problem is to find if there exists a path to visit all nodes exactly once, whereas, in TSP, it is known that several Hamiltonian tours already exist. The objective, alternatively stated, is to find the minimum weighted Hamiltonian tour. People have gone ahead and used TSP for several interesting problems like Pokémon Go [38] as shown in Fig. 1.6 and planning roadtrips across USA whilst visiting tourist sites of interest [39] as shown in Fig. 1.7. Just like the previous categories, multiple TSP problem have already been exhaustively researched in works like [40–42] where researchers have also addressed the increased computational complexity and proprosed polynomial time approximations.

Figure 1.6: Pokémon Go TSP [38]

Figure 1.7: USA road trip TSP [39]

1.2.1.4 Intelligent Environment Monitoring (IEM)

When it comes to OP, while the spatial correlations and temporal variations across the nodes have been addressed but the overall spatially varying density across all the nodes cannot be modeled. Additionally, TSP tries to plan a tour across all available nodes but the limited resources available with the robot makes this approach infeasible. When it comes to planning informative trajectories while selecting the locations (nodes) to observe, some prior has been done by researchers in [43–46] wherein the objective is to reduce the uncertainty in the model being used to emulate the phenomenon of interest whilst relying on submodular functions[1]. As opposed to these, works like [47] rely on non-submodular functions and focus the robot exploration only in the areas exhibiting extreme values. In [48], the authors select the subset of nodes to be observed and then use multiple TSP approach to obtain the most informative paths for multiple UAVs operating in presence of wind disturbances. Additionally, each robot is expected to return back to its own start location so this work is an instance of Multiple Depot Multiple TSP (MDMTSP) problem and can be considered as a cross between TSP and IEM aspects.

1.2.1.5 Synopsis about Path Planning approaches

All the aforementioned works are closely related to the problem discussed in the scope of this work in the sense that node selection with reward maximization under budget constraints will be one of the key areas that this work contributes to. Not only this, but this work additionally tries to generate accurate models of the underlying dynamics of the target phenomenon as the observations are being accrued. Only the models discussed in the Section 1.2.1.4 could simultaneously achieve the two tasks. However, the objective of this work is to contribute to the cost functions allowing for overall reduction in uncertainty over the entire domain as opposed to just the area exhibiting extrema, thereby making this contribution novel and important.

[1]Submodular functions are non decreasing functions that follow the property of diminishing returns. This means that the enhancement over the function gradually decays as more data is acquired.

1.2.2 Data Compression and Resource Optimization

When it comes to robot resource constraints, not only are there power source constraints which limit the mission time / path length *etc.,* but also includes the memory capacity available. Currently, the robots have limited memory and processing capacity which entails that only a handful of data can be stored on-board thus, it is essential to look into some forms of data compression techniques. Such works are extensively found in Signal processing and Information theory literature and are summarized under.

1.2.2.1 The CEO Problem

Since the author is interested in carrying out decentralized robot team exploration, a very closely related work from the Information Theory domain that pertains to decentralized sensing architecture is discussed here. The problem being described is called the *Centralized Estimation Officer (CEO)* problem [49] wherein the CEO or sometimes also referred to as the Fusion Center (FC) is interested in estimating the state of a system which cannot be observed directly by the CEO. For this, several decentralized agents are deployed in a disconnected fashion (agents are not allowed to communicate with each other) who gather the necessary information and pass on their respective decisions to the CEO who then needs to assimilate all acquired decision into one global decision. This architecture is applicable to both detection problem wherein the presence or absence of a signal is being addressed or to an estimation problem as being discussed here. The overall architecture for a multi-robot setup is shown in Fig. 1.8 wherein each robot acquires subset of information from the target phenomenon and their is a possibility of correlations between multiple robots owing to the disconnected nature of the team. The agents do not communicate with each other and simply compress and send off the data to the FC. The data is not transmitted in its raw form (compression is involved prior to send off) and the transmission is held off until the agent has finished its exploration/ mission where the termination is determined by a user-defined condition like low battery. The signals/measurements that will be discussed in the scope of this work can be effectively modeled using multivariate Gaussian distributions, it is rather sensible to mention here that the generalized variant of the CEO problem *i.e.,* Gaussion-CEO problem [50] address this aspect.

The channel for communication which is used by the agents to transmit their respective decisions to the CEO has a limited bandwidth owing to which the data must undergo some form of compression to meet the bandwidth requirements. In doing so, distortion is introduced and the received data when decoded, does not exactly match the transmitted data. The objective then is to minimize this distortion given the rate limitations of the channel. In literature, this is called the rate-distortion paradox and there are theoretical bounds like the Berger-Tung bounds that explain the theoretical limits of the achievable distortion at the given rate. The generalized conjecture however, is known to be inapplicable for more than two agents [51].

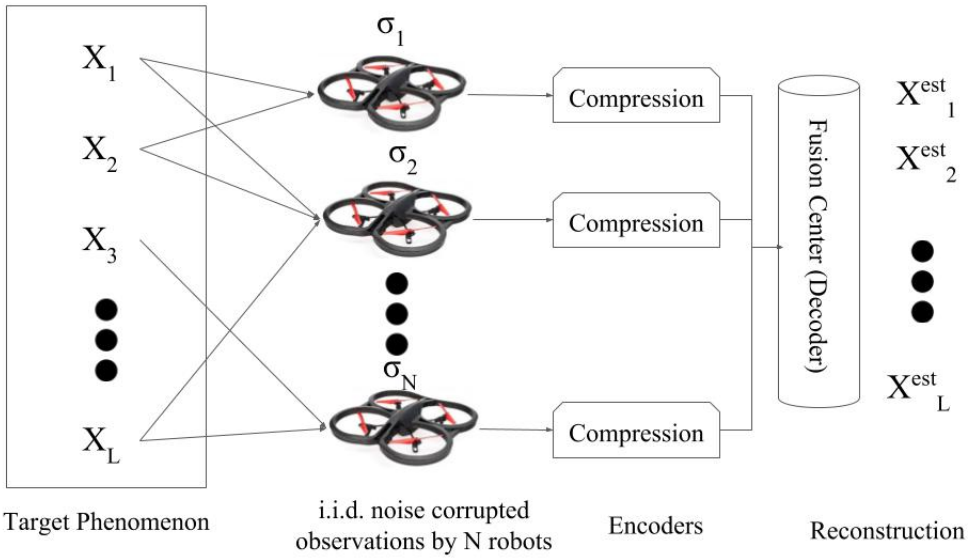


Figure 1.8: The CEO Problem. Several decentralized agents gather correlated observations from the phenomenon of interest and compress and pass along the "encoded" information to the centralized node hereby referred to as the CEO or Fusion Center (FC). The task of the CEO is then to "decode" the compressed information to reconstruct the original sequence of observations whilst minimizing the reconstruction error.

1.2.2.2 Compressed Sensing

Another optimization technique very commonly found in the Signal Processing literature is the Compressed Sensing [52] or invariably also known as the Compressive Sensing approach. In a nutshell, this refers to the problem of obtaining super-resolved signals using limited number of sensors in a non-adaptive fashion *i.e.,* signals are not converted into meaningful abstractions rather are directly acquired and numerically optimized to be transmitted. The motivation behind the numerical optimization lies in the redundant information that is picked up while acquiring signals. All such information can be filtered out and the input signal can instead be represented by a $K-$ sparse representation which refers to a low dimensional variant of the signal such that only K entries are non-zero. This is illustrated in Fig. 1.9. An extension to the distributed setting where multiple sources each gather individual noise corrupted signals is called Distributed Compressed Sensing (DCS) and further details are available in [53].

The important fact to note in this setup, both, for single and distributed sensing nodes, is that, the problem considers an under-determined linear system

1.2.2.3 Synopsis of Data Compression Techniques

On the very first glance it might appear that the CEO problem or the Compressed sensing problem directly address the core research problem of this work *i.e.,* resource constraint information acquisition by a decentralized and disconnected team which is then processed to make meaningful models which better explain the phenomenon of

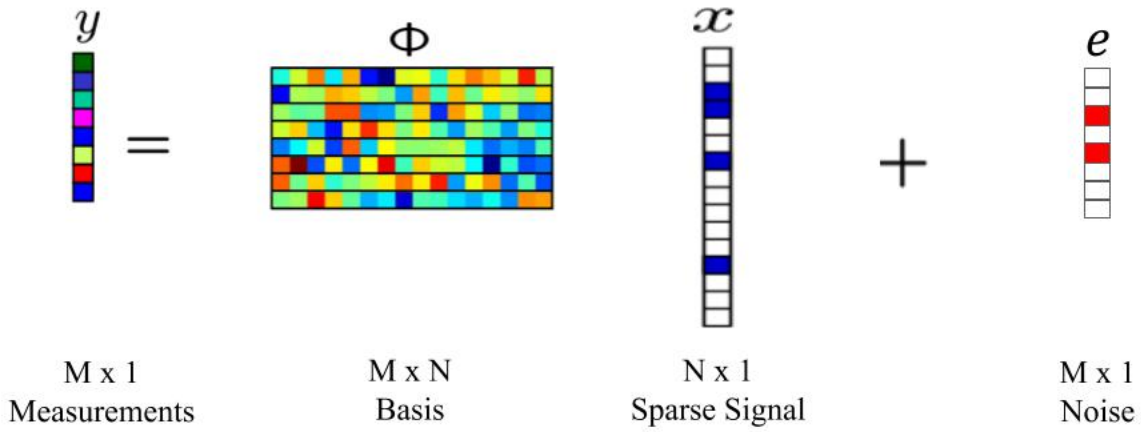


Figure 1.9: Compressed Sensing. With the apt choice of basis functions ϕ (alternatively known as the sensing matrix [54]), a $M \times 1$ measurement vector $\mathbf{y}$ can be represented using a K-sparse x i.e., a sparse representation of the signal with only K non-zero entries where K is comparatively small. Since most real-life systems are noisy, a noise corrupted variant of the architecture is addressed by considering the noise vector $\mathbf{e}$. The objective then is to reconstruct the original $\mathbf{y}$ with minimal reconstruction error and maximal compression.

interest. However, the author would like to caution the readers that these techniques address the non-adaptive sensing variant whereby the information acquired is passively compressed and sent across to the CEO or receipiant of interest via a communication channel. As opposed to this, the problem that the authors addresses lies in the domain of adaptive sampling with intermediate information processing that entails that the information acquired is not directly transmitted to the CEO in its raw form. Thus, the existing work do not directly solve the problem at hand and the author attempts to address such limitations of existing works.

1.3 Problem Description

This thesis serves to solve the following research problem:

> *Given limited resources, where should the robot (or a team thereof) sense the environmental phenomenon, in order to ensure good model quality and a safe return path to base station ?*

The overall problem can be broken down into several sub-problems:

Problem (a) *Where should the robot sense when operating on limited resources?*

Problem (b) *How many robots are required?*

Problem (c) *How to select the initial configuration (start locations) of the team?*

Problem (d) *What communication protocol should be used?*

Problem (e) *Will there be conflicting models? If so, how to resolve them?*

Problem (f) *How to ensure that the robots return to base and do not get stranded amidst the field?*

Problem (g) *Does the homing constraint compromise the model quality?*

Problem (h) *When should the mission be terminated?*

The author proposes the following solutions to the sub-problems (also shown in Fig. 1.10) discussed above:

Solution to Problem (a) Given the limited amount of resources (battery, payload, flight time, traversal time *etc.,*) the robot cannot perform blanket coverage of the vast sensing area that needs to be mapped and monitored. Instead, some of the most crucial locations must be chosen that are critical to be observed to allow the model to be generated accurately. For a detailed explanation and author's contribution towards this aspect, the readers are referred to Chapter 4.

Figure 1.10: A jigsaw of all sub-problems. All sub-problems explicitly addressed in this work are marked with !!! while those unmarked were solved using heuristic approaches.

Solution to Problem (b) Since the area of interest is large, it is usually the case that one robot is not sufficient. Now, as far as the number of additional robots are concerned, this largely depends on the supervisor and the project budget. Usually, a pre-determined size of robot team is given the task of exploration on a single battery discharge cycle and the agents are not added or removed from the team on the fly.

Solution to Problem (c) The area being monitored is largely unknown. The only known information about the area is the spatial expanse of the sensing domain. Thus, choosing the initial configuration again is a challenge in itself. It could either be solved based on some prior experience of the supervisor or the best bet is to use random initial configuration and to show the robustness of the model to such selections. For this work, radomized intial configurations were used and average performance over several trials was reported.

Solution to Problem (d) As was discussed earlier, when utilizing a team of mobile robots, there are several possibilities of establishing communication links between peers and peer-to-base. However, communication channels are faulty and are beset with packet loss, signal interference, attenuation, latency and a lot more related problems. Either an asynchronous sporadic communication protocol can be used where the robots are known to drop data packets and communicate sporadically with peers and base or an even more flexible solution of fully decentralized team can be adopted. While the communication channels can help to coordinate the team and may even further enhance the overall performance of the team, it is beneficial to develop models for harsh environments wherein communication is infeasible and to evaluate the worst-case performance.

Solution to Problem (e) Using a fully decentralized team *i.e.,* every robot is a master of its own will, has a slight drawback. Every robot will be observing parts of the domain which it seems essential for enhancing its own model. Based on what the robot has seen, its understanding about the rest of the unseen parts of the

environment would slightly differ and in fact may conflict with that of its peers. This can be resolved by carefully evaluating the confidence that a robot has on the map that it generated and fusing it with that of its peers by weighting their respective maps using their confidence values. Further details are explained in Chapter 6.

Solution to Problem (f) In one word, this problem can be called as "homing" *i.e.,* to ensure that the robot always reserves sufficient resources to return back to the base station [2]. Most of the cases where the robots are deployed, the data is gathered and stored on the robot itself owing to limited or faulty communication channels. Thus, loosing a robot amidst its mission owing to complete immobilization due to resource exhaustion is also a form of resource wastage. Additionally, loosing an agent also means loosing all the information that it gathered. To avoid this, homing should be considered as an additional constraint on top of the path planner being used to observe the environment. A detailed explanation of the procedure can be found in Chapter 4.

Solution to Problem (g) Owing to the additional homing constraints, it is but natural that the robot will not have the freedom to always observe the most informative locations. This ever lasting conflicting between *where to sense?* and *can the robot return to base from there?* slightly compromises the model performance and it must be empirically validated that this impact is not extreme. This aspect was analyzed in Chapter 4.

Solution to Problem (h) The robot should always maintain an estimate of how far it can reach given the available resources. When the resources are critically low or when their is a possibility for "homing" to fail, it is advisable that the mission be terminated immediately. To this end, the author makes contributions via Chapter 4 and Chapter 5.

1.3.1 Postulates

The problem to be tackled is vast (involving 8 sub-problems as previously discussed) and covers multiple domains of robotics and machine learning. It is beset with a lot of hardware and software limitations and utilizes a lot of man hours to be realized completely. Under the constraints faced by the author, this works proposes the aforementioned solutions in the light of following postulates:

- The environment in which the robots are being deployed is considered free of obstacles. This is usually the case, for instance, when deploying UAVs for monitoring pollution levels or AUVs for monitoring algal blooms *etc.,* which relate closely to the data set that will be used for empirical validations.

[2] could either be the robot's start location or any other pre-determined location which is known a priori to the robot.

- Robot localization is assumed to be perfect. Since the environment models, *i.e.,* Gaussian Process (GP), that will be used to model the environment largely rely on data association *i.e.,* tagging a measurements with its corresponding location which in Geostatistics is also known as geo tagging. Any uncertainty in data association directly affects the model performance. There are already solutions in the literature which solve the issue of uncertainty in localization but this will be addressed in the next phase of development.

- Robots are assumed to operate in communication devoid *(disconnected)* environments. The reason for such a drastic setting was that for applications like underwater surveillance, the AUVs need to resurface every time they intend to transmit data to the base. Resurfacing and diving back to the depth of interest, along with transmission via satellite based communication protocols are not resource friendly. As for the other settings like aerial or ground surveillance to monitor radiation leaks or pollution levels, if the robots are not required to be within communication range of each other (or the base station) they can spread out and cover large ground.

- Rogue agents are not considered in this work. During exploration, owing to sensor or actuation failures, some agents may be tagged as rogue agents. But deducing which agent has gone rogue and when, is a rather challenging task in itself and digresses from the focus of this work.

- Robots are assumed to have sufficient computational power so much so that, inference and predictions can be calculated in real-time or as per the requirements of the application.

- Each robot is a self-sustaining GP expert capable of selecting the locations that are deemed necessary to be observed and simultaneously updating the model. Multi-robot teams thus require multiple GPs. A discrete domain setting is considered.

- The nature of the environment being sensed and the spatial expanse of the sensing area are made known to the team as common knowledge before the mission unfolds.

- The time it takes for the each member of the team to complete its mission is comparatively smaller than the time it takes for the dynamics of the environment to evolve. Thus, only spatial domain modeling will be considered in the scope of this work.

- Size of the team remains constant throughout the mission. It is assumed that robots do not fail amidst the mission and no agents are added or removed from the team on the fly. Such hardware failure also falls under the category of rogue agents and can be handled by dynamically swapping the rogue agents with new ones or re-distribution of the task over the remainder of the team but this is beyond the current scope.

- As far as the starting configuration of the team is concerned, it is arbitrarily chosen by the supervisor. As for the base-station location, it can either be chosen as the start-location of the robot itself or any other location as per the supervisors whims. The base station location is also known as common knowledge to the entire team and for this work was chosen as a location different from the start- location of all agents. The reason for such a setting was that in real experiments, for instance, in case of nuclear fallout monitoring, the robots start very close to the contaminated area but the base-station and is set up at a safe distance to avoid the supervisors from picking up contamination.

- A point-sensing approach is considered which refers to the fact that the measurements are available at the robot location only.

- When planning trajectories via way-points, the pose of the robot is not restricted. It is believed that the robot follows the orientations which consume minimal energy and are within the non-holonomic constraints of the robot. As such, only the goal position needs to be attained and it is believed that the robot can reach this position deterministically.

1.3.2 Methodology

In Section 1.3, the author discussed several problems and proposed solutions to cater to them. In this section, within the boundaries of the postulates highlighted, the author now presents the overall system architecture as is illustrated in Fig. 1.11: In this figure, the target phenomenon (green block) is being modeled by a team of M robots each represented by a light blue block. There is no restriction on the nature of the robot or the team, *i.e.,* fully heterogeneous teams are also applicable. The intrinsic components of a robot are explained in detail below:

- **Map Phase:** This phase encompasses all robots individually generating the model of underlying dynamics. For this, the following blocks play a key role:

  - *Pre-processing:* In this step, recursive Bayesian denoising filters like Kalman filters [55] or digital signal processing techniques [56] are utilized to process the measurements before they are used for modeling. For the scope of this work, this problem was addressed by modeling observation noise alongside the environment dynamics.

  - *Localization:* Not only for the robot navigation but also for the models that will be deployed to infer the environment dynamics, it is essential that the robot can accurately decipher its own location. For this, signal strength based localization techniques like [57] can be utilized and were assumed to be solved. The required precision in localization is a function of the domain under consideration and was assumed to be satisfactory for the purpose of this work.

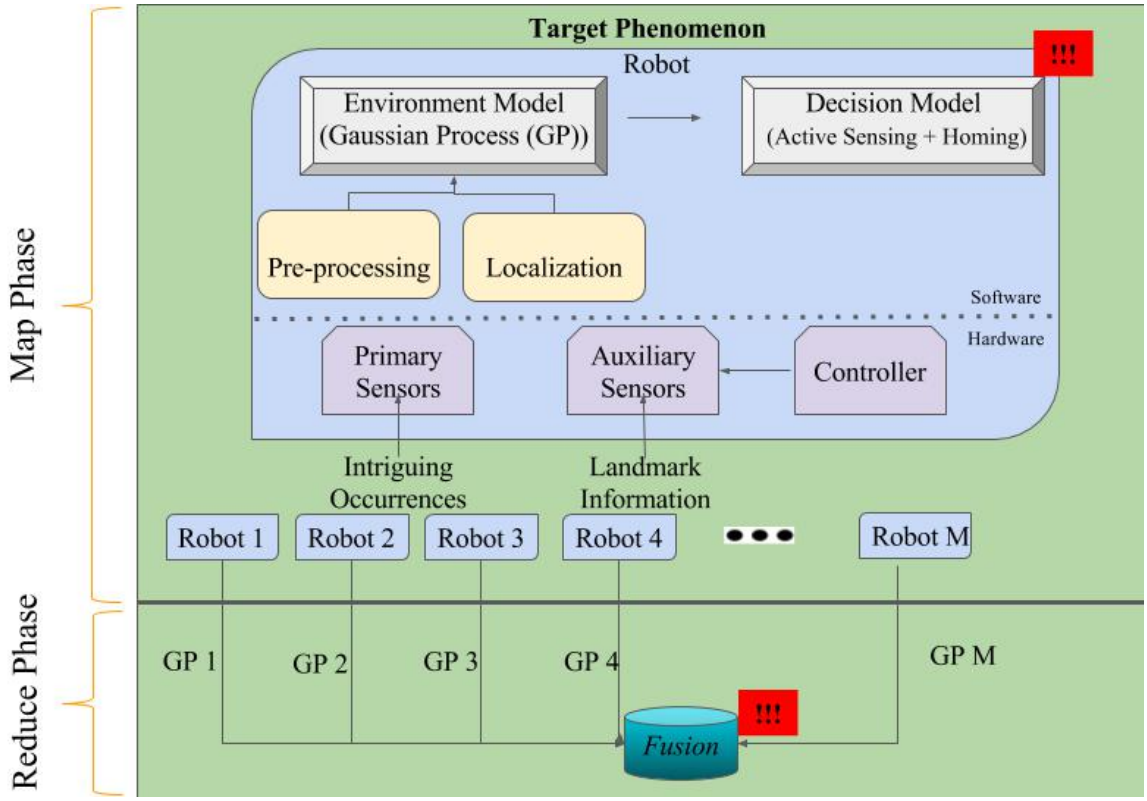


Figure 1.11: Block Diagram. This diagram shows the overview of the overall architecture which the author proposes to solve the environment monitoring problem. The architecture is split into ***Map phase*** *and* ***Reduce phase****, The former involves multiple robots generating the models of the environment utilizing the harmony between several hardware and software components while the latter encompasses the fusion of all models into a globally consistent model. The novel contributions made by the author are represented by !!!.*

  - *Environment Model:* Uses the location information fed by the localization module along with pre-processed sensor information to develop posterior probabilistic distributions of the environment dynamics using Non-parametric Bayesian methods called Gaussian Process (GP).
  - *Decision Model:* Assists the environment model by allowing the robot to effectively chose the best of sampling locations (also known as active sensing) from the entire domain wherein exhaustive coverage is rendered infeasible. Additionally, homing constraints *i.e.,* requirements for the robots to return to base station at the end of their respective missions, are imposed.
  - *Hardware Components:* All the aforementioned components form the software meta-layer which commands the robot actuators to take actions and gather information. For this, primary sensors *i.e.,* the sensors required to gather environment specific information and auxiliary sensors *i.e.,* additional sensors like lidar, sonar, camera *etc.,* for obstacle avoidance and safe navigation are utilized.

- **Reduce Phase:** In this phase, all robots feed their inferred models to a base station with sufficient computational power to fuse all models into one globally consistent model. This phase is done in one-pass, only after all robots have completed their missions and returned to base. This phase relies on just one component:

  - *Fusion:* This block represents the base station which fuses all the information from all models into one globally consistent model. This is well suited for humans and further details can be found in Chapter 6.

1.4 Contributions

Through this work, the author makes the following contributions:

1. **Formulation of the resource constrained information maximization.** To account for the robot resources while maximizing the information acquired, this work proposed a novel bi-objective function which is a weighted combination of entropy (information acquired) and resources used. In order to adapt this to the intelligent environment monitoring setting like those discussed in the scope of this work, the author proposed a novel approach which utilizes the available resources to deduce the weight of objective functions. The objective functions in consideration here are: 1.) performance of environment model (GP) and 2.) thrifty utilization of resources without compromising on model performance. Since there are only two objectives being considered for this work, this can be considered as a bi-objective optimization problem wherein the model performance and resource utilization are combined into a single objective function using these weights, which is then optimized to choose the optimal *next-best-location*. Here, the *next-best-location* refers to the critical or "informative" locations which a

robot can visit/observe given the resource constraints. Since a robot cannot perform blanket coverage of the entire domain, optimization over the said cost function allows to choose the most crucial subset of locations that can and should be observed.

2. **Homing guarantees while choosing the next-best-location.** The author further enhances the bi-objective optimization framework as a constrained bi-objective optimization approach where the "homing" requirements are posed as additional constraints atop the cost function being optimized. In doing so, we now ensure that the robot can choose to terminate its mission autonomously when the residual resources are critically low and choose to return to base. This not only avoids loosing the robots amidst the mission but also maximizes the number of observations gathered from the environment. Since the model being used is highly data driven, the higher is the variance and amount of data gathered, the better is the model performance.

3. **Estimating operational range to bound the maximum path length.** When the robot ventures out in the field, it sets out with a fully charged battery and it must keep track of the residual battery in order to avoid complete immobilization amidst the field. For this, the robot must bound its total path length by the maximum operational range it can achieve on a single discharge cycle. However, as the mission progresses and based on the maneuvers that the robot had to perform, this estimate needs to be updated in real time. For this, a range estimation framework is proposed which is then generalized to estimate the range for various classes of robots in *offline* and/or *online* fashions.

4. **Efficient model fusion for fully decentralized robot team.** Most of the time the environment being monitored is so vast that a single robot proves insufficient. To overcome this, multiple robots can be deployed but having a fully centralized framework incurs large communication overhead and constraints the robots to be within communication range of each other. To overcome this, the author uses a communication devoid setup which is well suited to rather harsh conditions. In doing so, each robot generates its own individual model and selects the *next-best-location* which it deems best fit for itself, irrespective of the others in the team. Finally, at the end of missions of all robots, the single globally consistent model must be deduced for which a weighted fusion of all individual models is discussed. In this technique, the prediction performance of each model is carefully weighted so as to ensure that no model is completely ignored (loss of information) or no model is blindly trusted (faulty information).

5. **Theoretical evaluation of resource constrained optimization using real-world datasets.** The algorithms proposed in this thesis for performing bi-objective optimization with homing guarantees and model fusion have been theoretically evaluated on real-world environment monitoring datasets using simulated robots. The robots were considered as point masses and their motion was considered as deterministic.

6. **Practical evaluation of operational range estimation in real world environments for real robots.** The algorithms proposed for operational range estimation have been exhaustively verified on real robot platforms viz., UGV and Multi-rotor UAV in real outdoor conditions.

1.5 Outline

This work can be split into 5 parts and the details of each parts and enclosed components are split into:

- **Part-I: Foundation**
  - **Chapter 1: Introduction** presents the motivation for this work, problem definition and proposed solutions.
  - **Chapter 2: Preliminaries** is meant to serve as a primer for the readers to familiarize them with the basic tools required to ease the understanding of this work.
- **Part-II: Map Phase**
  - **Chapter 3 : Modeling the Spatial Variations of the Environment using Stationary Homoscedastic GPs** details the choice of kernels used to infer the underlying dynamics of large-scale environmental phenomenon by considering only the spatial domain variations.
  - **Chapter 4: Resource Constrained Path Planning with Homing Guarantee** presents information acquisition functions that allow the robots to gather the highest quality observations which are within operational limits. Here, the resource-constrained variant of the information maximization problem is addressed with conflicting objectives (model quality v/s resource utilization) and optimal weights are deduced to fuse them into one global cost function to be optimized. The output of this acquisition function is the *next-best-location* which is beneficial for the robot to observe.
  - **Chapter 5: Operational Range Estimation** solves a critical problem of placing the upper bound over the path length that can be traversed by the robot while gathering observations to enhance its model. Both *offline* and *online* variants are presented to either use prior information in case of the former or to use real-time performance data in case of the latter. This serves as the true representation of the budget under which the robot would operate in a real scenario.
- **Part-III: Reduce Phase**

  - **Chapter 6: Fusion of Distributed Gaussian Process Experts *(FuDGE)*** discusses a novel technique to fuse multiple locally generated models into one globally consistent model by taking into account the confidence of each model in making predictions. This is a point-wise fusion approach and is done as one-pass by the base stations at the end of missions of all robots.

- **Part-IV: Spatiotemporal Modeling**

  - **Chapter 7 : Towards a Spatiotemporal Environment Monitoring for Continuous Domains** details the properties and choice of kernels used to infer the underlying dynamics of large-scale environmental phenomenon by considering both the spatial and temporal domain variations.

- **Part V: Epilogue**

  - **Chapter 8: Conclusion and Future Works** summarizes the contributions of this works and discusses the possible extensions that can be implemented to further enhance the framework and push it closer to being deployed on robots operating in real-world scenarios.

The interplay of each part and enclosed chapters is visually explained by Fig. 1.12.

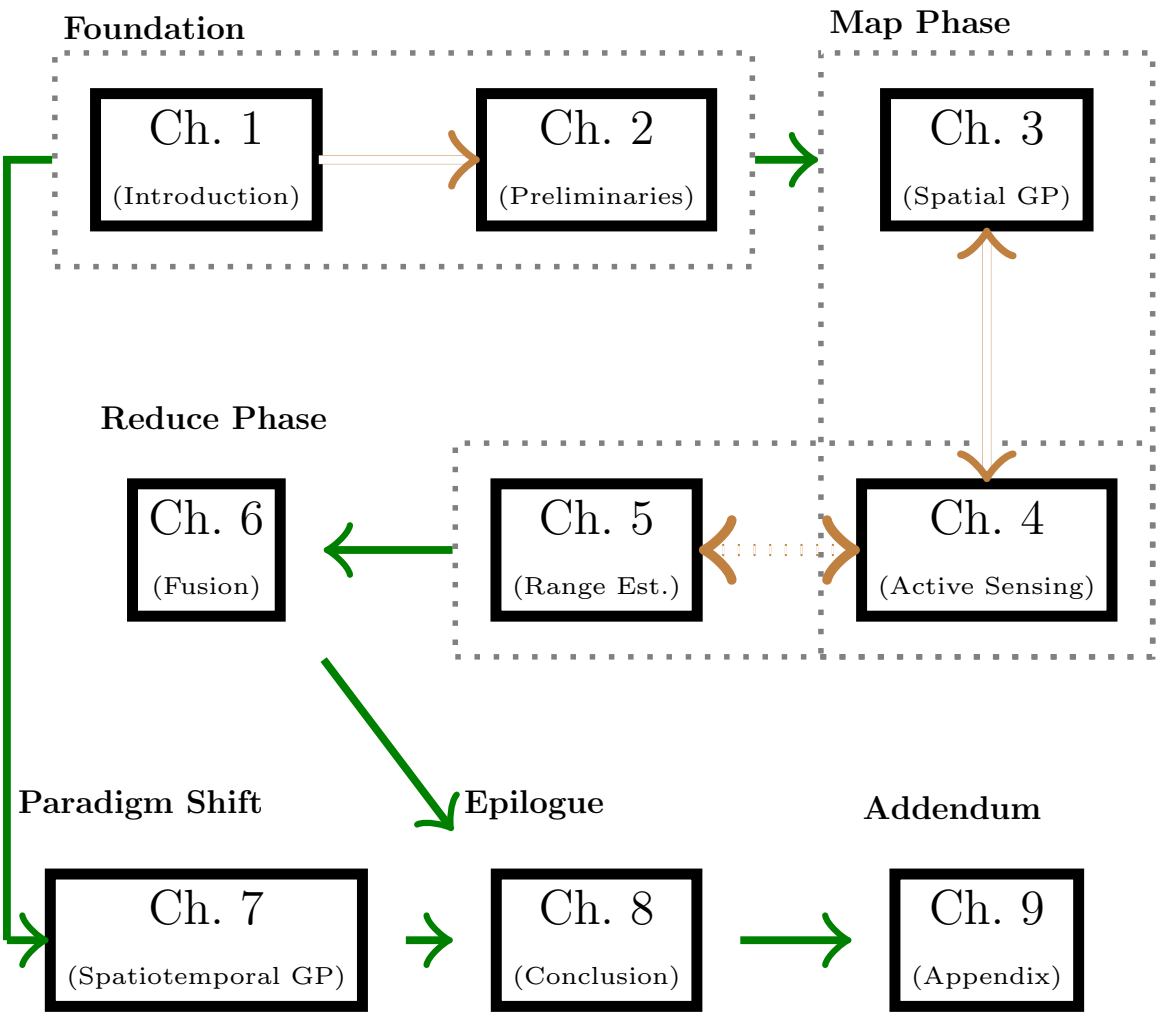


Figure 1.12: Chapter dependencies. This diagram shows the synergy between chapters of this work. This can be considered as a road-map and is useful to understand the outline of this work. All parts of the thesis are highlighted with corresponding blocks and all chapter connections are shown with $\rightarrow$*. The double sided* $\Longleftrightarrow$ *represents the intrinsic connection between the components of the* ***Map Phase****. The dashed arrow* $\Longleftrightarrow$ *represents weak dependance between Chapter 4 and Chapter 5 since the works are yet to be integrated.*

Bibliography

[1] C. Pirri, M. Cocuzza, L. Scaltrito, S. Ferrero, P. Rivolo, *et al.*, "Evolved sensing platform for the offshore sites sea water environmental monitoring," in *Offshore Mediterranean Conference and Exhibition*, Offshore Mediterranean Conference, 2017.

[2] M. Tavakol, R. Arjmandi, M. Shayeghi, S. M. Monavari, and A. Karbassi, "Developing an environmental water quality monitoring program for haraz river in northern iran," *Environmental Monitoring and Assessment*, vol. 189, no. 8, p. 410, 2017.

[3] F. Weiss, H. Wey, C. Stamm, C. Ruepert, C. Zurbrügg, and R. Eggen, "Passive sampling approaches used for time-integrated environmental monitoring and risk assessment in the tropical río tapezco catchment in costa rica," in *EGU General Assembly Conference Abstracts*, vol. 19, p. 13660, 2017.

[4] H. K. White, P.-Y. Hsing, W. Cho, T. M. Shank, E. E. Cordes, A. M. Quattrini, R. K. Nelson, R. Camilli, A. W. Demopoulos, C. R. German, *et al.*, "Impact of the deepwater horizon oil spill on a deep-water coral community in the gulf of mexico," *Proceedings of the National Academy of Sciences*, vol. 109, no. 50, pp. 20303–20308, 2012.

[5] J. T. James, T. F. Limero, S. W. Beck, M. Martin, P. A. Covington, L. Yang, D. Lind, and J. F. Boyd, "Environmental monitoring air quality," *Helen W. Lane*, p. 177, 2002.

[6] N. Domínguez-Morueco, S. Augusto, L. Trabalón, E. Pocurull, F. Borrull, M. Schuhmacher, J. L. Domingo, and M. Nadal, "Monitoring pahs in the petrochemical area of tarragona county, spain: comparing passive air samplers with lichen transplants," *Environmental Science and Pollution Research*, vol. 24, no. 13, pp. 11890–11900, 2017.

[7] D. Mage, G. Ozolins, P. Peterson, A. Webster, R. Orthofer, V. Vandeweerd, and M. Gwynne, "Urban air pollution in megacities of the world," *Atmospheric Environment*, vol. 30, no. 5, pp. 681–686, 1996.

[8] I. Krupnik and D. Jolly, *The Earth Is Faster Now: Indigenous Observations of Arctic Environmental Change. Frontiers in Polar Social Science.* ERIC, 2002.

[9] I. P. on Climate Change, *Climate Change 2014–Impacts, Adaptation and Vulnerability: Regional Aspects*. Cambridge University Press, 2014.

[10] G. T. Pecl, M. B. Araújo, J. D. Bell, J. Blanchard, T. C. Bonebrake, I.-C. Chen, T. D. Clark, R. K. Colwell, F. Danielsen, B. Evengård, *et al.*, "Biodiversity redistribution under climate change: Impacts on ecosystems and human well-being," *Science*, vol. 355, no. 6332, p. eaai9214, 2017.

[11] I. Dincer, "Renewable energy and sustainable development: a crucial review," *Renewable and Sustainable Energy Reviews*, vol. 4, no. 2, pp. 157–175, 2000.

[12] B. Bayat, N. Crasta, A. Crespi, A. M. Pascoal, and A. Ijspeert, "Environmental monitoring using autonomous vehicles: a survey of recent searching techniques," *Current Opinion in Biotechnology*, vol. 45, pp. 76–84, 2017.

[13] M. Egerstedt, "Persistent environmental monitoring: Robots that seemingly do nothing most of the time," 2017.

[14] M. Dunbabin and L. Marques, "Robots for environmental monitoring: Significant advancements and applications," *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 24–39, 2012.

[15] H. Messer, A. Zinevich, and P. Alpert, "Environmental monitoring by wireless communication networks," *Science*, vol. 312, no. 5774, pp. 713–713, 2006.

[16] M. Trincavelli, M. Reggente, S. Coradeschi, A. Loutfi, H. Ishida, and A. J. Lilienthal, "Towards environmental monitoring with mobile robots," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pp. 2210–2215, IEEE, 2008.

[17] M. Vidyasagar, *Nonlinear systems analysis*. SIAM, 2002.

[18] L. Arnold, "Stochastic differential equations," *New York*, 1974.

[19] J. Binney, A. Krause, and G. S. Sukhatme, "Informative path planning for an autonomous underwater vehicle," in *Robotics and automation (icra), 2010 IEEE international conference on*, pp. 4791–4796, IEEE, 2010.

[20] R. Murphy, *Introduction to AI robotics*. MIT press, 2000.

[21] S. McNEIL, "Growing algae bloom in Arabian Sea tied to climate change." `https://tinyurl.com/l3et3oy`, 2017. [Online; accessed 6-Dec-2017].

[22] N. Seltenrich, "Keeping tabs on habs: new tools for detecting, monitoring, and preventing harmful algal blooms," *Environmental health perspectives*, vol. 122, no. 8, p. A206, 2014.

[23] S. Arismendez, "For Submerged Oil Pollution in Western Gulf of Mexico, Restoration Is Coming After 2005 DBL 152 Oil Spill." `https://tinyurl.com/y8jzuto7`, 2013. [Online; accessed 6-Dec-2017].

[24] N. Oceanic and A. A. (NOAA), "New Seaglider Collects Data along Gulf Coast." `https://tinyurl.com/y7s9dt98`, 2012. [Online; accessed 6-Dec-2017].

[25] S. D. of Agriculture. `https://tinyurl.com/y922wnel`. [Online; accessed 6-Dec-2017].

[26] S. Koperski, "Experimental drone used for controlled burn." https://tinyurl.com/y72bulxw, 2016. [Online; accessed 6-Dec-2017].

[27] N. THABIT and J. BERTUCCI, "AWAKENING FROM THE DREAM: THE REAL FACTS ON FUKUSHIMA." https://tinyurl.com/y7qkr4f4, 2012. [Online; accessed 6-Dec-2017].

[28] I. R. I. for Nuclear Decommissioning (IRID), "Robots working inside the buildings at Fukushima Daiichi NPS (Part II) MHI-MEISTeR - Images taken inside the Unit 2 Reactor building by the robot -AWAKENING FROM THE DREAM: THE REAL FACTS ON FUKUSHIMA." http://irid.or.jp/en/research/meister/. [Online; accessed 6-Dec-2017].

[29] I.-M. Chao, B. L. Golden, and E. A. Wasil, "A fast and effective heuristic for the orienteering problem," *European journal of operational research*, vol. 88, no. 3, pp. 475–489, 1996.

[30] A. Gunawan, H. C. Lau, and P. Vansteenwegen, "Orienteering problem: A survey of recent variants, solution approaches and applications," *European Journal of Operational Research*, vol. 255, no. 2, pp. 315–332, 2016.

[31] J. Yu, M. Schwager, and D. Rus, "Correlated orienteering problem and its application to persistent monitoring tasks," *IEEE Transactions on Robotics*, vol. 32, pp. 1106–1118, Oct 2016.

[32] Z. Ma, K. Yin, L. Liu, and G. S. Sukhatme, "A spatio-temporal representation for the orienteering problem with time-varying profits," *arXiv preprint arXiv:1611.08037*, 2016.

[33] P. Toth and D. Vigo, *The vehicle routing problem*. SIAM, 2002.

[34] L. Ke, C. Archetti, and Z. Feng, "Ants can solve the team orienteering problem," *Computers & Industrial Engineering*, vol. 54, no. 3, pp. 648–665, 2008.

[35] R. S. Garfinkel, G. L. Nemhauser, *et al.*, *Integer programming*, vol. 4. Wiley New York, 1972.

[36] C. Chekuri and S. Khanna, "A polynomial time approximation scheme for the multiple knapsack problem," *SIAM Journal on Computing*, vol. 35, no. 3, pp. 713–728, 2005.

[37] A. A. Bertossi, "The edge hamiltonian path problem is np-complete," *Information Processing Letters*, vol. 13, no. 4-5, pp. 157–159, 1981.

[38] "Pokémon Go and Traveling Salesman Problem." http://www.math.uwaterloo.ca/tsp/poke/index.html, 2016.

[39] R. S. Olson, "Computing the optimal road trip across the U.S.." http://www.randalolson.com/2015/03/08/computing-the-optimal-road-trip-across-the-u-s/, 2017.

[40] T. Bektas, "The multiple traveling salesman problem: an overview of formulations and solution procedures," *Omega*, vol. 34, no. 3, pp. 209–219, 2006.

[41] S. Trigui, O. Cheikhrouhou, A. Koubaa, U. Baroudi, and H. Youssef, "Fl-mtsp: a fuzzy logic approach to solve the multi-objective multiple traveling salesman problem for multi-robot systems," *Soft Computing*, vol. 21, no. 24, pp. 7351–7362, 2017.

[42] S. Ozden, A. E. Smith, and K. R. Gue, "Solving large batches of traveling salesman problems with parallel and distributed computing," *Computers & Operations Research*, vol. 85, pp. 87–96, 2017.

[43] D. Golovin and A. Krause, "Adaptive submodularity: Theory and applications in active learning and stochastic optimization," *Journal of Artificial Intelligence Research*, vol. 42, pp. 427–486, 2011.

[44] A. Krause and C. Guestrin, "Near-optimal observation selection using submodular functions," in *AAAI*, vol. 7, pp. 1650–1654, 2007.

[45] A. Singh, A. Krause, and W. J. Kaiser, "Nonmyopic adaptive informative path planning for multiple robots.," in *IJCAI*, vol. 3, p. 2, 2009.

[46] A. Singh, A. Krause, C. Guestrin, W. J. Kaiser, and M. A. Batalin, "Efficient planning of informative paths for multiple robots.," in *IJCAI*, vol. 7, pp. 2204–2211, 2007.

[47] R. Marchant and F. Ramos, "Bayesian optimisation for intelligent environmental monitoring," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 2242–2249, IEEE, 2012.

[48] D.-H. Cho, J.-S. Ha, S. Lee, S. Moon, and H.-L. Choi, "Informative path planning and mapping with multiple uavs in wind fields," *arXiv preprint arXiv:1610.01303*, 2016.

[49] T. Berger, Z. Zhang, and H. Viswanathan, "The ceo problem [multiterminal source coding]," *IEEE Transactions on Information Theory*, vol. 42, pp. 887–902, May 1996.

[50] Y. Yang and Z. Xiong, "On the generalized gaussian ceo problem," *IEEE Transactions on Information Theory*, vol. 58, no. 6, pp. 3350–3372, 2012.

[51] S. Jana and R. Blahut, "Berger-tung problem: A systematic approach based on canonical description," in *Information Theory and Its Applications, 2008. ISITA 2008. International Symposium on*, pp. 1–6, IEEE, 2008.

[52] D. L. Donoho, "Compressed sensing," *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[53] H. Yin, J. Li, Y. Chai, and S. X. Yang, "A survey on distributed compressed sensing: theory and applications," *Frontiers of Computer Science*, vol. 8, no. 6, pp. 893–904, 2014.

[54] T. T. Do, Y. Chen, D. T. Nguyen, N. Nguyen, L. Gan, and T. D. Tran, "Distributed compressed video sensing," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pp. 1393–1396, IEEE, 2009.

[55] M. S. Grewal, "Kalman filtering," in *International Encyclopedia of Statistical Science*, pp. 705–708, Springer, 2011.

[56] A. Antoniou, *Digital signal processing*. McGraw-Hill, 2016.

[57] J. Ko and D. Fox, "Gp-bayesfilters: Bayesian filtering using gaussian process prediction and observation models," *Autonomous Robots*, vol. 27, no. 1, pp. 75–90, 2009.

[58] K. Deb, "Multi-objective optimization," in *Search methodologies*, pp. 403–449, Springer, 2014.

Chapter 2

Preliminaries

> *Give me 6 hours to chop down a tree and I will spend the first 4 sharpening the axe.*
>
> Abraham Lincoln, *1960*

This chapter serves to accustom the readers with the basic tools necessary to ease the understanding of this thesis. Herein, the first part explains the choice of machine learning tools employed, second part details the path planning approaches deployed to gather training samples followed by the evaluation criteria used.

2.1 Density Estimation using Gaussian Process (GP)

This section features the probabilistic models that can be used to model highly non-linear environmental dynamics. Multivariate Gaussian distributions are useful for modeling finite collections of real-valued variables. However, to make them well suited to real world scenarios, they need to be generalized to an infinite-sized collection of real valued variables which is referred to as **Gaussian Process (GP)** [1]. This is further assisted by Example 2.1.1.

Example 2.1.1 (From Gaussian distribution to Gaussian Process (GP))**.** *Let there be* $N = 100$ *discrete locations which report measurements of interest e.g., temperature, pressure, humidity, pollution, precipitation, etc. Then, in order to be able to predict the measurement at any one (or subset) of these locations given the observed measurements from all other locations, a joint posterior distribution over all* 100 *locations must be obtained. This involves modeling* $n = 100$ *random variables in a* 100 D *input space. However, depending upon the spatial resolution of the target phenomenon,* $n \to \infty$ *very fast. Thus, a generalization of Gaussian distribution which can account for such large number of random variables in a very high dimensional space is called the GP. Any set* $n_s \subset n$ *would still result in a joint distribution which is Gaussian in nature.*

2.1.1 Description

In what follows, the author describes the GP model formally, followed by introducing the jargon used in the literature to describe the nature of the GP model. The author also enlists some of the well-known kernels which are required to define the GP models and details the inference and estimation procedures. Potential applications of GPs are discussed along with the strengths and weaknesses of this model. Additionally, the author justifies why the GP models were chosen despite their weakness and formally defines the scope of GP models as far as this work is concerned.

2.1.1.1 Formal Definition

In particular, the GPs can be thought of as distributions not just over random vectors but in fact distributions over random functions. It must however be pointed out here that from Example 2.1.1 the readers may perceive that GPs always deal with infinite dimensional spaces. But, it is not necessarily required to deal with the infinite-dimensional entity as it is. The reason being that for the marginal of a Gaussian, only the covariance of the block of the matrix involving the unmarginalized dimensions matters. Thus, for any finite collection of data points, it still forms a multivariate Gaussian and the inference in the GP will give the same answer even if infinitely many other points were ignored, as it will if all of them were to be accounted for.

Just like a multivariate Gaussian distribution, GPs are fully defined in terms of their mean function and covariance function denoted by $\boldsymbol{\mu}(\cdot), K(\cdot,\cdot)$ respectively. If a latent function f is known to vary as a realization of a GP *i.e.,* $f \sim GP(\boldsymbol{\mu}(\cdot), K(\cdot,\cdot))$ then, $\boldsymbol{\mu}(\cdot) = \mathbb{E}[f(\cdot)]$ and $K(\cdot,\cdot) = \mathbb{E}[(f(\cdot) - \mathbb{E}[f(\cdot)])^2]$. An an example to assist with handling these notation, the readers are referred to Example 2.1.2.

Example 2.1.2 (Handling GP notations)**.** *Consider a $d-$dimensional sensing domain $D \subset \mathbb{R}^d$ represented as a network of spatially correlated nodes or pre-determined locations. Then, for any pair $\{x, x'\} \in D$, their respective means are given by $\mu(x), \mu(x')$ while their covariance $\sigma_{xx'}$ given by $K(x, x')$.*

There are a variety of kernels that could be used for GP models and the kernels can be broadly classified based on their nature and purpose served. Such details are explained hereafter.

2.1.1.2 The Kernel Jargon

Various classes of kernel functions are known in the literature and are usually addressed with specific terms (jargon). For the ease of the readers, they are categorized and detailed below along with an intuitive notion of the purpose served by the kernel itself.

- **Kernels:** Kernels are a flexible way to represent the data so that it can be used to compare the samples in a complex space. A kernel is basically a non-linear non-parametric transformation function that maps the data to a higher dimensional space enabling linearity and easier comparisons of complex features. By doing this mapping, kernels enable the machine learning approaches to utilize

a linear model in the new input space to quantify similarity between a pair of objects x and x'. This is equivalent to regression by utilizing infinitely many Gaussian shaped basis functions placed everywhere and not just over the training points.

- **Stationary v/s Non-Stationary Kernels:** A translation invariant kernel is called stationary. This means that:

$$K(x, x') = K(x + \Delta, x' + \Delta) \tag{2.1}$$

From Eq. (2.1), it can be intuitively inferred that a stationary kernel is a kernel that is purely a function of $x - x'$ and not the actual value of x, thereby making it translation invariant. Furthermore, if the kernel depends only on the $||x - x'||$, i.e., it is rotation invariant, then the kernel is said to be an *isotropic* stationary kernel and conversely, if the separation in features is a function of direction then such a kernel is termed as *anisotropic* stationary kernel [2].

Often times in the real world, it is noted that the correlations do not vary uniformly throughout the entire domain. thus, generic kernels that can explain the underlying dynamics as a function of the input itself are required. This means that:

$$K(x, x') = \int_{\mathcal{R}^d} K_x(\cdot) K_{x'}(\cdot) d(\cdot) \tag{2.2}$$

The kernels of the form given by Eq. (2.2) are referred to as non-stationary kernels since the correlation varies with the input.

- **Separable v/s Non-Separable Kernels:** Usually when dealing with inputs represented by the spatiotemporal tuple as $[\mathbf{s} \in \mathcal{R}^2, t \in \mathcal{R}]$ a natural way to build kernels dealing with spatial and temporal domains is to multiply a spatial kernel with a temporal kernel as:

$$K((\mathbf{s}, \mathbf{s}'), (t, t')) = K_s(\mathbf{s}, \mathbf{s}') \cdot K_t(t, t') \tag{2.3}$$

However, if a kernel is not reducible into spatial and temporal domains as explained in Eq. (2.3), then works like [3] can be utilized to create non-separable non-stationary space-time kernels that even allow for distinct variations in either domain. To test for separability, Hilbert-Schmidt Independence Criterion (HSIC) [4] can be used.

- **Homoscedastic v/s Heteroscedastic Kernels:** Usually the observations being gathered for training are noisy owing to sensor modalities. To account for such factors, a noise term is additionally added to the covariance function. For any two inputs x, x', this can be done as:

$$K_\epsilon(x, x') = K(x, x') + \delta \sigma_n^2 \tag{2.4}$$

In Eq. (2.4), the symbol δ refers to the dirac-delta function which is used to check if $x = x'$ and σ_n represents the distribution of the noise. Hence, the nature

of noise is a function of the data being observed. If $\sigma_n \sim \mathcal{N}(0, \epsilon)$, then, a uniform noise is assumed for the entire domain and this is termed as *Homoscedasticity*. On the other hand, if $\sigma_n \sim \mathcal{N}(0, \epsilon(x))$, then, input-dependent noise model is obtained and is termed as *Heteroscedasticity*.

- **Hyper-parameters:**

  Also known as the free parameters, these hyper-parameters control the behavior of the GP. Intuitively, the characteristic length scales represent the distance one must move in input space before the function value can change *significantly*. Thus, short length-scales mean the error bars (i.e., predictive variance) can grow rapidly away from the data points whilst large length-scales imply irrelevant features (function value would be constant function of that feature input). When dealing with multi-dimensional input spaces, it is possible that the variation is not similar for each dimension and hence the method of *Automatic Relevance Determination (ARD)* can be deployed which has been so named since the model determines the "relevance" of length scales per dimension. Besides the characteristic length scales, for convenience, two additional parameters are also accounted for in the same category *viz.,* signal variance which defines the amplitude of variance in the signal being monitored and similarly noise variance. Together, the entire set of signal variance, characteristic length scales and noise variance are referred to as hyper-parameters for the scope of this work.

2.1.1.3 Variety of Kernels

The extensive literature on machine learning and kernel methods discusses several available covariance functions (kernels), some of which are summarized in Table 2.1 where $\mathbf{d} \triangleq (\mathbf{x} - \mathbf{x}')^T L^{-1} (\mathbf{x} - \mathbf{x}')$ and $\tilde{\mathbf{x}} \triangleq [1, \mathbf{x}]$.

Table 2.1: Some of the well established kernel functions

Name	**Description** $K(\mathbf{x}, \mathbf{x}')$	**Hyper-parameters**
Constant	σ_o^2	$\boldsymbol{\theta} \triangleq \{\sigma_o\}$
Linear	$\sigma_{sig}^2(\sigma_o^2 + \mathbf{x}^T L \mathbf{x}')$	$\boldsymbol{\theta} \triangleq \{\sigma_{sig}^2, \sigma_o, L\}$
Gaussian Noise	$\sigma_{sig}^2 \delta_{\mathbf{x},\mathbf{x}'}$	$\boldsymbol{\theta} \triangleq \{\sigma_{sig}^2\}$
Exponential	$\sigma_{sig}^2 \exp\big(-\sqrt{\mathbf{d}}\big)$	$\boldsymbol{\theta} \triangleq \{\sigma_{sig}, L\}$
$\gamma-$Exponential	$\sigma_{sig}^2 \exp\big(-\mathbf{d}^{\gamma/2}\big)$	$\boldsymbol{\theta} \triangleq \{\sigma_{sig}, L, \gamma\}$
Squared Exponential (RBF) with ARD	$\sigma_{sig}^2 \exp\big(-\frac{1}{2}(\mathbf{d})\big)$	$\boldsymbol{\theta} \triangleq \{\sigma_{sig}, L\}$
Squared Exponential (RBF) without ARD	$\sigma_{sig}^2 \exp\big(-\frac{1}{2l^2}(\mathbf{x} - \mathbf{x}')^T(\mathbf{x} - \mathbf{x}')\big)$	$\boldsymbol{\theta} \triangleq \{\sigma_{sig}, l\}$
Matern $\frac{1}{2}$	$\sigma_{sig}^2 \exp\big(-\mathbf{d}\big)$	$\boldsymbol{\theta} \triangleq \{\sigma_{sig}, L\}$
Matern $\frac{3}{2}$	$\sigma_{sig}^2(1 + \sqrt{3\mathbf{d}}) \exp(-\sqrt{3\mathbf{d}})$	$\boldsymbol{\theta} \triangleq \{\sigma_{sig}, L\}$
Matern $\frac{5}{2}$	$\sigma_{sig}^2(1 + \sqrt{5\mathbf{d}} + \frac{5\mathbf{d}}{3}) \exp(-\sqrt{5\mathbf{d}})$	$\boldsymbol{\theta} \triangleq \{\sigma_{sig}, L\}$
Rational Quadratic	$\sigma_{sig}^2\big(1 + \frac{\mathbf{d}}{2\alpha}\big)^{-\alpha}$	$\boldsymbol{\theta} \triangleq \{\sigma_{sig}, L, \alpha\}$
Polynomial	$\sigma_{sig}^2(\sigma_o^2 + \mathbf{x}^T L \mathbf{x}')^p$	$\boldsymbol{\theta} \triangleq \{\sigma_{sig}, \sigma_o, L, p\}$
Periodic Kernel	$\sigma_{sig}^2 \exp\big(\frac{2\sin^2(\pi T\sqrt{\mathbf{d}})}{\rho^2}\big)$	$\boldsymbol{\theta} \triangleq \{\sigma_{sig}, L, T, \rho\}$
Neural Network	$\frac{2}{\pi}\sin^{-1}\big(\frac{2\tilde{\mathbf{x}}^T \Sigma \tilde{\mathbf{x}}'}{\sqrt{(1 + 2\tilde{\mathbf{x}}^T \Sigma \tilde{\mathbf{x}})(1 + 2\tilde{\mathbf{x}}'^T \Sigma \tilde{\mathbf{x}}')}}\big)$	$\boldsymbol{\theta} \triangleq \{\Sigma\}$

2.1.1.4 Model Selection (Inference)

The problem of deducing the structure of GP model that best explains the data is referred to as model selection *i.e.,* selection of apt mean and covariance functions. Without loss of generality, the prior mean is often chosen to be zero [1] while the hyper-parameters need to be tuned to fit the data. For this, *Maximum Likelihood Estimation (MLE)* [5] is employed. This method requires the marginal log likelihood function to be defined as shown in Eq. (2.5).

$$\mathcal{L}(f(\mathbf{x}), \mathbf{x}, \boldsymbol{\theta}) = -\frac{1}{2} f(\mathbf{x})^T K^{-1} f(\mathbf{x}) - \frac{1}{2} \log |K| - \frac{\#(K)}{2} \log |2\pi| \tag{2.5}$$

In Eq. (2.5), $f(\mathbf{x})$ represents the noise-free observations acquired from the nodes in set X. Then, the optimal set of hyper-parameters $\boldsymbol{\theta}_{opt}$ is given by:

$$\boldsymbol{\theta}_{opt} \leftarrow \arg\max_{\boldsymbol{\theta}} \mathcal{L}(f(\mathbf{x}), \mathbf{x}, \boldsymbol{\theta}) \tag{2.6}$$

In order to deduce the optimal hyper-parameters $\boldsymbol{\theta}_{opt}$, one of the following three ways can be adopted:

- Creating a grid of all possible values and then using cross-validation to select the optimal combination.
  - **Pros:**
    Exhaustively considers all possible combinations of parameters.
  - **Cons:**
    The grid resolution must be set manually which not only impacts the quality of soution but also affects the number of evaluations needed before an optimal solution can be found. Thus, making this approach rather slow. Additionally, the complexity incresases with the number of hyper-parameters being catered to.
- Fully Bayesian approach can be utilized whereby prior distribution on hyper-parameters are placed and posterior is optimized using *Markov Chain Monte Carlo (MCMC).*
  - **Pros:**
    The initial guess of the hyper-parameters can bias the inference. Thus, placing a distribution like exponential or normal over the parameters can help infer the optimal combination by marginalizing them while generating the posterior.
  - **Cons:**
    This adds to additional steps during inference can be challenging as the choice of prior can affect the quality of parameters. Additionally, the inference slows down owing to extra computational steps thus incurred. Sampling methods like MCMC need to be resorted to yet the practicality of such methods for real-time performance is questionable.

- A simple gradient descent methods for maximizing the log-likelihood can be utilized, the details of which, can be found in Lemma 9.3 in Appendix 9.1.

  – **Pros:**
  Simple yet elegant method to find the optimal combination of hyper-parameters which remedies the computational bottle-necks of the former methods.

  – **Cons:**
  The objective function being optimized is non-convex and hence, the optimizer can get stuck in a local optima. This can be resolved to some extent by performing multiple restarts.

The grid based and Bayesian optimization based methods are rather slow and pose practical challenges for real-life applications. This is alleviated using the maximum likelihood approach which is the preferred method of choice for this work.

2.1.1.5 Prediction (Estimation)

Given the observations acquired thus far, GPs can be queried to estimate the measurements that are likely to occur at locations of interest. Let $\mathbf{x}$ be the set of observed locations with corresponding measurements $\mathbf{y}$ and let $\mathbf{x}^*$ be the set of unobserved locations which are being queried. For this, the predicted observations for the set $\mathbf{x}^*$ are referred to as $\mathbf{y}^*$. Then, assuming a zero mean GP, the predictive distribution is given by:

$$\begin{aligned} \hat{\boldsymbol{\mu}}_* &\triangleq K_*(K + \sigma^2\mathcal{I})^{-1}\mathbf{y}\,, \\ \hat{\Sigma}_* &\triangleq K_{**} - K_*(K + \sigma^2\mathcal{I})^{-1}K_*^T\,. \end{aligned} \tag{2.7}$$

In Eq. (2.7), the author uses the following shorthand: $K_{**} = K(\mathbf{x}^*, \mathbf{x}^*), K_* = K(\mathbf{x}, \mathbf{x}^*)$ and $K = K(\mathbf{x}, \mathbf{x})$. From Eq. (2.7), it is evident that the predictive mean follows the observations while the predictive variance is shrunk by the information acquired from additional observations. Also, σ represents the noise in observations acquired.

In context of predictions, the author would like to define three keywords here:

- *Interpolation:* refers to making predictions over a previously unobserved location or a set thereof, for the current time-step.
- *Extrapolation:* refers to making predictions over future time-step(s) for a previously observed location or a set thereof.
- *Forecasting:* refers to making predictions for previously unobserved locations for a future time-step.

2.1.2 Potential Applications

GPs are highly modular non-parametric Bayesian methods and their flexibility in the choice of kernels allows their application to both *classification* and regressions applications alike. Some potential applications from each of these aspects are summarized below.

2.1.2.1 GPs applied to Classifications Tasks

- Usually when a patient is newly admitted to a hospital, the staff record several vital statistics like age, blood pressure, height, weight, previous medical ailments, current medications (if any), allergies to medications (if any), nature of ailment *etc.* Based on these parameters, a decision needs to be made if a patient is to be admitted to the general ward (GW) or the intensive care unit (ICU). Thus, there are two class labels *viz.,* GW, ICU and based on a dozen or so variables (vital stats) an accurate decision needs to be made. The problem could be further assessed as scheduling problem such that given the crtical nature of the patients, the patients are assigned to a queue and scheduled to attended in their respective wards.

- When applying for credit cards or bank loans, the concerned agencies receive several applications containing information about the applicants age, marital status, annual income, previous deficits, type of card/loan *etc.* Based on these parameters, a decision needs to be made if the applicant can be selected or rejected. Some applicants however, do not fit into either of these class labels and fall into a gray zone which must be manually handled by the supervisor or can be accounted for by adjusting the classification threshold of the classifier.

- Several applicants join the motor driver school to learn how to drive 2/4 wheeled vehicles either in low weight (motorbikes), moderate weight (cars) or heavy weight (trucks/buses) category. While evaluating the applicants, not only the theoretical test and road tests are important where the applicant is graded based on his/her knowledge of road signs and attentiveness while driving but also a stern background check is essential. The applicant should be in the appropriate driving age limits, physically fit with optimal eye-sight and should not have any prior offenses on record. After accounting for all such parameters, a decision can be made to allow or revoke the application.

- During an auction, several bidders place a bid for the items on display but depending on the age, background, financial status *etc.,* of the attendees a suitable starting bid needs to be selected.

- When a large industrial plant is setup, before a decision is made to buy a certain patch of land, several aspects need to be accounted for. *E.g.,* proximity to the regions that will be used to acquire the raw materials, proximity to the buyers so that transport cost can be minimized, size of the work force, chances of attracting the said size of work force *etc.* Should all these parameters evaluate

to the most feasible combination, only then can a decision be made if the land under consideration should be purchased or not.

- Space exploration missions are carried out using either using mobile robots or CCD cameras which send noisy images of sub-optimal (owing to bandwidth limitations) resolution to the base station. Scientists then need to label all the objects in the scene as stars, galaxy, craters *etc.* with a very high certainty.
- Handwritten digit recognition has been extensively studied and has intrigued researchers for quite some time. Classifying the acquired digit as one of $\{0-9\}$ can also be assisted by supervised learning using GPs.
- Occupancy Grid Maps (OGMs) have also been developed by utilizing GPs to observe the collection of laser beams that bounce of the environment in order to detect if a location is *free* or *occupied*.

2.1.2.2 GPs applied to Regression Tasks

- Inverse Kinematics of robot arm to deduce the joint angles which must be set to reach a required end-effector position.
- Soil Mapping, erosion mapping, surface water monitoring like environmental applications where based on the training samples gathered by a robot or static sensors, a model is generated using GPs to interpolate and generate predictions over other unforeseen regions.
- In financial mathematics, GPs can be used to predict stock market and urban housing prices based on the trends depicted in the past.
- Just like localization *w.r.t* the geometric configuration of the environment, localization can also be performed with respect to the measurement domain. This is alternatively known as *signal strength based localization* where the robot needs to infer its location based on currently acquired noisy observations.

The application of interest as far as this work is concerned, is *regression* and in particular *Environment Monitoring* which some people also refer to as *Intelligent Environment Monitoring* [6] since robots are required to intelligently select and observe parts of a large-scale phenomenon.

2.1.2.3 Pseudo-Code for GP Regression

This section presents the pseduo code for performing the GP regression as shown in Algorithm 1: The algorithm includes calculating the log marginal likelihood which was detailed in Section 2.1.1.4. All matrix inversion operations are assisted by Cholesky Decomposition [7] for stability and faster execution reasons.

Algorithm 1 GPR $(\mathbf{x}, \mathbf{y}, K(\cdot,\cdot), \boldsymbol{\theta}_{init}, \mathbf{x}^*)$

1: **Input**:
- $\mathbf{x}$: Training Inputs (nodes/locations)
- $\mathbf{y}$: Targets (measurements/observations)
- $K(\cdot,\cdot)$: Covariance Kernel function (*e.g.,* rbf, matern$\frac{1}{2}$ *etc.*)
- $\boldsymbol{\theta}_{init}$: Initial guess about hyper-parameters
- $\mathbf{x}^*$: Testing Inputs for interpolation

2: **Output**:
- $\hat{\boldsymbol{\mu}}_*$: Posterior Mean
- $\hat{\Sigma}_*$: Posterior Covariance
- $-\mathcal{L}$: Negative Log Marginal Likelihood

3: $[\sigma_s, l_s, \sigma_n] \leftarrow \boldsymbol{\theta}_{init}$
4: $K = K(\mathbf{x}, \mathbf{x})$ ▷ Compute necessary Covariances
5: $K_* = K(\mathbf{x}^*, \mathbf{x})$
6: $K_{**} = K(\mathbf{x}^*, \mathbf{x}^*)$
7: $K_\epsilon = K + \sigma_n \times \mathcal{I}$ ▷ Noisy Observations
8: $L = \textsc{Cholesky}(K_\epsilon)$ ▷ Cholesky decomposition for Matrix Inversion
9: $\alpha = L^T \backslash (L \backslash \mathbf{y})$ ▷ Solve matrix equation
10: $\mathcal{L} = 0.5 * (\mathbf{y}^T * \alpha - \#(K)\log(2\pi)) - sum(\textsc{Diag}(\log(L)))$ ▷ LML
11: $v = L \backslash K_*$
12: $\hat{\boldsymbol{\mu}}_* = K_* * \alpha$ ▷ Posterior Mean
13: $\hat{\Sigma}_* = K_{**} - v^T v$ ▷ Posterior Covariance
14: **return** $\hat{\boldsymbol{\mu}}_*, \hat{\Sigma}_*, -\mathcal{L}$

2.1.2.4 1-D Interpolation Example

Consider a homoscedatic stationary setting where the domain is 1-D *i.e.,* $x \in \mathcal{R}^1$. Let the noise-free observations be defined by $\mathbf{y} = \mathbf{x}\sin(\mathbf{x})$ where the underlying latent sinusoidal function needs to be inferred. Using a simple 1-D RBF kernel with initial hyper-parameters as $\sigma_s = 1, l = 0.01$, the zero mean prior GP can be generated as shown in Fig. 2.2[1]. From this figure, it is apparent that the term *non-parametric* does not mean that there are no parameters for the model, but it simply means that there are no assumptions about the underlying latent function (sinusoidal in this case). The GP samples from a function space, all possible functions that can best explain the data. Additionally, the impact of length scales is shown via Fig. 2.3 and Fig. 2.4 wherein smaller length scales lead to larger fluctuations and vice versa.

As and when the observations become available, the posterior can be updated. In this case, a batch of 5 observations were made available and the posterior hence generated by plausible functions from the function space is given by Fig. 2.5. In order

[1] drawn using [8].

to find the best-fit function, MLE must be calculated using Eq. (2.5). Then, the optimal hyper-parameters can be obtained which are used to generate the posterior. A noise free case is shown in Fig. 2.6. However, in reality, the observations are noisy *i.e.,* $\mathbf{y} = \mathbf{x}\sin(\mathbf{x}) + \epsilon$ which is modeled in Fig. 2.7. In either case, to ensure that the MLE solution is not a local optima, several restarts were performed and the solution with highest maximum likelihood was finally selected and reported here.

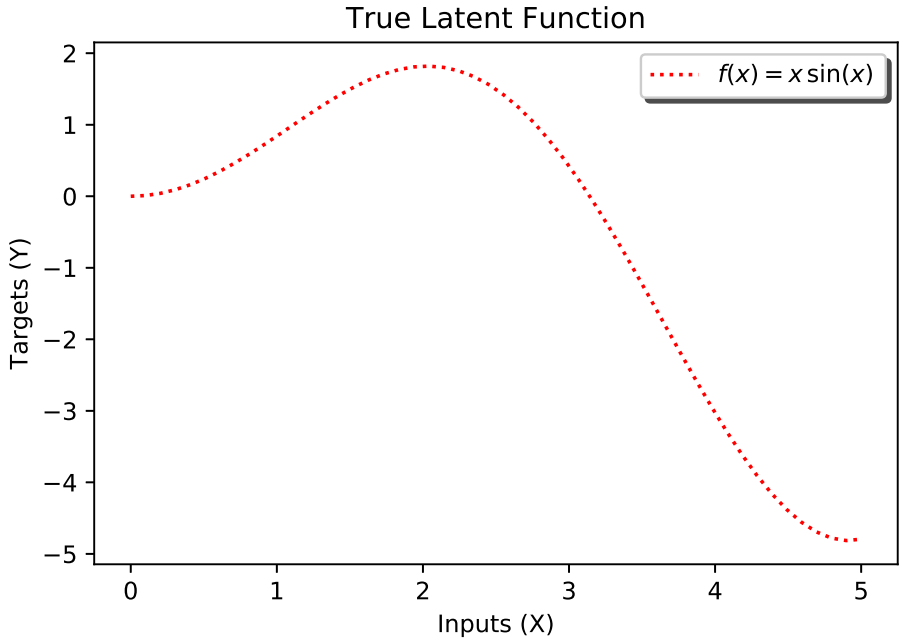


Figure 2.1: 1D Simulated data generated using the user defined sinusoidal function such that $\mathbf{y} = \mathbf{x}\sin\mathbf{x}$.

2.1.2.5 Why use GPs ?

Below, the advantages of GPs are highlighted. Owing to these, they have come across as really powerful tools for modeling highly non-linear environmental dynamics.

- **Analytic inference.** GPs are supervised learning models but the human supervision can be said to be limited to choosing the apt kernel functions. Based on the observations acquired, optimal parameters can be inferred using maximum likelihood estimation from Eq. (2.5).

- **Expressiveness.** A wide variety of variations in the environment of interest can be expressed using appropriate choice of covariance functions. Can even be used to articulate correlations between multiple output variables using multi-output GPs [9]. This is useful in case when the target measurements *e.g.,* algal bloom are also affected by multiple additional natural conditions like wave patterns, humidity, light intensity *etc.*

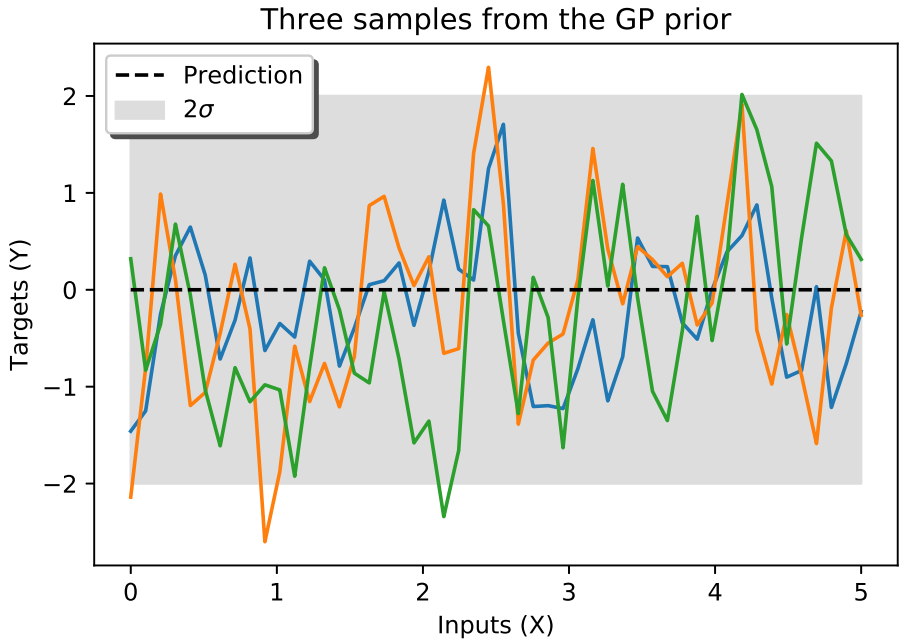


Figure 2.2: 1D Priors. Drawing 3 samples from the prior distribution using a stationary RBF kernel with zero mean and $\sigma_s = 1, l = 0.01$.

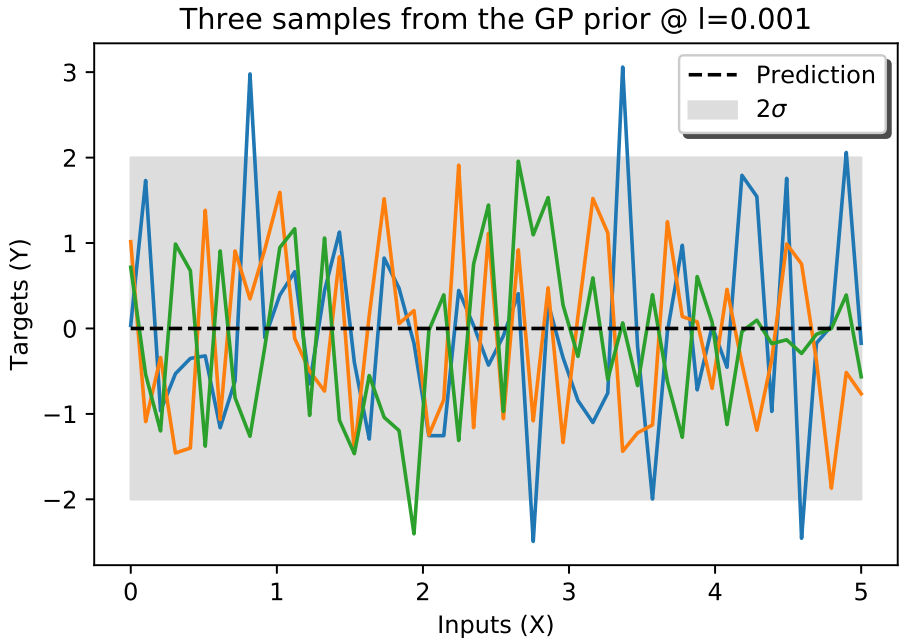


Figure 2.3: 1D Priors with smaller length scales. Drawing 3 samples from the prior distribution using a stationary RBF kernel with $\sigma_s = 1, l = 0.001$.

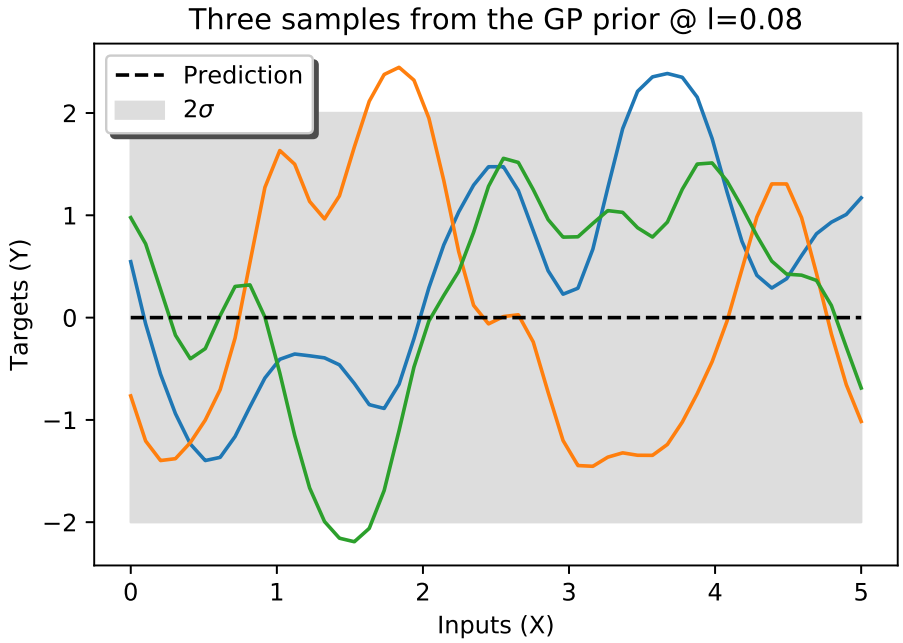


Figure 2.4: 1D Priors with larger length scales. Drawing 3 samples from the prior distribution using a stationary RBF kernel with $\sigma_s = 1, l = 0.08$.

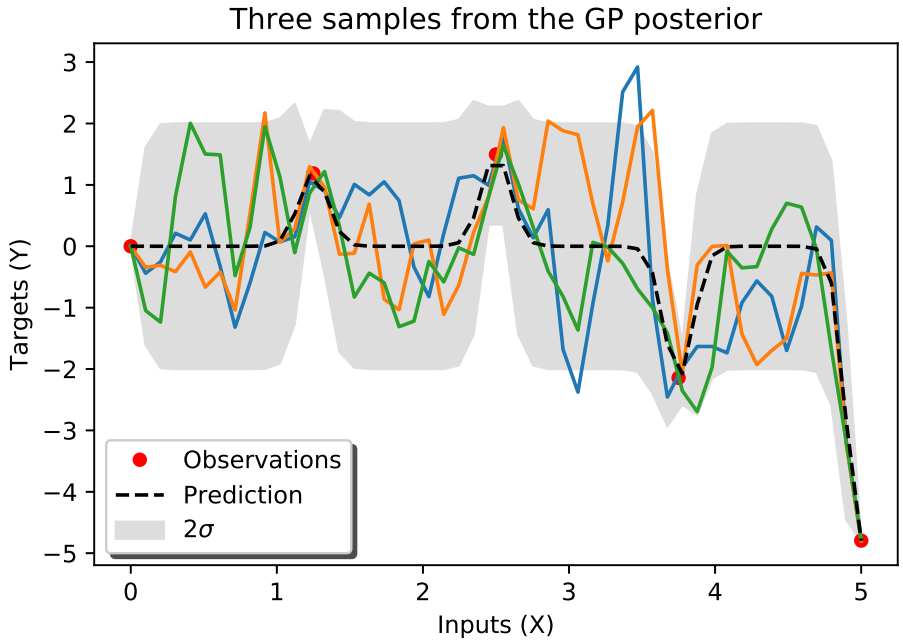


Figure 2.5: 1D Posterior Function (Interpolation). Interpolation after 5 observations are made available.

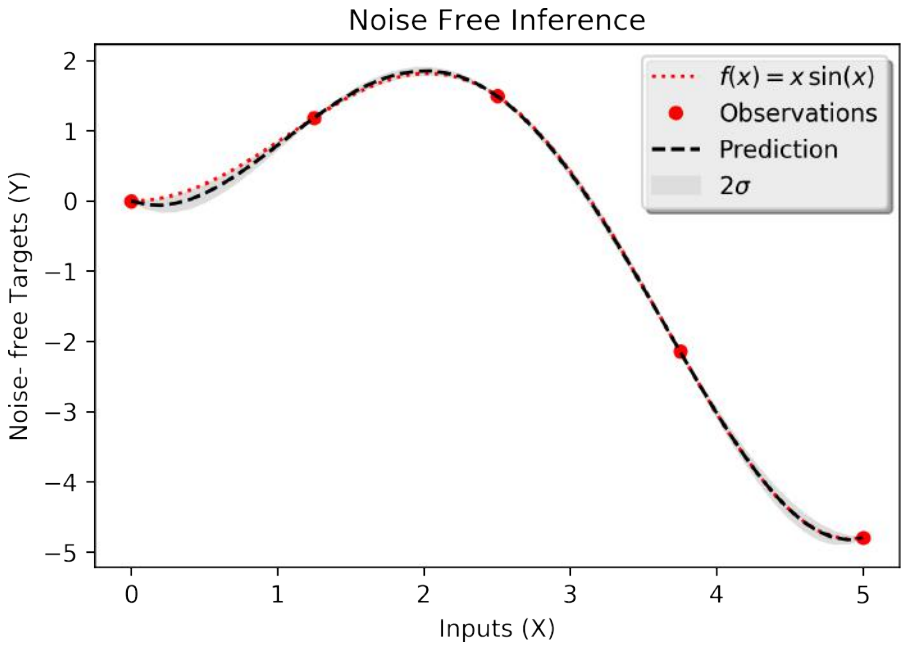


Figure 2.6: 1D Noise Free Interpolation with MLE.

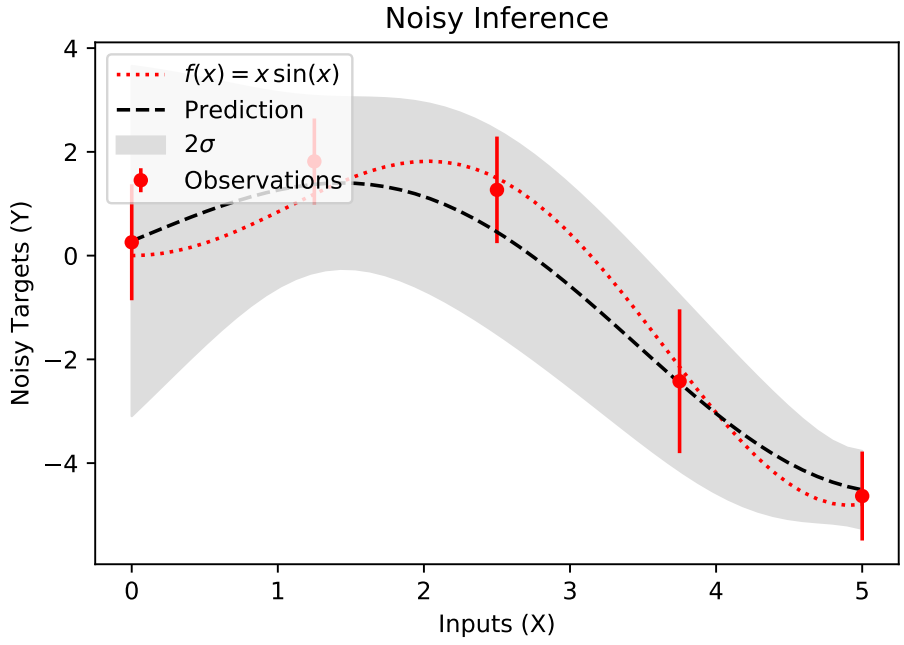


Figure 2.7: 1D Noisy Interpolation with MLE.

- **Closed-form predictions.** Given a set of observations, GPs can be queried to deduce measurements/ function values at unobserved locations. Equivalently, this is the same as generating posterior predictive distributions which have closed form solutions.
- **Quantifiable predictive uncertainty.** Supervised learning frameworks may need additional models to quantify predictive uncertainty like [10] but the predictive uncertainty for GPs is easily quantifiable using the posterior covariance (entropy).

2.1.2.6 Limitations of GPs

GPs are beset with several limitations. Some of them are summarized below.

- **Slow Inference.** Optimizing the hyper-parameters and predicting the function values at locations of interest incurs $\mathcal{O}(N)^3$ time and $\mathcal{O}(N)^2$ memory, for N being the size of the dataset being modeled. This restricts the exact inference only to the order of 10^3 datapoints. Any larger datasets must be addressed using approximate inference methods.
- **Apt Kernel Choice.** Although GPs are non-parametric in nature but the covariance function need to be chosen in accord with the dynamics of the field being modeled since the kernel essentially represents the input space. Optimizing the marginal log likelihood from Eq. (2.5) can provide with optimal parameter setting but the supervisor needs to set the parametric form of the kernel.
- **No consideration for robot resources.** GPs were designed as machine learning models that can reach high computational speeds using GPU optimization but when deploying real robots either the robots are forced to serve as mobile sensor nodes relaying the data to a powerful base node or approximate methods need to be used to optimize the GPs to work with limited hardware capabilities.
- **Designed for batch processing.** GPs were inherently designed for batch-processing and big-data applications. Optimizing the parameters using a stream of sequential data is a rather young research domain.

Having discussed both the pros and cons of using the GPs and several potential applications where they can be seamlessly integrated, one daunting questions still remains: *What is it that makes them the model of choice?* The answer to this question is rather concise and that is, the fact that these models have a flexible covariance kernel that can be used to explain the correlations in most of the potential applications discussed above along with the ability to quantity the predictive capabilities makes them the model of choice. Other machine learning methods like Neural Networks, Gaussian Mixture Models *etc.,* could have been used but then additional models would be required to quantify the performance. Thus, for all intents and purposes, this works will utilize GPs as the model that can effectively explain the underlying dynamics of the phenomenon of interest *viz.,* for *Intelligent Environment Monitoring*.

2.2 Information-Theoretic Path Planning

This sections introduces information-theoretic path planning approaches which are best suited for gathering optimal training samples to train the GPs. Information-theoretic path planning or simply Informative path planning corresponds to the task of autonomous decision making for acquiring useful data for modeling the target phenomenon being observed. There are several ways in which *"information"* can be defined. They are enlisted below:

- *Entropy [11]:*
  This metric models the uncertainty in the random variable. Let $\mathbf{y} \in D$ with a probability mass function $p(\mathbf{y})$ be the continuous random variable of interest. Then, the entropy is defined as:

  $$\mathbb{H}(\mathbf{y}) = -\int_D p(\mathbf{y}) \log p(\mathbf{y}) d\mathbf{y} \tag{2.8}$$

  For continuous random variables, this metric is more commonly known as Differential entropy or continuous entropy as the entropy itself can also be defined for discrete cases which were not considered herewith.

- *Mutual Information (MI) [11]:*
  This metric encodes the reduction in uncertainty about the random variable $\mathbf{y} \in D$ given some observations were acquired. In other words, this represents a measure of average information acquired by the observations with respect to the random variable of interest. Mathematically,

  $$MI(\mathbf{y}) = \mathbb{H}(\mathbf{y}) - \mathbb{H}(\mathbf{y}|\mathbf{y}') \tag{2.9}$$

  where $\mathbf{y}'$ represents the observations.

- *Rény Information or α divergence [12]:*
  It is used to measure the distance or difference between two probability mass functions (pmf) *viz.,* posterior and prior over the parameters of the model which define the dynamics of the phenomenon of interest. If p represents the posterior over the parameter $\mathbf{y} \in D$ which is updated as and when new observations are available while q represents the prior over $\mathbf{y}$, then,

  $$D_\alpha(p||q) = \frac{1}{\alpha - 1} \log \int_D p^\alpha(\mathbf{y}) q^{1-\alpha}(\mathbf{y}) d\mathbf{y} \tag{2.10}$$

- *Kullback-Leibler (KL) divergence [12]:*
  As the value of $\alpha \to 1$, Rény Information $\to$ Kullback-Leibler (KL) divergence. Thus,

  $$D_{KL}(p||q) = \int_D p(\mathbf{y}) \log \frac{p(\mathbf{y})}{q(\mathbf{y})} d\mathbf{y} \tag{2.11}$$

- *Cauchy-Schwarz (CS) divergence [12]:*
  This metric is useful when the distributions of $p(\mathbf{y})$ and $q(\mathbf{y})$ for $\mathbf{y} \in D$ cannot be defined in parametric forms. Mathematically, this means:

$$D_{CS}(p||q) = \log \frac{\int_D p^2(\mathbf{y})d\mathbf{y} \int_D q^2(\mathbf{y})d\mathbf{y}}{[\int_D p(\mathbf{y})q(\mathbf{y})d\mathbf{y}]^2} \tag{2.12}$$

- *Fisher Information [13]:*
  For a continuous random variable $\boldsymbol{\theta}$, the Fisher Information Matrix (FIM), $FIM(\mathbf{y})$, explains the amount of information contained in the set of observations given by $\mathbf{y}$. In order to ensure that the observations are highly informative, the FIM should be maximized in the norm sense [14]. Let the observation be given $\boldsymbol{\zeta}$ and $p(\mathbf{y})$ represent the functional form of the random variable, then the $FIM(\mathbf{y})$ is given by:

$$FIM(\mathbf{y}) = \int_{\mathbf{y}\in D} \Big(\frac{d \log p(\mathbf{y}|\boldsymbol{\zeta})}{d\boldsymbol{\zeta}}\Big)^2 p(\mathbf{y}|\boldsymbol{\zeta})d\mathbf{y} \tag{2.13}$$

  where, the term $\dfrac{d \log p(\mathbf{y}|\boldsymbol{\zeta})}{d\boldsymbol{\zeta}}$ represents the sensitivity of functional form of the random variable being modeled [15].

In environment monitoring applications, since probabilistic models are employed, information refers to reduction in uncertainty to better predict the phenomenon. Thus, the goal position of the robot is not pre-meditated, rather, is sequentially chosen based on the locations that lead to maximal information gain. Of all the possible ways of defining the *"information"*, this now reduces to entropy based methods. The existing cost functions for Informative path planning usually focus on two aspects *viz.,* :

- *Density estimation:* Objective functions are designed to reduce the overall uncertainty of the model trying to make a globally confident density estimator. This has previously been used in the works like [16–18]. There are two very closely related approaches to density estimation path planning:

  - **Entropy Maximization Criterion:**
    These functions usually consider information in terms of *entropy* which is a logarithmic function of variance. Let $\sigma(x)$ represent the variance at a location x. Then entropy of the location is given by $\mathbb{H}(x) \triangleq 0.5 \log \sigma(x)$. Entropy is a measure of uncertainty or a lack of information about location x. Thus, these path planners choose the *next-best-location* in the areas of highest uncertainty using the following criteria:

$$\hat{x} \triangleq \arg\max_{x>} \{\mathbb{H}(x>)\} \tag{2.14}$$

    where $x>$ represents all possible candidates that are being evaluated.

* **Pros:**
  Information is acquired to be able to understand the dynamics of the entire domain.
* **Cons:**
  Gaining expertise over the entire phenomenon is rather difficult. The dynamics of the environment get more complicated as the expanse of the sensing area increases and makes it harder to perform equally well in all parts of the field. Moreover, entropy maximization criterion implies that the observations be gathered from far away locations (near the boundary of the sensing area) which leads to wastage of information [11].

– **Mutual Information Criterion:**
  While using the conditional entropy based formulation as explained in Eq. (2.14), is useful to observe the locations which are most informative with respect to the entire domain, an improved designed criterion as argued by authors in [11] is the *Mutual Information*. The objective here is to find the subset of nodes from the domain which significantly reduce the uncertainty over the rest of the unobserved portion of the field. Formally, if $x >$ representes the candidate locations that can be observed and O represents the observations already acquired, then, the mutual information based informative path planning objective function is given by:

$$\hat{x} \triangleq \arg\max_{x>} \{\mathbb{H}(x >) - \mathbb{H}(x > |O)\} \tag{2.15}$$

  From Eq. (2.15), it is apparent that the set $\hat{x}$ maximally reduces the entropy over the space $D \backslash \hat{x}$ where D represents the domain of the phenomenon.

  * **Pros:**
    Candidate locations to be observed are selected based on prediction quality over the unobserved subset of the sensing area thereby preventing observations from being acquired along boundaries all the time.
  * **Cons:**
    Only reduces the uncertainty over the unobserved subset of the field as opposed to the former criterion which reduces uncertainty over the entire field.

- *Hotspot estimation:* Objective functions are biased only towards hotspots *i.e.,* areas exhibiting extreme values. These functions consider information *i.e.,* $\sigma(x >)$ in unison with the predicted measurements ($\mu(x >)$) at the location of interest. Thus, these path planners choose the *next-best-location* in the areas of peaked measurements using:

$$\hat{x} \triangleq \arg\max_{x>} \{\mu(x >) + \kappa\sigma(x >)\} \tag{2.16}$$

where, the parameter κ controls the amount of exploration. This cost function was previously discussed in [6].

- **Pros:**
  The aim is to be an expert of modeling the hotspot. Being an expert over a small subset of the domain is easier than performing equally well over the entire domain.
- **Cons:**
  Given that the environment is largely unknown, it cannot be known as to how many hotspots exist. Additionally, just by modeling the hotspot, the dynamics of the entire phenomenon cannot be modeled unless the hotspot dissemination model is given which explain how the correlation varies as the distance from hotspot increases.

The former is useful in scenarios where predictive errors could lead to monetary losses as is the case in stock market forecasting or even pose a risk to human lives like in civil engineering, architectural designing projects *etc.* while, the latter is useful in scenarios like oils spills, nuclear radiation *etc.* whereby the hotspot must be accurately identified and contained. In this work, density estimation based informative path-planning will be addressed in light of resource-constraints such that limitations of placing sensors at boundary locations can be remedied.

2.3 Assessment Criteria

In order to empirically evaluate the model performance, the estimated values can be compared with respect to the ground truth values using any of the following indicators. Each performance metric is evaluated over all the M samples that were used to generate the model.

- **RMSE**: This performance indicator is used to evaluate the performance of the generated model by giving uniform significance to each and every test point. Then the RMSE is given by:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{M} \left(\boldsymbol{\mu}(x_i) - \boldsymbol{\mu}_{\mathbf{gt}}(x_i)\right)^2}{M}} \tag{2.17}$$

- **WRMSE**: A further enhancement to RMSE is now defined as Weighted RMSE [6] where instead of uniformly evaluating the performance over the entire field, now we perform a weighted evaluation. The weight essentially determines the performance of the estimator in areas of keen interest i.e., either evaluated over the areas that exhibit extreme values or those that showcase extreme variance. Then the WRMSE is given by:

$$WRMSE = \sqrt{\frac{\sum_{i=1}^{M} \left[\left(\boldsymbol{\mu}(x_i) - \boldsymbol{\mu}_{\mathbf{gt}}(x_i)\right)^2 \times W_i\right]}{M}} \tag{2.18}$$

where,

$$W_i = \begin{cases} \frac{\boldsymbol{\mu_{gt}}(x_i) - \min(\boldsymbol{\mu_{gt}}(x_i))}{\max(\boldsymbol{\mu_{gt}}(x_i)) - \min(\boldsymbol{\mu_{gt}}(x_i))}, & \text{if evaluating over extreme values} \\ \frac{\boldsymbol{\sigma_{gt}}(x_i) - \min(\boldsymbol{\sigma_{gt}}(x_i))}{\max(\boldsymbol{\sigma_{gt}}(x_i)) - \min(\boldsymbol{\sigma_{gt}}(x_i))}, & \text{otherwise} \end{cases} \tag{2.19}$$

- **Cramér-Rao Lower Bound (CRLB) [19]**: Given the Fisher Information Matrix (FIM) [20] generated based on observations up until the current iteration and the current estimate of hyper-parameters, the CRLB states that:

$$\mathbb{E}\left[(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})^T\right] \geq FIM^{-1} \tag{2.20}$$

Eq. (2.20) states that the minimal error in estimation of the hyper-parameters of the covariance matrix is bounded by the inverse of the FIM where, the FIM is given by:

$$[FIM(\mathbf{y}, \mathbf{x})]_{ij} = \frac{1}{2} tr\left(K^{-1}\frac{\partial K}{\partial K(x_i)}K^{-1}\frac{\partial K}{\partial K(x_j)}\right) \tag{2.21}$$

Given that the underlying (true) latent function that explain the environmental dynamics cannot be known, the hyper-parameters obtained upon convergence of the inference methods are always estiamted values of the true hyper-parameters of the generating function. The only way to evaluate the quality of these hyper-parameters is to theoretically evaluate the quality *i.e.,* variance of each of the hyper-parameters. For this, the A-optimality criterion of optimization of FIM can be utilized over the currently available best estimate $\hat{\boldsymbol{\theta}}$ as explained in [19]. Minimizing the CRLB in turn leads to minimization of uncertainty in estimation of true hyper-parameters and hence represents the theoretical bound of confidence that the model can achieve given the current estimates.

2.4 Hardware Description

Part of this work has been evaluated on real robot hardware. For this, a commercially available light weight UAV *Parrot ARDrone 2.0* shown in Fig. 2.8, and a custom UGV with a DIY chassis and custom electronics were utilized. For the UGV, a preliminary variant V1.0 (Fig. 2.9) was designed which was used for indoor trials followed by its successor V2.0 (Fig. 2.12) which was used for outdoor field trials.

2.4.1 ArDrone 2.0

The ArDrone 2.0 (shown in Fig. 2.8) is a light-weight quadrotor platform designed by Parrot® and is well suited for both indoor and outdoor missions. It comes equipped with a $720p$ $30fps$ HD camera, 4 brushless motors operating at $14.5W, 28500rpm$, 3 axis gyroscope with $2000°/sec$ precision, 3 axis accelerometer with $\pm 50mg$ precision, 3 axis magnetometer with 6° precision, along with additional hardware for on-board processing and in-flight stabilization. Besides these, for outdoor experiments, an external GPS flight recorder with an accuracy of $\pm 2m$ was also utilized.

Figure 2.8: ArDrone 2.0 with GPS flight recorder [21] used for outdoor experiments.

2.4.2 Rusti V1.0

This test platform is a lightweight Omnirover 2.0 kit which was named *Rustic-Wanderer* or *Rusti V1.0*, equipped with 4 omni-directional wheels as shown in Fig. 2.9. In the original kit, the robot itself was powered by a 4×AA battery pack with alkaline batteries and controlled by Arduino/Genuino ATMega328p MCU. However, to be able to store lengthy data logs and repeat the experiments with varying duty cycles, the default MCU was replaced by an Arduino Mega2560 board (shown in Fig. 2.10) and power source was replaced by a rechargable $7V/2200mAH$ LiPo battery instead.

The same power source supplied power to the MCU and the $6V$ BO motors. The maximum attainable velocity of the robot is $\approx 1\ m/s$. Constant transmission power and idling power for the XBee (short range wireless communication module) were assumed. Based on the data sheet[2], it was concluded that the idling power for the XBee communication module (Fig. 2.11) is $0.165W$ while the transmission power increases the consumption by a meager $0.001W$.

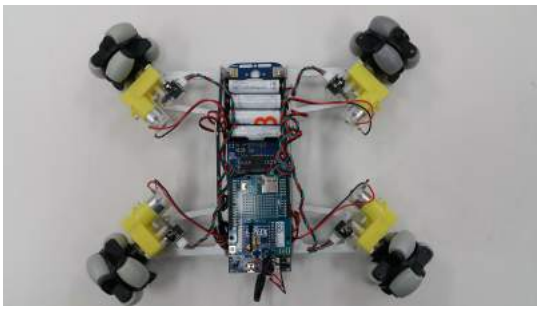


Figure 2.9: Rusti V1.0 with omnidirectional wheels used for indoor experiments.

[2]Available at: https://tinyurl.com/zgcv3ol

Figure 2.10: Arduino Mega2560 micro-controller unit (MCU) and ultrasonic range sensor used with Rusti V1.0.

Figure 2.11: XBee ZB ZigBee Mesh Module 2.4GHz 2mW with Wire Antenna for short-range wireless communication [22].

2.4.3 Rusti V2.0

As an enhnacement over its successor, *Rusti V2.0* (showcased in Fig. 2.12) has a stronger alloy frame and powerful $12V$ DC geared motors with high torque. Instead of Arduino Mega2560, the Raspberry Pi 3 kit was utilized for on-board data logging which was also fed control commands from the human operator using wired Double Pole Double Throw (DPDT) switches. Circuit diagram is shown in Fig. 9.1, Appendix. 9.3.1. Since, the Raspberry Pi outputs $3.3V$ while the motor driver requires $5V$ as logic 1 to enable the motors, a level shifter was designed, the details of which can be found in Fig. 9.2, Appendix. 9.3.1. Since the robot was tested in uneven outdoor terrains, an external wearable sensor from Empatica [23] was fixed firmly on the robot and the internal 3 axis accelerometer was harnessed to record elevation changes online. The default operational frequency of 32 Hz was used and the sensor is shown in Fig. 2.13.

Figure 2.12: Rusti V2.0 with all-terrain wheels and external 3-axis accelerometer sensor for outdoor field experiments carried out on asphalt, grass and tiles.

Figure 2.13: External 3-axis accelerometer sensor called Empatica E4 [23] used with Rusti 2.0 to record terrain elevation changes during outdoor field trials.

2.5 Summary

This chapter serves as a primer to lay out the foundation for developing the planning algorithms for this thesis. GP regression is a complex regression technique based on Bayesian inference, which is used to model an unknown function using noisy observations. By learning the model hyper-parameters, a GP provides an accurate probabilistic representation of any function along with uncertainty estimates. GPs along with *density estimation* based path planning approaches will be used hereon to model real world environment monitoring phenomenon. For a more hands-on experience with $1D$ GPs the readers are referred to [25] or [26].

Bibliography

[1] C. E. Rasmussen and C. K. Williams, *Gaussian processes for machine learning*, vol. 1. MIT press Cambridge, 2006.

[2] M. G. Genton, "Classes of kernels for machine learning: a statistics perspective," *Journal of machine learning research*, vol. 2, no. Dec, pp. 299–312, 2001.

[3] S. Garg, A. Singh, and F. Ramos, "Learning non-stationary space-time models for environmental monitoring," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 25, p. 45, 2012.

[4] A. Gretton, K. Fukumizu, C. H. Teo, L. Song, B. Schölkopf, and A. J. Smola, "A kernel statistical test of independence," in *Advances in neural information processing systems*, pp. 585–592, 2008.

[5] F. Scholz, "Maximum likelihood estimation," *Encyclopedia of statistical sciences*, 1985.

[6] R. Marchant and F. Ramos, "Bayesian optimisation for intelligent environmental monitoring," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 2242–2249, IEEE, 2012.

[7] G. Upton and I. Cook, *A dictionary of statistics 3e*. Oxford university press, 2014.

[8] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: experiences from the scikit-learn project," in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108–122, 2013.

[9] M. A. Osborne, S. J. Roberts, A. Rogers, S. D. Ramchurn, and N. R. Jennings, "Towards real-time information processing of sensor network data using computationally efficient multi-output gaussian processes," in *Proceedings of the 7th international conference on Information processing in sensor networks*, pp. 109–120, IEEE Computer Society, 2008.

[10] S. Kim, Z. Yu, R. M. Kil, and M. Lee, "Deep learning of support vector machines with class probability output networks," *Neural Networks*, vol. 64, pp. 19–28, 2015.

[11] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *Journal of Machine Learning Research*, vol. 9, no. Feb, pp. 235–284, 2008.

[12] W. Lu, *Autonomous sensor path planning and control for active information gathering*. PhD thesis, Duke University, 2014.

[13] D. Levine, B. Luders, and J. P. How, "Information-rich path planning with general constraints using rapidly-exploring random trees," 2010.

[14] D. Ucinski, *Optimal measurement methods for distributed parameter system identification*. CRC Press, 2004.

[15] A. Ly, M. Marsman, J. Verhagen, R. Grasman, and E.-J. Wagenmakers, "A tutorial on fisher information," *arXiv preprint arXiv:1705.01064*, 2017.

[16] A. Krause and C. Guestrin, "Near-optimal observation selection using submodular functions," in *AAAI*, vol. 7, pp. 1650–1654, 2007.

[17] A. Singh, A. Krause, and W. J. Kaiser, "Nonmyopic adaptive informative path planning for multiple robots.," in *IJCAI*, vol. 3, p. 2, 2009.

[18] A. Singh, F. Ramos, H. D. Whyte, and W. J. Kaiser, "Modeling and decision making in spatio-temporal processes for environmental surveillance," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 5490–5497, IEEE, 2010.

[19] Y. Xu and J. Choi, "Adaptive sampling for learning gaussian processes using mobile sensor networks," *Sensors*, vol. 11, no. 3, pp. 3051–3066, 2011.

[20] S. M. Kay, *Fundamentals of statistical signal processing*. Prentice Hall PTR, 1993.

[21] P. Inc., "Parrot Flight Recorder GPS for AR.Drone 2.0 Flying Drone." `https://tinyurl.com/yaf7krpv`. [Online; accessed 21-Dec-2017].

[22] G. US, "XBee ZB ZigBee Mesh Module 2.4GHz 2mW with Wire Antenna." `http://www.gravitech.us/xbzbmo22mwwa.html`. [Online; accessed 21-Dec-2017].

[23] Empatica, "Real-time physiological signals-E4 EDA/GSR sensor." `https://www.empatica.com/e4-wristband`.

[24] K. Deb, "Multi-objective optimization," in *Search methodologies*, pp. 403–449, Springer, 2014.

[25] K. Tiwari, "Hands-on experience with gaussian processes (gps): Implementing gps in python-i," *arXiv preprint arXiv:1809.01913*, 2018.

[26] K. Tiwari, "Hands-on Experience with Gaussian Processes (GPs)." `https://sites.google.com/view/exposure-to-gp/home`, 2018.

Part II

Map Phase: Environment Modeling

Chapter 3

Modeling the Spatial Variations of the Environment using Stationary Homoscedastic GPs

> *Mystery however is a very necessary ingredient in our lives. Mystery creates wonders and wonder is the basis of man's desire to understand. Who knows what mysteries will be solved in our lifetime, and what new riddles will become the challenge of the new generations?*
>
> Neil Armstrong, *1969*

Humans are innately curious and the mysteries of nature intrigues them. The underlying dynamics are highly non-linear making them difficult to model and even more so to quantify the model quality. In Chapter 2, the author briefly discussed a non-parametric Bayesian method called **Gaussian Process (GP)** [1]. In this chapter, the author further expatiates by discussing the GPs applied specifically to model the natural environmental phenomenon like precipitation, surface run-off, surface water quality, pollution levels *etc.* GPs are fully characterized by their mean and covariance functions and the choice of these functions critically affects the model quality. For the scope of this work, a zero mean GP with stationary homoscedastic (input independent noise) kernel was used, the details of which follow.

3.1 Mean Function

For convenience, without any loss of generality, most researchers set the prior mean function to be zero [1]. This assumption is not restrictive as the posterior mean takes the conjugate form of the likelihood. However, if the prior mean function is known to be non-zero, change of variables proves helpful such that, if, $f \sim GP(\boldsymbol{\mu}, K)$ then, $f' \triangleq f - \boldsymbol{\mu}$ is the new zero mean GP with $f' \sim GP(0, K)$. Hence, instead of doing

inference for f, the same can be done for f' and once the posterior mean is obtained, it can simply be added back the prior mean ($\boldsymbol{\mu}$) to obtain the posterior for f.

3.2 Covariance Function

In order to model the spatially correlated domain $D \in \mathcal{R}^d$, this work resorts to stationary homoscedastic kernel *viz.,* radial basis function (rbf) or more commonly known as the squared exponential kernel [1]. This kernel is *isotropic* and takes the form:

$$K(\mathbf{x}, \mathbf{x}') = \sigma_{sig}^2 \exp\big(-\tfrac{1}{2}(\mathbf{x}-\mathbf{x}')^T L^{-1}(\mathbf{x}-\mathbf{x}')\big) \tag{3.1}$$

where $\mathbf{x}, \mathbf{x}' \in D$, $L = \text{diag}(l_1^2, \ldots, l_d^2)$ and the l_i are characteristic length scales, which determine the relevance of the corresponding input dimension for modeling the spatial phenomenon. σ_{sig}^2 corresponds to the amplitude of the signal to be modeled.

Usually, in real environments, the observations are noisy owing to hardware limitations thus, the noisy variant of the kernel shown in Eq. (3.1)[1] is now obtained by:

$$K_\epsilon(\mathbf{x}, \mathbf{x}') = \sigma_{sig}^2 \exp\big(-\tfrac{1}{2}(\mathbf{x}-\mathbf{x}')^T L^{-1}(\mathbf{x}-\mathbf{x}')\big) + \sigma_n^2 I \tag{3.2}$$

where σ_n^2 describes the magnitude of the noise. For the scope of this work, the author considers input-independent uniform noise level in the environment such that $\sigma_n \sim \mathcal{N}(0, \epsilon)$. The hyper-parameters are $\boldsymbol{\theta} \triangleq \{\sigma_{sig}^2, \sigma_n^2, l_1, l_2, \ldots, l_d\}$.

3.2.1 Optimal Hyper-parameters

The hyper-parameters are trained using the standard procedure of evidence (type-II marginal likelihood) maximization [2]. Evidence maximization avoids over fitting by automatically trading off data fit and model complexity. The detailed derivations are in Lemma 9.3 of Appendix 9.1. Let $O \subset D$ define the set of observed nodes associated with $\mathbf{y}$ as the observed measurements. The log marginal likelihood deduced by marginalizing out the hyper-parameters is given by:

$$\mathcal{L}(\mathbf{y}, O, \boldsymbol{\theta}) = -\frac{1}{2}\mathbf{y}^T K_\epsilon^{-1}\mathbf{y} - \frac{1}{2}\log|K_\epsilon| - \frac{\#(K_\epsilon)}{2}\log|2\pi| \tag{3.3}$$

where $K_\epsilon = K_\epsilon(O, O)$.

Then, the optimal hyper-parameters which maximize the log likelihood are then given by:

[1]The readers are hereby cautioned that there is a slight abuse of notation here. $\mathbf{x} - \mathbf{x}'$ does not represent subtraction between vectors. Rather it denotes pair-wise distance operation resulting in a matrix where the i, j element represents the distance between $[\mathbf{x}_i, \mathbf{x}'_j]$.

$$\hat{\boldsymbol{\theta}}_{opt} \leftarrow \arg\max_{\boldsymbol{\theta}} \mathcal{L}(\mathbf{y}, O, \boldsymbol{\theta}) \tag{3.4}$$

In Eq. (3.4), the optimal hyper-parameters are represented by $\hat{\boldsymbol{\theta}}_{opt}$ since at any given time only the estimation of the parameters is known instead of the true value. The objective function shown in Eq. (3.3) is non-convex which means that the optimizer can get stranded in a local maximum. An illustration of such a situation is shown in Fig. 3.1 wherein a sample log-marginal likelihood landscape is shown. As annotated in the figure, there are potentially two local maxima: the first one is obtained for a combination of small noise and shorter length scale while the second is obtained for a combination of high noise and longer length scales. The former combination explains the data purely based on noise-free model while the latter tends to model the data variations purely by noise. A shorter length scale would lead to good model fit but the kernel would be diagonal making very complex models since the $\frac{1}{2}\log|K_\epsilon|$ term would be large while the larger length scales lead to better model fit but the entries of the kernel would be very close to 1 leading to a very small $\frac{1}{2}\log|K_\epsilon|$ and a very simple model. The log-marginal likelihood would be higher for the former case but depending on the initial configuration of hyper-parameters (for *e.g.,* shown by + and + in Fig. 3.1), the optimizer may converge to either optima. Thus, multiple restarts are preferred such that the optimizer finds all potential saddle points and converges to the optimal that maximizes the log marginal likelihood. This additionally opens up a new challenge:

> *It can not be known a priori, as to how many local optima are to be expected. In light of this, how many random restarts are enough?*

This is an interesting yet complex problem since each local minima is a particular interpretation of the data and was not addressed in the scope of this work. However, some prior work can be found from [7].

3.3 Posterior (predictive) distribution for Interpolation

Let $U \subset D$ define the set of (interpolated) unobserved nodes such that $U = O^c$ with associated observations denoted by $\mathbf{y}^*$. Thus, the posterior distribution over elements in U is given by $\mathcal{N}(\boldsymbol{\mu}_{U|O,\boldsymbol{\theta}}, \Sigma_{UU|O,\boldsymbol{\theta}})$ where:

$$\boldsymbol{\mu}_{U|O,\boldsymbol{\theta}} \triangleq K_* K_\epsilon^{-1}(\mathbf{y} - \boldsymbol{\mu}_O)\,, \tag{3.5}$$

$$\Sigma_{UU|O,\boldsymbol{\theta}} \triangleq K_{**} - K_* K_\epsilon^{-1} K_*^T\,. \tag{3.6}$$

In Eq. (3.5) and Eq. (3.6), the following short hands were used: $K_* = K(U, O), K_{**} = K(U, U)$ and $K_\epsilon = K_\epsilon(O, O)$. The posterior mean from Eq. (3.5) is used to predict the measurements at unobserved nodes $\mathbf{x}^* \in U$ while the posterior covariance from

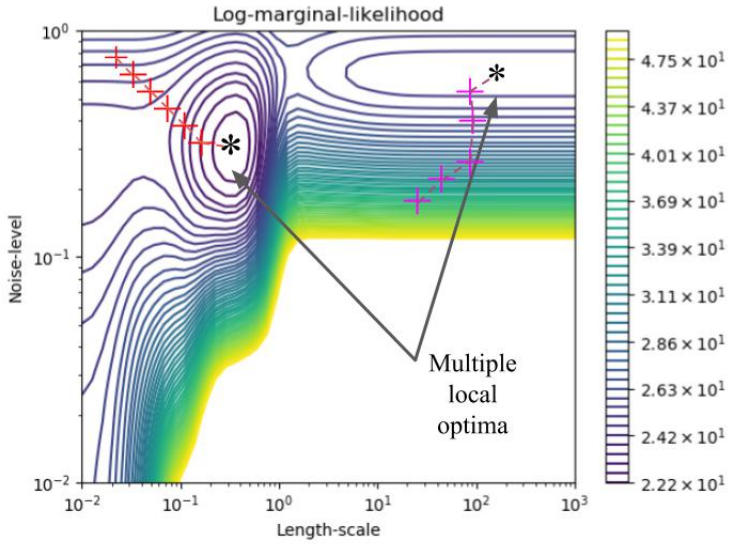


Figure 3.1: An illustration of Log-marginal likelihood landscape taken from [3]. Sample start locations for gradient descent (during multiple restarts) are shown by + and +.

Eq. (3.6) is used to infer the uncertainty of the inferred model. The uncertainty (entropy) of the GP model can be defined in one of two ways:

$$\mathbb{H}_{\mathbf{y}^*|\mathbf{y},\boldsymbol{\theta}} \triangleq \text{tr}[\Sigma_{UU|O,\boldsymbol{\theta}}] \tag{3.7}$$

or,

$$\mathbb{H}_{\mathbf{y}^*|\mathbf{y},\boldsymbol{\theta}} \triangleq \frac{|U|}{2}\ln(2\pi e) + \frac{1}{2}\ln(|\Sigma_{UU|O,\boldsymbol{\theta}}|) \tag{3.8}$$

The former (Eq. (3.7)) assumes conditional independence between measurements in U while this is accounted for aptly by the latter (Eq. (3.8)). Eq. (3.8) avoids model overfitting by evading overestimation of model uncertainty [4].

3.4 Extending to Multiple GPs for Distributed Computations and Interpolation

Owing to its cubic complexity in the size of data for inverting the matrix and quadratic time complexity in Eq. (3.5)-Eq. (3.10), GP regression is restricted to dataset sizes of the order of 10^3 samples. To overcome this, several researchers have proposed selecting a subset of the data and performing inference only over the chosen subset like [4, 5]. Even though the computational cost is reduced owing to operations on a small subset of data, but the amount of variance over these points cannot be utilized to infer the variance over the entire dataset. Similar approaches are utilized in dimensionality reduction using sparse GP regression which utilize the "inducing points" to represent the entire dataset [6]. Here again the computational cost will be less than that of handling the entire dataset but tuning and apt selection of inducing points in real-time

is a challenge in itself. In light of these claims, this work proposes to use a fully decentralized ensemble of multiple GPs each making locally stationary models based on the training samples gathered.

Consider $m \in [1, 2, \ldots, M]$ GPs each with their respective set of observed nodes $O_m \subset D$, interpolated unobserved nodes $U_m \subset D$ along with their realizations $\mathbf{y}_k$ and $\mathbf{y}_k^*$ respectively. Then, each GP generates the following posterior for interpolated locations:

$$[\boldsymbol{\mu}_{U|O,\boldsymbol{\theta}}]_m \triangleq [K_*]_m [{K_\epsilon}^{-1}]_m ([\mathbf{y}]_m - [\boldsymbol{\mu}_O]_m)\,, \tag{3.9}$$

$$[\Sigma_{UU|O,\boldsymbol{\theta}}]_m \triangleq [K_{**}]_m - [K_*]_m [K_\epsilon^{-1}]_m [K_*^T]_m\,. \tag{3.10}$$

Deploying multiple GPs has the benefit of distributing the computational load. The reason being that now each GP expert only handles O_m observations to generate self-reliant posterior prediction over interpolated nodes. This is beneficial since the $\#(O_m) \ll \#(D)$ which translates to $\mathcal{O}(O_m)^3$ computational complexity and $\mathcal{O}(O_m)^2$ memory footprint. This makes the architecture amicable for robots operating with limited hardware capabilities. The only downside is the additional cost of fusion: Owing to a fully decentralized GP ensemble, multiple GP models will be obtained. In order to understand the overall dynamics of the environment, these models must be fused into a globally consistent model which would lead to additional computations which are borne by the base station. Additionally, the number of GP experts required to model the target phenomenon depends on human supervisor who usually sets the size of the robot team based on the project budget. Moreso, each robot acts as a self-sustaining GP expert and the initial arrangement of the team is assumed to be optimal.

3.5 Summary

This chapter presented the GPs customized to the scope of this work. The choice of kernel function used and the inference approach were explicitly discussed. Also, the posterior predictive form and model uncertainty were deduced for the spatial domain which come in handy for the rest of the work. Model uncertainty will be used for information acquisition and posterior distribution will be iteratively updated as more data is gathered. An extension to multi-GP setting was also presented.

Bibliography

[1] C. E. Rasmussen and C. K. Williams, *Gaussian processes for machine learning*, vol. 1. MIT press Cambridge, 2006.

[2] C. E. Rasmussen and C. K. Williams, "Gaussian processes in machine learning," *Lecture notes in computer science*, vol. 3176, pp. 63–71, 2004.

[3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[4] J. Chen, K. H. Low, Y. Yao, and P. Jaillet, "Gaussian process decentralized data fusion and active sensing for spatiotemporal traffic modeling and prediction in mobility-on-demand systems," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 901–921, 2015.

[5] N. D. Lawrence, "Gaussian process latent variable models for visualisation of high dimensional data," in *Advances in neural information processing systems*, pp. 329–336, 2004.

[6] J. Quiñonero-Candela and C. E. Rasmussen, "A unifying view of sparse approximate gaussian process regression," *Journal of Machine Learning Research*, vol. 6, no. Dec, pp. 1939–1959, 2005.

[7] T. Dick, E. Wong, and C. Dann, "How many random restarts are enough?,"

Chapter 4

Resource Constrained Path Planning with Homing Guarantee

> *The conservation of natural resources is the fundamental problem. Unless we solve that problem it will avail us little to solve all others.*

Theodore Roosevelt, *1907*

Chapter 3 introduced the model that explains the complex dynamics of the environmental phenomenon. This chapter serves to assist the previously mentioned model by endowing the robot(s) with the capability to autonomously plan paths through waypoints (informative locations) while paying heed to the resource constraints. As was exclaimed by President Theodore Roosevelt, ignoring the conservation of resources belittles the significance of solutions proposed for all other problem. Thus, here the author addresses the problem of resource constrained informative path planning (alternatively, resource constrained active sensing) by proposing a novel information acquisition function that can elegantly trade-off the quality of information gathered to the residual resources without significantly compromising on either.

Over the past decade, a lot of researchers have started to investigate the application of autonomous mobile robots to assist in large-scale enviroment monitoring [1–4]. The task at hand is challenging owing to complex dynamics, data-yearning models and resource constrained robots which often tend to conflict each other. Thus, the proposed algorithm aims to find the paths over space at a specific time[1] to be followed by a fully decentralized team of robots in order to best infer the spatially varying dynamics of the environment phenomenon. The target phenomenon to be observed is made available as the common knowledge to the entire team. Each robot behaves like a self-sustaining GP expert and must make decisions (independent of the peers) to select the most informative observations that can best enhance its model.

[1]Candidates are only evaluated using locations for the current time-step (interpolation). Continuous time path planning over both spatial and temporal domains have been investigated in [5] and can be used for forecasting.

4.1 Planning over Waypoints

Waypoint (*c.f. Definition 4.1*) based path planning for environmental monitoring applications works under the assumption that the time required for a robot to finish its mission is comparatively smaller than the time required by the field dynamics to evolve. The reason why such an assumption is critical is that, if the field dynamics were to update faster than the robot motion, then the robot would rather chose to stay in its current location as it would present novel information across time. In light of this assumption, for any particular time-slice t_i, the spatiotemporal waypoint selection for a single robot case could be represented like Fig. 4.1. In this setting, the set of locations chosen to be observed for one time-slice become independent of those chosen for any other. Similarly, interpolated observations at one time-slice are also independent of those for any other.

Definition 4.1 (Waypoint)**.** *An intermediate goal position that must be reached in order to gather observation for updating the GP model. Observations can only be gathered from way points and the current robot positions. Of all the potential candidates that may be chosen as the waypoint, only those that return the maximal reward are selected where reward is evaluated in terms of amount of information accrued.*

As previously discussed in Chapter 2, informative path planning can be broadly categorized into either *Density based* or *Hotspot based* approaches. For the scope of this work, the author focuses only on density based information acquisition functions which are detailed below. For this, the following notational convention will be used: red $\mathbf{x}^*$ represents the locations in the set $U \subset D$ which represent the set of unobserved locations over which predictions need to be made; blue $\mathbf{x}$ represents the list of visited locations in $O \subset D$; green $\mathbf{x}>$ represents the candidate locations which can be observed next while $\hat{\mathbf{x}}$ represents the optimal *next-best-location*. Similarly, red $\mathbf{y}^*$, blue $\mathbf{y}$ and green $\mathbf{y}>$ represent the realized measurements respectively.

4.1.1 Entropy Maximization (full-DAS)

This acquisition function drives the robot to follow the gradient of information to sample observations from areas with high uncertainty or least knowledge. Mathematically, this is defined by harnessing the entropy of GP as:

$$\hat{\mathbf{x}}_F = \underset{\mathbf{x}>}{\arg\max} \ \left\{ \mathbb{H}_{\mathbf{y}>|\mathbf{y},\boldsymbol{\theta}} \right\} \tag{4.1}$$

From hereon, Eq. (4.1) will be referred to as *full-DAS* wherein *DAS* stands for Decentralized Active Sensing. This objective function is designed for a team of disconnected and decentralized multi-robot team which can be used to obtain the purely explorative behavior. However, one major drawback of such an objective function is that the robot tends to visit far off locations thereby incurring prohibitively high travel costs [6].

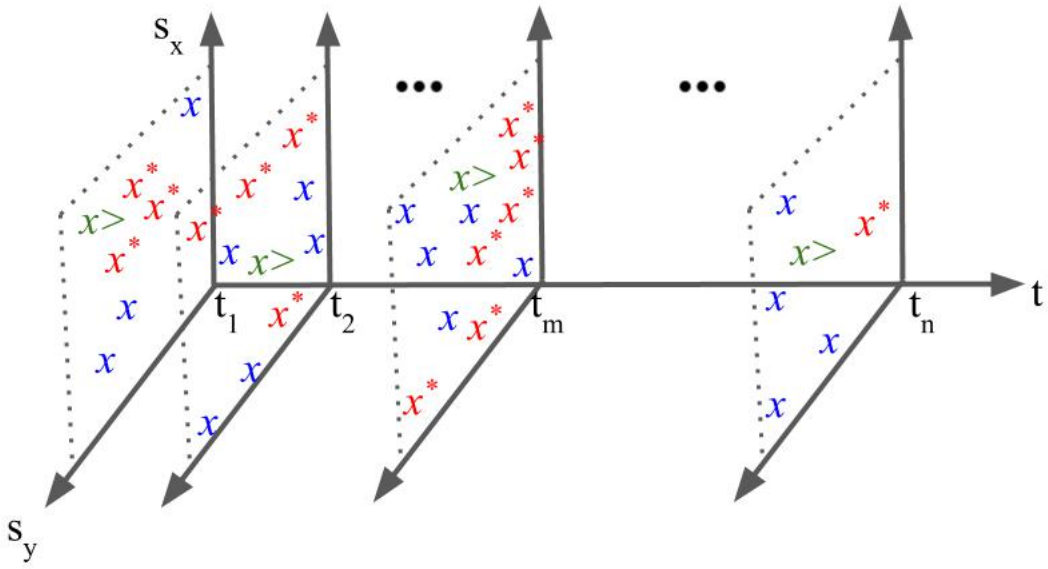


Figure 4.1: Planning over waypoints. llustrating waypoint selection and interpolation mechanism and generalizing over multiple time-steps. The objective of this waypoint path planning is to do interpolation i.e., making accurate predictions at locations marked with red x^*. All the visited locations are shown with blue x and the next-best-location i.e. green $x>$ is iteratively chosen. Interpolation across one time-slice t_i is independent of that across $t_j; i \neq j$.

4.1.2 Nearest Neighbor (NN)

A naïve solution to remedy the extreme travel costs incurred by *full-DAS* could be to strictly focus on thrifty utilization of resources. In other words, this means to visit the nearest neighbors which leads to minimal travel costs but from Tobler's first law [7] it is evident that this leads to acquisition of highly correlated observations. Simply put, high correlation translates to redundant or un-informative observations thereby incurring high resource wastage and poor model quality.

$$\hat{\mathbf{x}}_N = \arg\min_{\mathbf{x}>} \{\ln\|\mathbf{x} - \mathbf{x}>\|\} \tag{4.2}$$

4.1.3 Resource Utilization Efficacy Amelioration while maximizing Information gain

In order to avoid incuring extreme travel cost and also refrain from gathering highly correlated observations, this work proposes a novel acquisition function. The new acquisition function trades-off the quality of observations (high entropy) to resource optimization (low travel cost) using a bi-objective optimization framework. In [8] the author proposed a novel bi-objective optimization framework which is referred to as *Resource Constrained Decentralized Active Sensing (RC-DAS)*. This approach can effectively trade-off model quality to travel distance without compromising the model performance significantly and bounds the travel costs. Essentially this problem is difficult to solve owing to the conflicting nature of the objectives: GP models are highly data-driven models *i.e.,* the larger the amount of training samples, the better is the predictive performance. But in order to acquire large amounts of "informative" training samples, the robot would incur excessive travel costs. Thus, the challenge is to optimize over both the objectives simultaneously without significantly compromising on either. This can be done elegantly using the proposed objective function:

$$\hat{\mathbf{x}}_R = \arg\max_{\mathbf{x}>} \left\{\alpha\, \mathbb{H}_{\mathbf{y}>|\mathbf{y},\boldsymbol{\theta}} - (1-\alpha) \ln\|\mathbf{x} - \mathbf{x}>\|\right\} \tag{4.3}$$

In Eq. (4.3), the model performance ($\mathbb{H}_{\mathbf{y}>|\mathbf{y},\boldsymbol{\theta}}$) and travel cost ($\ln\|\mathbf{x} - \mathbf{x}>\|$) are amalgamated into one bi-objective optimization routine using the weight α. The weight α handles the exploration-exploitation trade-off. The approach is scalable to a multi-robot setting as is summarized in Alg. 2.

Algorithm 2 RC-DAS (D,B)

1: **Input**:
- $\{\mathbf{x}\}_{m=1}^{M} \leftarrow \mathbf{x}_m^{[1]}$
- $\{\mathbf{y}\}_{m=1}^{M} \leftarrow NULL$
- $\{O\}_{m=1}^{M} \leftarrow NULL$

2: **Output**:*Next-best-location* $\hat{x}_R$
3: **for** `agent` $m = 1, \ldots, M$ **do**
4: **while** $B > 0$ **do**
5:
6: /***SENSE***/
7:
8: $y \leftarrow Sense(x_m)$ ▷ obtain measurement
9: $\mathbf{y}_m \leftarrow [\mathbf{y}_m, ; y]$ ▷ store observation
10: $O_m \leftarrow [O_m; x_m]$ ▷ store location
11:
12: /***PLAN***/
13:
14: $\hat{\boldsymbol{\theta}}_{opt} \leftarrow MLE(\mathbf{y}_m, O_m)$ ▷ obtain hyper-parameters
15: ▷ deduce most uncertain locations
16: $\mathbf{x} >_m \leftarrow CalcUncertainNeighbors(D, x_m)$
17: ▷ Compute predicted measurements
18: $\boldsymbol{\mu} >_m, \Sigma >_m \leftarrow CompPosterior(\mathbf{z}_m, O_m, \mathbf{x} >_m, \hat{\boldsymbol{\theta}}_{opt})$
19: ▷ RC-DAS Objective Function
20:

$$Obj \triangleq \left(\alpha\, \mathbb{H}_{\mathbf{y}>|\mathbf{y},\boldsymbol{\theta}} - (1-\alpha) \ln(D_{Hav}(\mathbf{x} - \mathbf{x}>)\right)$$

21: ▷ optimal *Next Best Location*
22: $\hat{x}_R \leftarrow \arg\max_{\mathbf{x}>}(Obj)$
23:
24: /***ACT***/
25:
26: ▷ pass target location to robot controller
27: $x_m \leftarrow MoveToNextBestLoc(\hat{x}_R)$
28: $B \leftarrow B - (S + T)$ ▷ update remaining budget
29: **end while**
30: **end for**

In Alg. 2, when the robot is at a certain location x_m and obtains a sensor measurement y, the observation (line 9) and input location (line 10) are stored for inference (line 14). Then, the most uncertain neighbors surrounding the current robot location are evaluated, which are within accessible limits of the robot (line 16). Following suit, the posterior prediction over these locations as shown in line 18 is deduced. To evaluate the *most informative next-best-location*, the proposed cost

function from Eq. (4.3) is evaluated to optimize the travel distance and simultaneously reduce the prediction uncertainty (line 20). Upon jointly maximizing over this cost function, the feasible *next-best-location* as shown in line 22 can be obtained which is then set as the current goal position to be attained by our mobile robot. In line 28, the remaining budget B is updated by subtracting the *Sensing cost* (S) and *Travel cost* (T) incurred as a result of motion to the new location $\hat{x}_R$. For the purpose of evaluations, the *Sensing cost* (S) and *Travel cost* (T) have been defined by the author in Definitions 4.2 and 4.3 respectively.

Definition 4.2 (Sensing cost)**.** *The cost incurred by the robot to gather a measurement. This involves the cost of operating the sensor, processing the measurements, heat-losses from sensors (if any), etc. This cost is usually static and in this work, will be defined as:*

$$C_S(x) = \arg\min_{\forall x'} ||x - x'|| \tag{4.4}$$

The intuition behind such a choice of sensing cost was that irrespective of how good the sensor models and energy dissipation models are, the robot will always consume energy (E); $E \geq \arg\min_{\forall x'} ||x - x'|| + \delta$ where the term δ accounts for all the other aforementioned ancillary losses incurred while operating the sensor.

Definition 4.3 (Travel cost)**.** *The cost incurred by the robot to move from its current location to the next-best-location for gathering the measurement. This cost encompasses the motor losses, aerial drag losses, losses owing to unforeseen environmental conditions, battery energy consumed to move from one waypoint to another, mechanical losses, etc. This cost is highly dynamic and accurately modeling it is challenging. For the scope of this work, the following definition of travel cost would suffice:*

$$C_T(x, x') \triangleq ||x - x'|| \tag{4.5}$$

4.1.4 Comparative Analysis of Information Acquisition Functions

The weight factor α used in *RC-DAS* can be adapted to change its behavior. When $\alpha \to 0$, the path planner behaves like *nearest neighbor* and when $\alpha \to 1$ the same behaves like *full-DAS*. A comparative analysis of all 3 acquisition functions is visually aided by Fig. 4.2. Besides, the computational complexity of *full-DAS/RC-DAS/NN* are all given by $\mathcal{O}(MN)$ for a team of M robots evaluated over N nodes spread across the domain D.

full-DAS is highly explorative in nature owing to which it gathers a handful of highly "informative" training samples whilst *nearest neighbor* is highly exploitative forcing the robot to gather highly correlated (un-informative) observations. *RC-DAS* is elegantly positioned between the former 2 path planning approaches delicately handling the quantity and quality of observations being accrued. As was discussed earlier, GPs are highly data-driven models. So, neither scanty amounts of high quality observations nor excessive amounts of redundant information would be of any significance.

4.2 Homing

Despite considering the resource constraints in *RC-DAS* acquisition function, there remains one severe lacuna yet *i.e.,* the robot tends to use its resources conservatively but this does not guarantee that the robot would return to the base station (home) at the end of its mission. This is referred to as "homing" and if not taken care-off could simply lead to the robot getting immobilized amidst the mission. To this end, a dynamic weight selection mechanism is proposed which not only updates the weights of the objective functions of *RC-DAS* objective function as the resources are being used but also explicitly takes the homing issue into consideration.

Definition 4.4 (Homing)**.** *The ability of a robot to successfully execute a trajectory such that it is always guaranteed to return to the base station (hardware failures not accounted for) is referred to as homing. Homing is essential for* 2 *reasons:*

- ***Reduce Phase:*** *For the next phase of the architecture (details in Chapter 6) wherein all models generated by various robots are to be fused into one globally consistent model.*
- ***Averting Immobilization:*** *If a robot is immobilized amidst the field owing to complete expenditure of the allocated resources, then, additional robots may need to be deployed to retrieve the stranded agent. Retrieval of the standed agent is important to obtain the model generated by the agent but doing so with the help of additional agents will increase the overall project cost and delay further deployments.*

4.2.1 Dynamically Choosing Weights for Optimization

Thus far, static weights were chosen for *RC-DAS* acquisition function but an apt choice drastically modifies the behavior of the robot. This can be attributed to the fact that selection of the weights biases the robot to either pay heed to model quality or resource utilization or both. Thus, in what follows, an optimal weight (α) selection mechanism is devised such that the weights are dynamically updated as a function of residual budget. This is done as follows:

$$\alpha^{[t]} \leftarrow \frac{B_{res}^{[t]}}{B}\,. \tag{4.6}$$

where,

$$B_{res}^{[t]} \triangleq B_{res}^{[t-1]} - (C_S^{[t]}(x >) + C_T^{[t]}(x, x >)) \tag{4.7}$$

In Eq. (4.6), the current *weighting factor* ($\alpha^{[t]}$) is determined based on the current residual budget ($B_{res}^{[t]}$) as defined in Eq. (4.7). In Eq. (4.7), the current residual budget ($B_{res}^{[t]}$) is defined as the difference between the previously available budget $B_{res}^{[t-1]}$ and the cost that was incurred in moving and gathering measurement in the previous time step $(t-1)$ such that $0 < B_{res}^{[t]} \leq B$. The current location of the robot is given by x and the next location being evaluated is given by $x >$. At $t = 0$, $B_{res}^{[0]} \triangleq B$ such that

$\alpha^{[0]} \leftarrow 1$. When plugging in apt weights (α) from Eq. (4.6) into Eq. (4.3), not only can the *next-best-location* be found, but also, explorative and exploitative nature of the objective function are also adjusted in accord with the residual resources.

4.2.2 Additional Homing constraints

By adapting the weights of the components as the resources are being depleted, the robot can effectively use the available resources to observe maximal part of the field however, this still suffers from one major loop-hole: the robot cannot return to base since it does not account for homing while planning paths. To remedy this, homing is posed as additional constraints while planning the trajectory and this revised cost function will now be addressed as RC-DAS† [9]. It looks like follows:

$$\begin{aligned} \hat{\mathbf{x}}_R = \underset{\mathbf{x}>}{\arg\max} \left\{ \alpha\, \mathbb{H}_{\mathbf{y}>|\mathbf{y},\boldsymbol{\theta}} - (1-\alpha) \ln \|\mathbf{x} - \mathbf{x}>\| \right\} , \\ \text{s.t.} \underset{\mathbf{x}>}{\arg\min} \{ C_T(\mathbf{x}, Home), C_T(\mathbf{x}>, Home) \} \end{aligned} \tag{4.8}$$

In Eq. (4.8), the additional constraint checks if the robot has enough resources to execute the trajectory to reach the chosen candidate and still return to base otherwise, the exploration is immediately terminated and the robot returns to base right away.

4.3 Experiments

The aforementioned information acquisition functions were empirically evaluated on a significantly large scale spatiotemporal environment monitoring dataset which is publicly available.

4.3.1 Dataset

The **US Ozone Dataset** was used. This dataset includes ozone concentrations (in parts per billion) collected by US Environmental Protection Agency [8]. In this dataset, the measurements were recorded for several years at 59[2] static monitoring stations across USA but only one of the years were chosen for evaluation of interpolation performance. For each station, the annual average ozone concentration was assigned as the sample measurement for that station. A high speed robot was simulated wherein the speeds and mechanical capabilities of the robots were assumed befitting to the requirements of covering extensive distances within small time spans (of the order of seconds).

4.3.2 Analysis without Homing Guarantees

The first set of experiments focuses on analysing the acquisition functions *viz., RC-DAS* pegged against *full-DAS* without any homing guarantees. For this, a pre-determined

[2]after removing missing entires.

value of $\alpha = 0.7$ was chosen for *RC-DAS*. This meant that irrespective of the residual budget, 70% importance was given to the model quality while only 30% importance to resource utilization. In order to evaluate the path length incurred and model quality attained for this parameter setting, the following experiment setup was designed: 4 robot decentralized team was considered to explore and generate models of **US Ozone Dataset**. To avoid solving the collision avoidance issue, the robots were restrained within arbitrarily demarcated zones and were allowed to explore using *full-DAS, NN* and *RC-DAS* active sensing schemes as shown in Fig. 4.3. As is visually apparent, *NN* trajectory is the longest with small-step increments while *full-DAS* trajectories are short with long-step increments driving the robots significantly far away. The average model accuracy and path lengths incurred by the team are summarized in Table 4.1.

Table 4.1: Analysing performance without homing.

Item	full-DAS	RC-DAS	NN
Average RMSE	23.671 ± 3.526	$\mathbf{19.3259 \pm 1.820}$	21.128 ± 7.034
Average Path Length (Kms)	9.5 ± 3.879	$\mathbf{16.5 \pm 4.319}$	17.1 ± 4.346

From Table. 4.1, it was concluded that the average path costs incurred by *full-DAS* is the least while that of *NN* is the maximal which is in accord with Fig. 4.2. Owing to scanty amounts of highly informative samples, the *full-DAS* model cannot attain optimal performance and neither can *NN* owing to hoards of correlated observations. Thus, *RC-DAS* is elegantly placed in the middle with optimal model quality and bounded path length.

It is important to reiterate here that, these results were obtained for a fixed $\alpha = 0.7$ but in reality, a robot will not be able to substain this highly explorative nature as the resources are being depleted. Thus, in the following, the impact of homing and dynamic weight updates will be evaluated. Also, since *NN* avails no benefits in terms of model quality, only *full-DAS* and *RC-DAS*† were evaluated.

4.3.3 Analysis with Homing Guarantees

For this set of experiments, *RC-DAS*† was compared to *full-DAS* whereby the former inherently ensures homing while the latter was artificially enforced to do so. The analysis is split into two segments: *First,* the analysis of trajectories is carried out followed by analysis of model quality hence generated.

4.3.3.1 Path Cost analysis with Homing Enforced

Firstly, the length of walk *i.e.,* number of locations observed by a single robot starting from multiple start locations is analyzed. The results are shown in Fig. 4.4, wherein, each trend represent a random starting location such that each of the available 59 locations were chosen as start location at least once.

From Fig. 4.4, it can be deduced that *RC-DAS*† is highly conservative in utilizing the available resources and hence can allow the robots to observe more locations. The length of walk of *RC-DAS*† is significantly larger than *full-DAS*. Besides, this also

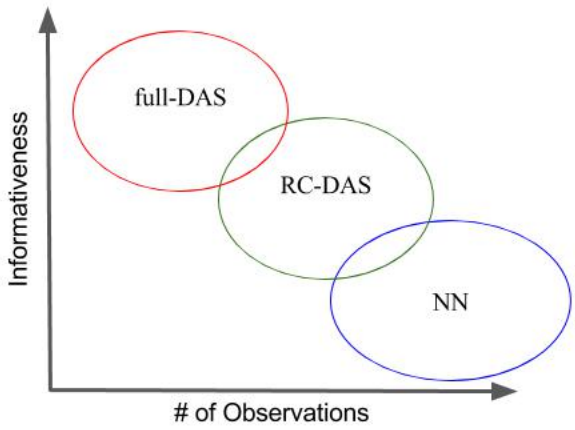


Figure 4.2: Comparing acquisition functions. Comparing the information acquisition function on a spectrum of highly informative to highly resource constrained options.

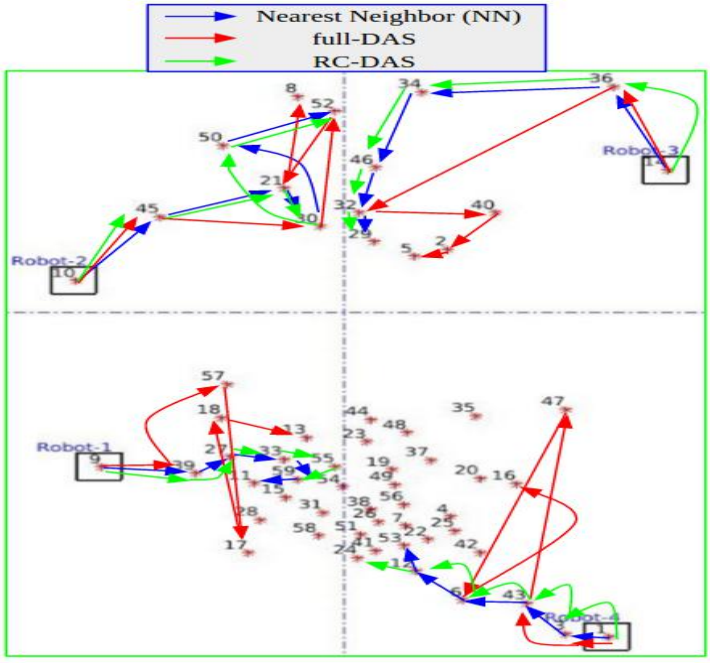


Figure 4.3: Trajectory without homing. Illustrating trajectories of 4 robots utilizing full-DAS, NN and RC-DAS active sensing schemes. To avoid collisions, the robots were restrained within pre-allocated sensing zones as is demarcated with dashed lines.

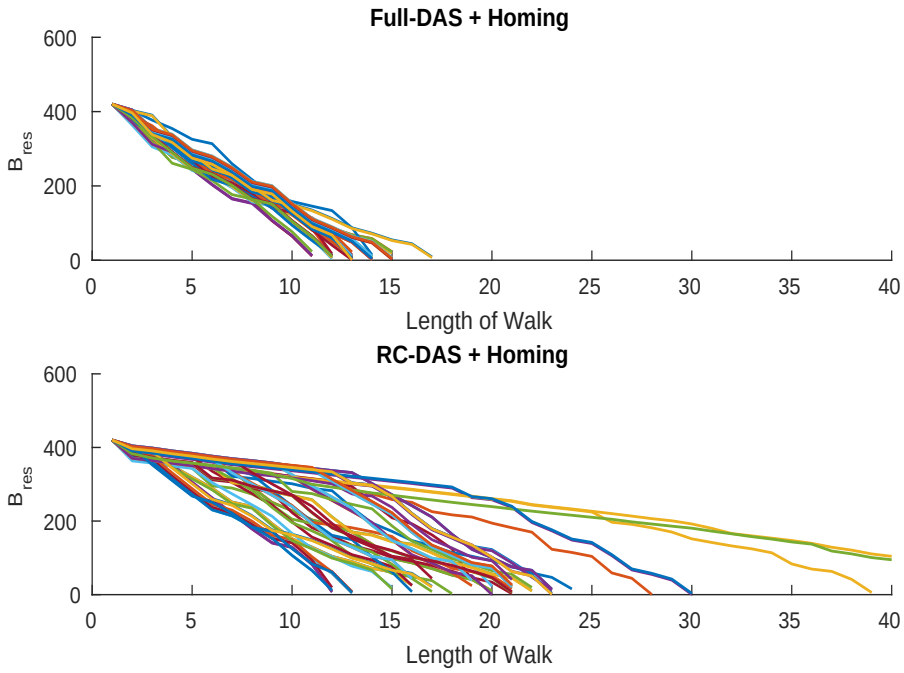


Figure 4.4: Budget decay. Analyzing how the budget is consumed (decayed) while gathering observations using the full-DAS and RC-DAS† active sensing schemes. Tests are performed for artificially enforced homing constraints using full-DAS. Each trend represents budget decay for the respective scheme for a chosen starting point.

proves the invariance of the *RC-DAS*† to the chioce of start location. This is important because the information available, guides the robot along the steepest gradient and if the robot is adversely affected by the choice of the start location, then convergence cannot be guaranteed.

4.3.3.2 Model Quality Analysis with Homing Enforced

In what follows, the model quality was analyzed when using *full-DAS* and *RC-DAS*† to ensure that imposing homing guarantees does not compromise the model quality significantly. Before doing so, a few evaluation criteria are defined in Definitions 4.5 and 4.6.

Definition 4.5 (Model Performance)**.** *The model performance when using a chosen active sensing scheme is defined as the Root Mean Squared Error (RMSE) over the predicted measurements for $\forall u \in U$ for a robot. Lower the RMSE, the better is the model performance and hence more accurate is the map.*

Definition 4.6 (Precision (P))**.** *If a total of N experiments are performed during which N_f represents the number of times full-DAS generated a more accurate map than RC-DAS and N_R represents vice versa, then precision (P) for full-DAS is given by:*

$$P_F \triangleq \frac{N_f}{N}\,, \tag{4.9}$$

and the precision (P) *for RC-DAS*† *is given by:*

$$P_R \triangleq \frac{N_R}{N}\,. \tag{4.10}$$

Thus, P *represents the chances of generating a better model*[3] *of the environment given the choice of active sensing scheme. The model accuracy can be evaluated by comparing the predicted measurements with respect to the ground truth values and evaluating the RMSE to associate a scalar value as a uniform performance measure for the model being considered.*

In this set of experiments, the precision of *RC-DAS*† was compared against *full-DAS*. All possible start locations were considered and the average performance is reported in Table 4.2. Since *RC-DAS*† always considers homing, two subsets of experiments needed to be performed: *Firstly,* considering *full-DAS* in its current form i.e. without homing and *Secondly,* by manually enforcing homing on *full-DAS*.

Table 4.2: Impact of Homing on the Precision of full-DAS vs RC-DAS†*.*

	P_F	P_R
Full-DAS w/o Homing	**63.33%**	36.67%
Full-DAS with Homing	45%	**55%**

From Table 4.2, it is evident that *full-DAS* has a higher precision when homing is not performed but owing to homing constraints the proposed *RC-DAS*† has superior performance. Alternatively, this also states that when homing is a necessary requirement, then the *RC-DAS*† is more robust to the choice of start location assigned to the robot. Since, the start locations directly affect the trajectory and the terminal quality of the prediction model, robustness to the choice of start location is of utmost importance.

4.4 Summary

A novel class of acquisition function belonging to the *density based* acquisition family was discussed that allows the robot to acquire information from the most uncertain areas whilst placing strict bounds on the net path length, guaranteeing homing and not compromising on the model performance significantly. The acquisition function is meant for and scales with the size of the team of mobile robots that may be operating in communication devoid environments in a fully decentralized fashion. The results show a significant reduction in net path costs whilst attaining similar or even better precision as shown by state-of-the-art acquisition function *viz., full-DAS*. Besides this, robustness to start location was also empirically validated which is of utmost importance when robot set to venture out in unknown environments. Since, the information acquisition functions in consideration follow the gradient of information,

[3] *c.f.* Definition 4.5.

if the robot starts from the least informative locations (which cannot be known a priori), it may drastically affect the model quality but *RC-DAS*† can efficiently handle such scenarios.

Bibliography

[1] M. Dunbabin and L. Marques, "Robots for environmental monitoring: Significant advancements and applications," *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 24–39, 2012.

[2] T. Wilson and S. B. Williams, "Active sample selection in scalar fields exhibiting non-stationary noise with parametric heteroscedastic gaussian process regression," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pp. 6455–6462, IEEE, 2017.

[3] A. Singh, F. Ramos, H. D. Whyte, and W. J. Kaiser, "Modeling and decision making in spatio-temporal processes for environmental surveillance," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 5490–5497, IEEE, 2010.

[4] J. Chen, K. H. Low, Y. Yao, and P. Jaillet, "Gaussian process decentralized data fusion and active sensing for spatiotemporal traffic modeling and prediction in mobility-on-demand systems," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 901–921, 2015.

[5] R. Marchant and F. Ramos, "Bayesian optimisation for informative continuous path planning," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 6136–6143, IEEE, 2014.

[6] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *Journal of Machine Learning Research*, vol. 9, no. Feb, pp. 235–284, 2008.

[7] N. Waters, "Tobler's First Law of Geography," *The International Encyclopedia of Geography*, 2017.

[8] K. Tiwari, V. Honoré, S. Jeong, N. Y. Chong, and M. P. Deisenroth, "Resource-constrained decentralized active sensing for multi-robot systems using distributed gaussian processes," in *2016 16th International Conference on Control, Automation and Systems (ICCAS)*, pp. 13–18, Oct 2016.

[9] K. Tiwari, S. Jeong, and N. Y. Chong, "Multi-uav resource constrained online monitoring of large-scale spatio-temporal environment with homing guarantee," in *Industrial Electronics Society, IECON 2017-43rd Annual Conference of the IEEE*, pp. 5893–5900, IEEE, 2017.

[10] R. Marchant and F. Ramos, "Bayesian optimisation for intelligent environmental monitoring," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 2242–2249, IEEE, 2012.

Chapter 5

Operational Range Estimation

> *Do not save what is left after spending, but spend what is left after saving.*
>
> Warren Buffet, *On Saving*

Erstwhile, the proposed algorithms for path planning were claimed to be bounded by the "budget", but it is very important to decipher what it means by this keyword and how the robot's actions lead to dissipation of the budget. Thus, this chapter discusses how to evaluate the maximum attainable range *i.e.,* operational range for a robot given some prior information about the mission characteristics and the robot itself (including the nature of the power source used to propel the robot). This is what is referred to as "budget" for this work. Accurate estimation of operational range plays a significant role in allowing the robot to plan close loop trajectories and prevent complete immobilization amidst the field.

5.1 Importance of Operational Range Estimation

Mobile robots are increasingly being deployed to assist in situations where human intervention is deemed risky or tedious. *E.g.,* actively pursuing evaders, patrolling sensitive areas like cross-country borders or high rise buildings (Fig, 5.1), exploring outer space or assisting in search-and-resuce (Fig, 5.2). In such time critical scenarios, the robots cannot abandon their posts for recharging while carrying out their respective missions.

Most robots have a rough estimation of their battery life based on the operation time [1–5]. Nonetheless, regardless of how the mission is carried out, robots must be retrieved when the operational time is close to the estimated maximum value, or the estimated remaining battery time is close to zero. However, this general approach neglects two facts: 1.) Missions are highly dynamic which in turn incur variable power consumption, thereby making the nominal estimated battery life time too broad, and

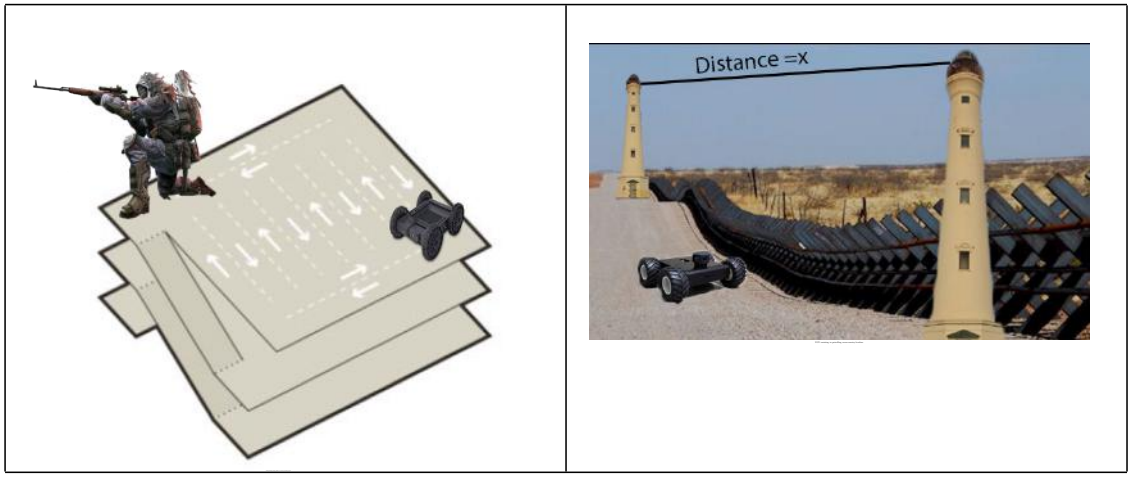


Figure 5.1: Simplified operational range estimation. On the left, a multi-level parking lot is shown which serves as a vantage point for armed forces to provide over-watch when a high valued individual visits a public spot. These spots once cleared for use by friendly armed personnel cannot be left unmanned. On the right, a border patrol scenario is shown where the robot must know its operational range in order to plan the patrolling missions between watch-towers.

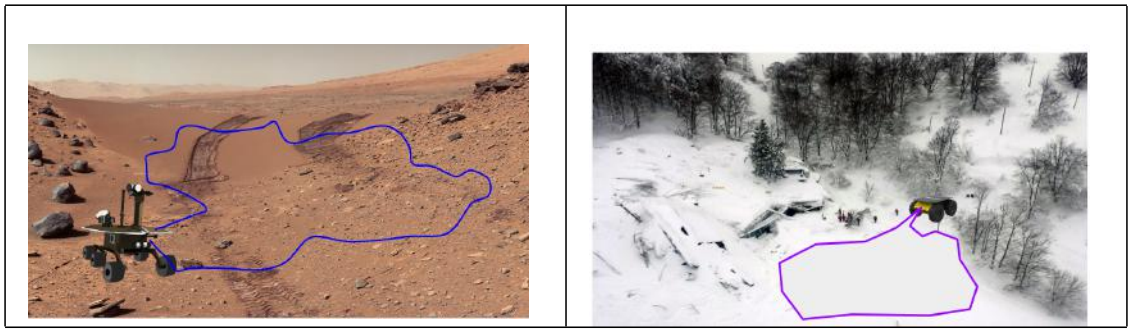


Figure 5.2: Real-world operational range estimation. On the left, a planetory rover exploring the surface of Mars is shown. On the right, a ground robot assisting in search-and-rescue after an avalanche is illustrated. In either case, the robot cannot abandon its post or its mission while it is actively surveilling. Thus, recharging is infeasible and the terrain is highly uneven and unpredictable.

especially too conservative. 2.) In outdoor/ indoor/ space exploration missions, other than the mission time, researchers are more concerned about the proportion of the unknown area that the robot covers. Not much research has been done to look into how the energy stored in the battery is distributed amongst different components and how this would affect the maximum traversal range.

Most of the aforementioned works consider the mission profile to be known a priori and then try to estimate/ optimize the energy requirements based on locomotion models only. As opposed to this, it is more rational to consider the energy itself to be fixed and then modulate the mission profile accordingly. The reason for such an unconventional "inverse" setting is that given the battery characteristics, the amount of energy stored within and the rate of discharge are known a priori and serve as the limiting factor for the mission, when the robots sets out to explore the environment. Thus, the inverse problem should be tackled where the energy constraint is known and the path length must be optimized within these constraints. The benefits of the knowledge about the maximal operational range based on a reasonable energy model can be harnessed for both tele-operated robots and fully autonomous robots. This means that either the operators can know the optimal time to retrieve the robots for a tele-operated robot or in case of an autonomous robot, the robot itself can gauge the best time to return to base. Using too much of the provided battery energy could lead to complete failure of the robots amidst the mission and using too less of the resources could significantly reduce the environmental perception.

In this chapter, two important keywords are introduced: **framework** and **model**. Since these cannot be used interchangeably, the author clearly defines them in Definitions 5.1 and 5.2 below.

Definition 5.1 (Framework)**.** *The pandora's box which takes in the description of the robot and the power source being used in order to deduce the maximum attainable operational range. Only provides a theoretical upper bound on the maximum range and does not include a path planner to assist with homing.*

Definition 5.2 (Model)**.** *Intrinsic component of framework that serves a specific purpose. In order to deduce the operational range of a robot, several models need to be utilized so that the necessary results can be coupled to deduce the maximum attainable range. The inclusion of all necessary modules gives rise to a framework.*

5.2 Rationale behind the Maverick Approach

The existing models for endurance [6] or mission energy estimation [7] are made specific to the robot under consideration. Furthermore, the author believes that there exists almost no framework to estimate the operational range of robots functioning on a single discharge cycle let alone be comprehensive enough to estimate the operational range for any category of robot. This is the philosophy behind the proposed operational range estimation architecture which not only considers the locomotion model along with the ancillary functions which tend to draw power from the same source, but, can also accommodate any robot while retaining the same composition.

In order for the robot to know its maximal operational range, it is very important that an *energy dissipation model* is developed which elucidates how the energy from the power source is distributed across the system along with the uncalled for losses owing to heat dissipation, friction *etc.* This chapter, takes into account multiple such energy consumption sources and analyzes how the energy stored in the battery is consumed during one single discharge cycle. The research problem addressed here is:

> *Given a robot equipped with a battery, what is the maximum attainable range on a single discharge cycle to avoid complete immobilization?*

The task at hand is challenging owing to the fact that different robots have distinct motion models and usually operate in diverse environmental conditions which cannot be anticipated in advance. Moreover, making a generic framework that can help estimate operational range for a multitude of robots is useful but non-trivial.

5.3 Work Flow

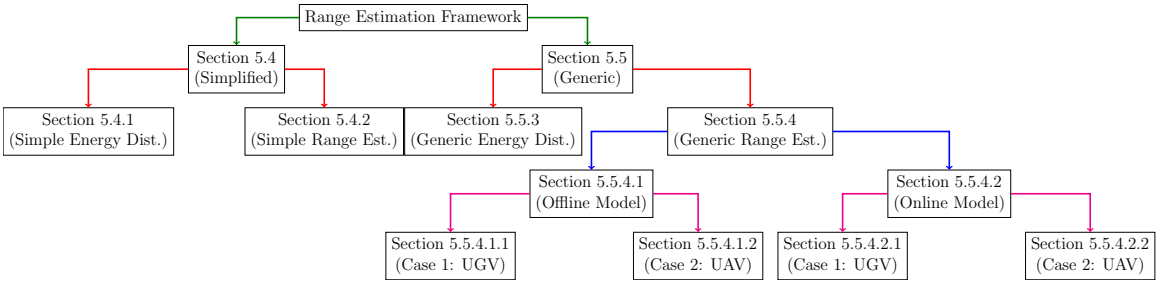


Figure 5.3: Work flow for operational range estimation.

For the ease of the readers, this section is meant to provide a roadmap to the sections as they follow hereon. The organization of the rest of this chapter is shown in Fig. 5.3 and is detailed as follows: The overall range estimation framework is categorized into 2 types:

- *Simplified Framework* meant for simple environmental conditions focusing only on UGVs is first presented in Section 5.4. Within this section, a simplified energy distribution model with corresponding losses is accounted for in Section 5.4.1 along with the corresponding range estimation model presented in Section 5.4.2.

- *Generalized Framework* which is extension of the previous framework meant to account for various classes of robots. This is discussed in Section 5.5. As before, the enhanced energy distribution model is presented in Section 5.5.3 followed by the enhanced range estimation model in Section 5.5.4. In order to estimate the range using the generic framework, an *Offline* variant is presented in Section 5.5.4.1 followed by its *Online* counterpart as discussed in Section 5.5.4.2. For each of these models, two case studies are presented using UGVs and UAVs in Sections 5.5.4.1.1, 5.5.4.1.2, 5.5.4.2.1, 5.5.4.2.2 respectively.

5.4 Simplified Range Estimation Framework for UGV

This section presents a simplified framework focusing only on UGVs operating on smooth terrains and smoothly varying elevations like those shown in Fig. 5.1 [8]. A framework is said to be comprised of two components *viz.,* 1.) Energy distribution model which explains how the energy is consumed across the system and 2.) Range estimation model which transforms the useful energy into attainable distance.

5.4.1 Energy Distribution Model

Consider a rudimentary robot equipped with only a power source and motors for propelling the robot. In such a setting, the energy from the battery is directly used for maneuvering as shown in Fig. 5.4. However, almost never does a robot have such limited payload. A robot that is being used for exploration of unknown environment be it search-and-rescue or patrolling, always has sensors, microcontrollers and other essential payload deemed necessary for the respective mission. In such a setting, energy from the battery is distributed amongst the maneuvering and ancillary branches and also sustains losses as shown in Fig. 5.5.

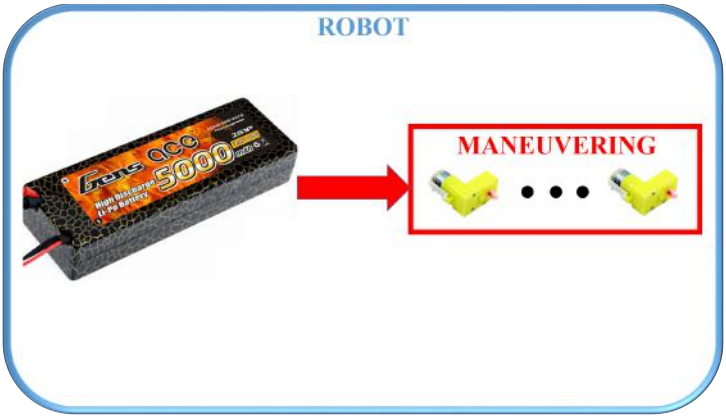


Figure 5.4: Ideal battery dissemination model. All energy stored in battery is used for maneuvering without any losses.

There are 4 kinds of losses associated with the entire robotic system which, in turn, affect the maximum attainable range of a mobile robot. They are:

- **Battery charge storage loss** (η_1)**:** refers to the battery self-discharge characteristics. Even without any load attached, the battery tends to suffer self-discharge thereby reducing the net amount of energy available for a mission.
- **Drive motor losses** (η_2)**:** owing to internal friction along with actuation losses.
- **Mechanical losses** (η_3)**:** refers to power train losses like friction in transmission, damping from lubricants *etc.*

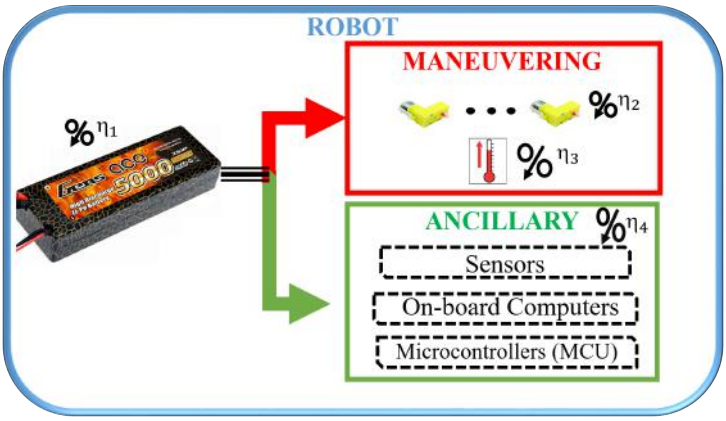


Figure 5.5: Realistic lossy battery dissemination model. Energy from battery is distributed amongst the maneuvering and ancillary branches. The sustained losses are marked herewith.

- **Ancillary losses** (η_4)**:** accounts for heat losses incurred by sensors, motor drivers, micro-controllers *etc.*

So, the overall system efficiency can be summarized as $\Omega \triangleq \Pi_{i=1}^{4} \neg\eta_i$ and the procedure for calibration to obtain these losses is explained next.

5.4.1.1 System Identification

The procedure of calibrating the system to deduce the 4 kinds of losses mentioned above is referred to as *System Identification*. For this, a *minimal load test* was designed in which the only load drawing power from the source were the motors. During this test, the motors were allowed to run until the battery was completely drained and the following 3 quantities need to be calculated in order to identify system losses:

- **Internal Friction Power** (P_{IF})**:** is the power required to overcome the internal friction offered by the motors during the *minimal load test* which lasted for T_{test}. All the energy supplied from the battery to the motors (E_{motor}) is transformed into mechanical energy (E_{mech}) to turn the motors and motor losses lost as heat ($I_{motor}^2 R_{motor} T_{test}$) to the environment. Thus, the following relationship holds from the law of conservation of energy:

$$\begin{aligned} \underbrace{\tilde{E}_{motor}}_{\text{Net Energy for motors}} &= \underbrace{E_{mech}}_{\text{Mechanical Energy}} + \underbrace{I_{motor}^2 R_{motor} T_{test}}_{\text{Heat Loss during test}}, \\ \Rightarrow P_{IF} &= I_{motor}^2 R_{motor}, \\ &= \frac{\tilde{E} - E_{mech}}{T_{test}}. \end{aligned} \tag{5.1}$$

- **Field Trial Power** (P_{FT})**:** is the power consumed during the field trials which lasted for T_M. Besides the power required to overcome the internal friction of the

motors, additional power will now be consumed to overcome terrain resistance. Thus, the overall energy from source is now related to internal friction and terrain interactions as:

$$\underbrace{\tilde{E}_{motor}}_{\text{Net Energy for motors}} = \underbrace{E_{mech}}_{\text{Mechanical Energy}} + \underbrace{P_{IF}T_M}_{\text{Heat Loss during test}} + \underbrace{E_{terr}}_{\text{Terrain Interaction}} , \quad \Rightarrow P_{FT} = \frac{P_{IF}T_M + E_{terr}}{T_M} \tag{5.2}$$

- **Net Component Power** (P_{CP})**:** is the net energy consumed by all components on board including motors, sensors, controllers *etc.,* for a mission duration of T_M. This is given by:

$$P_{CP} \triangleq \underbrace{\frac{\tilde{E}_{motor}}{T_M}}_{\text{Motor Power}} + \underbrace{P_{IF}}_{\text{Motor Loss per unit } T_M} + \Sigma_{\forall sensor}\Big(\underbrace{P_{sensor}}_{\text{Sensor Power}} + \underbrace{I^2_{sensor}R_{sensor}}_{\text{Sensor loss per unit } T_M} \Big) \tag{5.3}$$

Now, the system losses can be calculated for the mission time T_M as:

- $\eta_1 \triangleq 100 \times \frac{\tilde{E}_{net} - P_{CP}T_M}{\tilde{E}_{net}}$ where $\tilde{E}_{net} \triangleq \tilde{E}_{motor} + \Sigma_{\forall sensor}\tilde{E}_{sensor}$.
- $\eta_2 \triangleq 100 \times \frac{\tilde{E}_{motor} - P_{IF}T_M}{\tilde{E}_{motor}} = 100 \times \frac{\tilde{E}_{mech}}{\tilde{E}_{motor}}$.
- $\eta_3 \triangleq 100 \times \frac{P_{FT} - P_{IF}}{P_{FT}}$.
- $\eta_4 \triangleq 100 \times \frac{\Sigma_{\forall sensor}(\tilde{E}_{sensor} - I^2_{sensor}R_{sensor}T_M)}{\Sigma_{\forall sensor}(\tilde{E}_{sensor})}$.

Here, the zeroth order polynomial have been used *i.e.,* the first order approximation of the efficiency/losses of the system to estimate its lower bound. However, if a more complex model (higher order polynomial) were to be used that perhaps can also account for mechanical degradation, changes in current demands owing to variable motor loads, elevation changes and operational velocity modulations *etc.,* better estimates can be obtained. An even higher complexity model could also track the changes in these parameters in real-time which can be used to account for system efficiencies in an *online* fashion and maintain tighter bounds. Having said this, the challenge still remains to identify such models and quantify their parameters. For the scope of this work, only the first order approximations were retained.

5.4.2 Simplified Operational Range Estimation

Having modeled the energy distributed across the entire system using Fig, 5.5, the maneuvering energy can now be transformed into operational range. For this, the free-body diagram of a robot on an elevated plain is drawn in Fig. 5.6 wherein all the forces in the state of equilibrium are shown along with the friction offered by

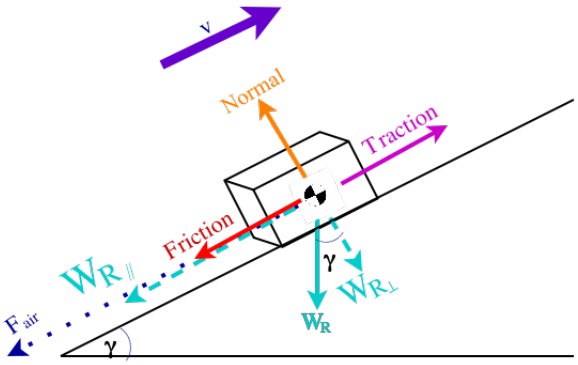


Figure 5.6: Free Body Diagram. Illustrating all forces that impede the motion of a robot. The cuboid shown represents the robot with all forces acting around its center of mass. $W_R(= m_R g)$ represents the weight of the robot. The weight components were decomposed into the parallel component $W_{R_\parallel}(= m_R g sin\gamma)$ acting along the terrain and the perpendicular component $W_{R_\perp}(= m_R g cos\gamma)$ acting against the $Normal(N)$ which represents the normal force. $Friction$ represents the surface friction offered by the ground and F_{air} represents the aerodynamic drag force.

the terrain itself which varies based on the surface type and wheel built. Also, the impact of aerodynamic drag force[1] was accounted for. From this figure, the following equilibrium conditions were deduced:

$$N = m_R\ g \cos\gamma\,.$$

$$\begin{aligned} Traction &= \ Friction + F_{air} + W_{R_\parallel}\,, \\ \Rightarrow Traction &= C_{rr}\ N + cv^2 + m_R\ g \sin\gamma\,. \end{aligned} \tag{5.4}$$

Thus, in a realistic setup, the energy needed for displacing the robot by an amount d on an elevated terrain can be given by:

$$\begin{aligned} ME &= Traction \times d\,, \\ &= (C_{rr}\ N + cv^2 + m_R\ g \sin\gamma)d\,, \\ &= (C_{rr}\ m_R\ g \cos\gamma + cv^2 + m_R\ g \sin\gamma)d\,. \end{aligned} \tag{5.5}$$

In Eq. (5.5), an elevation angle $\gamma \in [0, \gamma_{max}]$ was considered, such that increasing the velocity also increases the ME needed to attain the distance d. However, so far system efficiency parameters are yet to be examined. In Fig. 5.5, 2 main consumers of the battery energy were presented: *Firstly,* the **Maneuvering module** which accounts for traversal, steering etc. and *secondly,* the **Ancillary Functions module** which accounts for sensing, on-board computations, communication to peers (or

[1] Although, this factor has been considered to make the model realistic but in case of mobile robots, since the operational speed is of the order of few m/s, this factor can be neglected.

base station) etc. Both these consumers draw power from the same source which is transformed into useful energy along with unwarranted losses. From the works of [9], the power consumption model for sensors was obtained and extended to also account for energy consumed during computations and communications as:

$$P_{anc} = \underbrace{\{s_0 + s_1 f_s\}}_{P_{sensing}} + P_C \tag{5.6}$$

The advantage of using this power model for the ancillary branch is that, it can elegantly take care of situations when the sensor is idling or when it is actively gathering measurements. In Eq. (5.6), the terms in $\{\cdot\}$ refer to the power consumed for gathering measurements whereby f_s refers to the sampling frequency (Hz) which is contingent on the sensor type. *E.g.,* in case of laser range finder, sonars, ultrasonic sensors it could refer to the number of rays emitted per second whilst in case of a camera it could refer to the fps rate. Also, computation cost will only be incurred when sensor measurements are gathered. The term P_C accounts for two factors: *1.)* the power utilized by micro-controllers to command the wheels and sensors, *2.)* the power used by the on-board computation module. Since the micro-controller tasks are usually fixed, the author assumes a static power consumption which is stable [9] but the power consumption for computational requirements may vary based on the different tasks like *SLAM, localization, Occupancy Grid Mapping etc.* Thus, the P_C jointly accounts for the power consumed for computations and short-range wireless communications. However, if more complex models for architectural power consumption were deemed necessary, then, the readers are referred to other works like [10].

Motivated by the exponential battery discharge model from [11], this work also uses an exponential decay function to represent this trend using positive coefficients k_1, k_2 as:

$$\tilde{E} \triangleq E_O \exp^{-(k_1\zeta + k_2 t)} \tag{5.7}$$

In Eq. (5.7), as opposed to prior works, the author also considers the impact of several charge-discharge cycles (ζ) along with the age of the battery (t).

In order to estimate maximum achievable range, first, the total energy model in a real world setting is established as the sum of the Ancillary Energy (AE) and the Traversal Energy (TE)[2]

[2]Net Maneuvering Energy (ME) available *i.e.,* $\frac{ME}{^{UGV}\Omega_{Man}}$, where $^{UGV}\Omega_{Man}$ is the efficiency of Maneuvering branch.

$$
\begin{aligned}
\tilde{E} &= AE + TE\,, \\
&= Ancillary\ Power \times T_M + \frac{ME}{{}^{UGV}\Omega_{Man}}\,, \\
&= P_{anc} \times \frac{d}{vD} + \frac{(C_{rr}\ m_R\ g\cos\gamma + cv^2 + m_R\ g\sin\gamma)d}{{}^{UGV}\Omega_{Man}}\,, \\
&= d \times \left\{ \frac{P_{anc}}{vD} + \frac{(C_{rr}\ m_R\ g\cos\gamma + cv^2 + m_R\ g\sin\gamma)}{{}^{UGV}\Omega_{Man}} \right\}\,, \\
\Rightarrow d &= \frac{\tilde{E}}{\left\{ \frac{P_{anc}}{vD} + \frac{(C_{rr}\ m_R\ g\cos\gamma + cv^2 + m_R\ g\sin\gamma)}{{}^{UGV}\Omega_{Man}} \right\}}\,.
\end{aligned}
\tag{5.8}
$$

Now, in order to evaluate the theoretical maximum attainable range, the optimal operational velocity (v_{opt}) and reduced battery capacity need to be considered. Thus, the theoretical maxima is given by:

$$
d_{max} = \left\{ \frac{\tilde{E}}{\frac{P_{anc}}{v_{opt}D} + \frac{(C_{rr}\ m_R\ g\cos\gamma + cv^2 + m_R\ g\sin\gamma)}{{}^{UGV}\Omega_{Man}}} \right\} \tag{5.9}
$$

In Eqs. (5.8) and (5.9), an additional symbol D was utilized which stands for the *duty cycle*. Albeit the author assumes the constant operational velocity for carrying out the mission, the robot may sometimes get overwhelming/too sparse amounts of data (*e.g., Fig. 5.7a*) or may lose connection with the base station (*e.g., Fig. 5.7b*) for which it must stop and manage the situation. To allow the robot to do so, the term D is very important which represents the proportion of the net mission time which the robot spent for actually moving and covering ground. The term D additionally accounts for the fact that the ancillary power is consumed incessantly throughout the mission and as the robot stops more often *i.e.,* $D \downarrow$, the ancillary power $(AE) \uparrow$. Also, the theoretical upper bound on operational range i.e. d_{max} is calculated by using the optimal velocity v_{opt}. The choice of v_{opt} is rather challenging since this is determined by the safe operational velocity given the distribution of obstacles in the environment and environment conditions (nature of terrain, average elevation etc.). Thus, for the scope of this work, only the safe operational velocity is taken as the v_{opt} and this is determined by the human operator.

In Eq. (5.8), the ancillary power (P_{anc}) is computed with respect to the mission time which is calculated as the ratio of distance to the average speed (*i.e.,* velocity normalized by duty cycle, D; $T_M = \frac{d}{vD}$) while the maneuvering energy is computed with respect to the travel distance, d. Mechanical efficiency (Γ) is the ratio of the energy that is actually used to accomplish mechanical work to propel the robot forward to the total energy that actually goes into the maneuvering branch. It takes into

(a) Feature extraction for ORB-SLAM [12]

(b) Faulty communication channels

Figure 5.7: Need for introducing duty cycle (D). In Fig. 5.7a visual SLAM scenario is shown wherein sometimes not enough features may be available for the robot to localize itself based on the frames captured by the camera(s) and at some other occasions, overwhelming amounts of features may be extracted like in cases of extremely cluttered environments. In Fig. 5.7b, a common communication channel fault is shown whereby the robots may occasionally face technical difficulties while parsing messages to and from the base station. In either scenario, the robot needs to wait to recover and can only proceed when the problem has been resolved.

account the aforementioned losses η_2 and η_3: $\Gamma = (1 - \eta_2) * (1 - \eta_3)$. η_1 accounts for the energy loss before the battery output, which is embedded in Eq. (5.7). η_4 is the percentage of the battery output energy that goes into the ancillary branch: $AE = \eta_4 \hat{E}$. Despite the definition in Eq. (5.8), the overall system efficiency can be summarized as $\Omega \triangleq \Pi_{i=1}^{4} \neg\eta_i$ where $\neg$ represents the complement operator which is used to obtain the efficiencies from losses.

5.5 Generic Range Estimation Framework for Diverse Classes of Robots

The prior work [8] is now extended to not only account for environmental factors like variable and uneven terrains, strong wind gusts *etc.,* like those faced in real exploration missions shown in Fig. 5.2, but also to broaden the outreach of the operational range estimation framework to account for different classes of robots like UGVs and UAVs as shown in Fig. 5.8. Furthermore, *offline* and *online* variants for range estimation and an enhanced ancillary power model are presented and the empirical performance is discussed henceforth.

5.5.1 First Things First

Before presenting the unified framework [13], it is essential to clarify the difference between the terms "framework" and "model". This is aided by Fig. 5.8 wherein

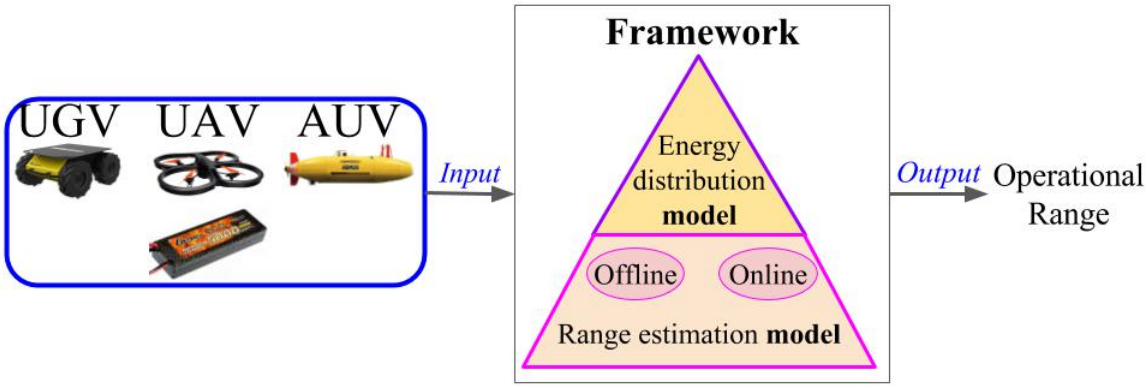


Figure 5.8: Framework v/s Model. ***Framework*** *refers to the entire architecture in unison which can be considered as a black box for operational range estimation. The nature of the robot and its power source are fed as inputs and the operational range is generated as output.* ***Model*** *refers to intrinsic components viz., the energy distribution model for inferring the distribution of energy in the system and range estimation model which convert the useful energy into operational range. On a high level, the black-box can account for various robot types while the differences occur only at the lower levels for adjusting to the distinct locomotion models. Hence, the notion of unification is justified.*

"framework" refers to the entire architecture which is applicable to all classes on robots on a high level and "model" refers to inherent modules that model the energy distribution of the system and propose a methodology to transform this into operational range. The only adjustment which must be made as a function of the robot type occurs at the lower level modules of *offline/online* estimators which take into account the variety of locomotion models. Thus, the notion of *unification* comes into play.

5.5.2 Enhancements over the Simplified Framework

In this section, the novel additions to the operational range estimation framework are discussed. So far, standalone researches have looked into development on analytical models for mission energy and time consumption of specific robotic platforms. Their main focus was to estimate either the endurance or the energy requirements for robots given a preset mission. Some models were made *offline* whilst others were generated *online* based on real-time operation data. However, the major limitation of these models was that none of them could estimate the maximal operational range of the robot given some a priori known information about the execution of the mission. Furthermore, premeditated trajectories were considered which are not feasible for real-world applications and robot-specific models were developed.

Thus, the aim now is to develop one global framework such that given the battery capacity of a robot which may or may not be further aided by some a priori known characteristics of the mission, the maximum operational range for any type of mobile robot can be estimated. Not only this, but a variety of unforeseen environmental factors like sudden changes in terrain elevation or wind gusts *etc.,* along with the flexibility to stop and process the data are now catered to. These factors inherently affect the maximum attainable range and avoiding any premeditated trajectories makes the framework better suited to pragmatic applications. Additionally, the ancillary power consumption model from Eq. (5.6) is now further revised to account for variable data traffic using wireless communication. For converting the rest of the useful maneuvering energy into operational range, 2 variants of range estimation model are also proposed and validated.

5.5.3 Energy Distribution Model for Diverse Robots

The upgraded energy distribution model of the unified framework is shown in Fig. 5.9: Irrespective of the nature of the robot, the energy available from the battery is always utilized by 2 kinds of processes *viz., maneuvering* and *ancillary functions*. The proportion of the energy used for the former is referred to as *traversal energy* (TE) while that of the latter is referred to as ancillary energy (AE). In an ideal situation, the net energy from the battery (E), the *traversal energy* (TE) and the ancillary energy (AE) are related as:

$$E = AE + TE \tag{5.10}$$

Based on the system losses as previously discussed in Section 5.4.1, the overall system efficiency of any robot (r) can be summarized as ${}^{r}\Omega \triangleq \Pi_{i=1}^{4} \neg\eta_i$. The

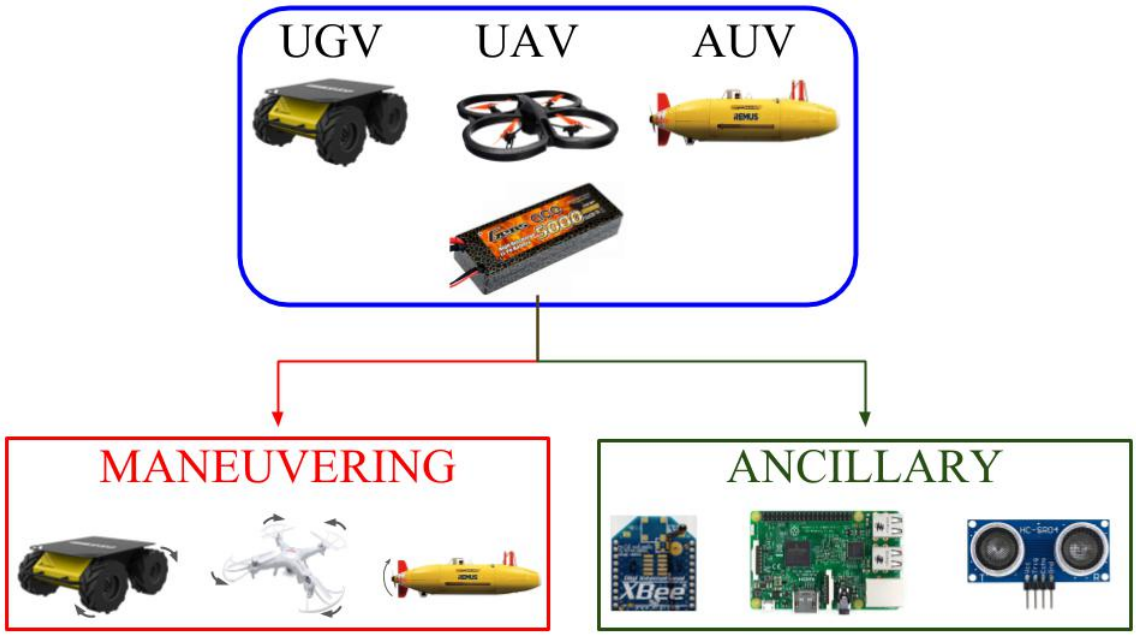


Figure 5.9: Energy distribution model for unification framework. Any type of robot, whether a micro UGV, quadrotor UAV, or AUV, uses portable battery packs which are utilized for essentially two functions: Firstly, maneuvering like propulsion, hovering, navigation etc. and secondly, ancillary functions like wireless communication, sensing, on-board processing etc.

maneuvering efficiency is given by ${}^{r}\Omega_{Man} \triangleq \Pi_{i=2}^{3}\neg\eta_i$ and the ancillary efficiency is given by ${}^{r}\Omega_{anc} \triangleq \neg\eta_4$.

As for the *traversal energy*, any robot (r), carrying out a mission (m) in an environment of choice experiences 4 kinds of forces:

1. Constant resistive force $F(r, m)$ as a function of robot (r) and the mission (m): *e.g.,* the force acting on a robot when it is traversing in a straight line under the influence of a constant magnetic field.

2. Environment dependent force $F(x, r, m)$ which is dependent on the current position x: *e.g.,* changing gravitational potential along with changing frictional force because of change in coefficient of friction.

3. Time dependent resistive force $F(t, r, m)$ which is a function of current time t: *e.g.,* unforeseeable disturbances (strong wind gusts *etc.*).

4. Instantaneous operational velocity dependent resistive force $F(v, r, m)$ which varies with instantaneous velocity v: *e.g.,* aerodynamics and gyro effect.

Thus, the net *traversal energy (TE)* is given in terms of *mechanical energy (ME)*

from longitudinal dynamics model and the net mechanical efficiency (${}^{r}\Omega_{Man}$) as:

$$\begin{aligned} TE &= \frac{ME}{{}^{r}\Omega_{Man}}, \\ &= \frac{{}^{Path}\int F_{net}dx}{{}^{r}\Omega_{Man}}, \\ &= \frac{{}^{Path}\int\{F(r,m)+F(x,r,m)+F(t,r,m)+F(v,r,m)\}dx}{{}^{r}\Omega_{Man}} \end{aligned} \tag{5.11}$$

Then, the instantaneous time (t) can be expressed as a function of position (x), velocity (v), mission (m) and duty cycle (D) as:

$$t = g(x, v, D, m) \tag{5.12}$$

During the mission (m), the robot traverses at an instantaneous velocity (v) and a fixed duty cycle (D). Thus,

$$\begin{aligned} TE &= \frac{{}^{Path}\int\{F(r,m)+F(x,r,m)+F(g(x,v,D,m),r,m)+F(v,r,m)\}dx}{{}^{r}\Omega_{Man}}, \\ &= \frac{\{F(r,m)+F(v,r,m)\}d}{{}^{r}\Omega_{Man}} + \frac{d_{Path}\int\{F(x,r,m)+F(x,v,D,r,m)\}dx}{d\ {}^{r}\Omega_{Man}} \end{aligned} \tag{5.13}$$

Moreover, the ancillary energy (AE) is given by:

$$AE = \frac{P_{anc}d}{v_{avg}D} \tag{5.14}$$

where,

$$\begin{aligned} P_{anc} &= \underbrace{\{s_0 + s_1 f_s\}}_{P_{sensing}} + \underbrace{\{P_{computation} + P_{communication}\}}_{P_C} \\ &= \{s_0 + s_1 f_s\} + \{P_{computation} + k \times data\ size \times f_{comm}\} \\ &= \{s_0 + s_1 f_s\} + \{P_{computation} + k \times data\ rate\} \end{aligned} \tag{5.15}$$

As an enhancement to the ancillary power consumption model over Eq. (5.6), here, the communication power ($P_{communication}$) is related with both the size of the data and the frequency (f_{comm}) at which the communication takes place. These two terms could be unified into data rate, *i.e.,* the amount of data sent in unit time. The communication power is then proportional to the data rate with a constant coefficient k (*c.f.* [14]) while the computation power ($P_{computation}$) is a function of the task allocated to the robot quantifying which is beyond the scope of this work. As for the power consumed by sensors given by $P_{sensing}$, it can be modeled as a function of the sampling frequency f_s. The scalars s_0, s_1 refer to the static power consumption and operational power consumption coefficient respectively.

So, the operational range for any robot can be generalized to:

$$d = \frac{\tilde{E}}{\frac{\{F(r,m)+F(v,r,m)\}}{{}^{r}\Omega_{Man}} + \frac{{}^{Path}\int\{F(x,r,m)+F(\frac{x}{vD},r,m)\}dx}{d\ {}^{r}\Omega_{Man}} + \frac{P_{anc}}{v_{avg}D}} \tag{5.16}$$

Here, ${}^{r}\Omega_{Man}$ is the net maneuvering efficiency of the robot, *i.e.,* the percentage of energy used to do actual mechanical work from the maneuvering branch. From Eq. (5.16), it is evident that in order to estimate the operational range, the term $\frac{{}^{Path}\int\{F(x,r,m)+F(\frac{x}{vD},r,m)\}dx}{d\ {}^{r}\Omega_{Man}}$ needs to be approximated and the operational range estimate would be as good as the approximation of this term.

5.5.4 Range estimation models for diverse robots

In order to approximate Eq. (5.16), 2 different approaches *viz., 1.) Offline* estimation which is a one-shot prediction model wherein range estimate is conjectured at the beginning of the mission itself and predictions are not corrected based on the new data acquired during mission, and *2.) Online* estimation whereby the estimation is recursively updated using all available operational data. In the conventional setting, *offline* estimates are generated once all the data is made available, while the *online* estimates are limited to the data currently available. As opposed to this setting, the *offline* model being referred to here relies on defining the required parameter values a priori and retaining the estimates. The *online* model on the other hand recursively updates the estimates as more data is made available. Furthermore, for each approach, as case studies, particular models for UGVs and multi-rotor UAVs are discussed.

5.5.4.1 Approach 1: *Offline* operational range estimation for diverse mobile robot platforms

Assuming some a priori known mission characteristics like driving profile, terrain attributes and system efficiency, *firstly* an Offline model is explained to estimate the operational range for diverse robots.

5.5.4.1.1 Case 1 : UGV operating in uneven terrain

In Section 5.4.2, the operational environment was assumed to be a smooth terrain with a constant elevation. However, in reality, this assumption is often rendered invalid as can be seen from Fig. 5.2. To suit such settings, the framework was further updated to envelope uneven terrains with variable elevation making it better suited to real world scenarios.

Any natural terrain can be modeled using three features: *1.) flats:* smooth surfaces with negligible gradient, *2.) slopes:* smooth surfaces with appreciable gradient and *3.) rubble:* uneven rough surfaces with no particular gradient characteristics. The operational terrain may have an average slope (γ) with respect to which the operational

range d should be calculated. In Fig. 5.10, the dashed line represents the actual terrain which must be traversed where QP represents the actual d. QR is the horizontal reference with respect to which the instantaneous road elevation is calculated.

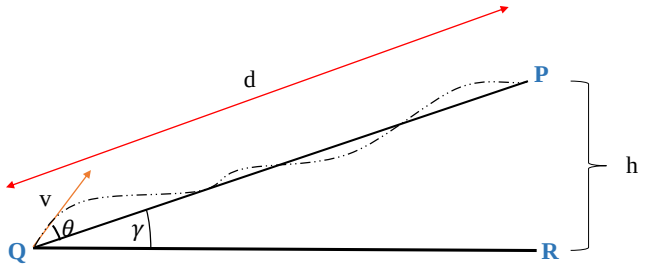


Figure 5.10: Schematic of actual terrain profile without rubble. v represents the instantaneous velocity, γ is the average terrain elevation and θ represents average road gradient with respect to γ. h represents the elevation gain and d represents the operational range.

1. Considering *flat* terrains exclusively, the only resistive force acting on the robot is the (rolling) friction between the wheels and the ground. This is defined as:

$$F_{Flats} = Normal\ Force \times C(x)_{rr} \tag{5.17}$$

   where, $C(x)_{rr}$ refers to the coefficient of rolling resistance.

2. Accounting for *slopes*, the net force acting will be friction along with the weight component along the motion of the robot. These forces are a function of the robot location x and the terrain elevation at x given by $\theta(x)$.

$$\begin{aligned} F_{slopes} &= F_{flats} + m_R g \sin(\gamma + \theta(x))\,, \\ &= C(x)_{rr} m_R g \cos(\gamma + \theta(x)) + m_R g \sin(\gamma + \theta(x)) \end{aligned} \tag{5.18}$$

3. Finally, considering *rubble*, in our force model, excess forces (F_{rubble}) acting due to presence of rubble need to be accounted for. Let $k(x)_{terr}$ be the terrain coefficient which depends on size, shape, density and resistance offered by the rubble. Then, the net forces (F_{net}) acting on the robot can be given by:

$$F_{net} = F_{slopes} + F_{rubble} \tag{5.19}$$

   Considering the limiting case, the following relationship holds:

$$\begin{aligned} F_{net} &= F_{slopes} + F_{rubble}\,, \\ &\triangleq k(x)_{terr}(F_{slopes}) \end{aligned} \tag{5.20}$$

Thus, the net maneuvering force (${}^{UGV}F(x)_{Man}$) for any UGV on an uneven natural terrain is given by:

$$ {}^{UGV}F(x)_{Man} = k(x)_{terr} m_R g[C(x)_{rr}\cos(\gamma+\theta(x)) + \sin(\gamma+\theta(x))] \tag{5.21} $$

In order to estimate achievable range d, first, the total energy model is defined in a real world setting as a sum of Ancillary Energy (AE) and Traversal Energy (TE):

$$
\begin{aligned}
\tilde{E} &= AE + TE\,, \\
&= \{Ancillary\ Power \times time\} + \frac{\int_{Path} {}^{UGV}F(x)_{Man}dx}{{}^{UGV}\Omega_{Man}}\,, \\
&= P_{anc} \times \frac{d}{v_{avg}\cos(\theta_{avg})D} \\
&+ \frac{\int_{Path} k(x)_{terr} m_R g[C(x)_{rr}\cos(\gamma+\theta(x)) + \sin(\gamma+\theta(x))]dx}{{}^{UGV}\Omega_{Man}}\,, \\
&= P_{anc} \times \frac{d}{v_{avg}\cos(\theta_{avg})D} \\
&+ \left\{\frac{m_R g}{{}^{UGV}\Omega_{Man}} \times \frac{\int_{Path} k(x)_{terr}[C(x)_{rr}\cos(\gamma+\theta(x)) + \sin(\gamma+\theta(x))]dx \times d}{d}\right\}\,, \\
&= d \times \left\{\frac{P_{anc}}{v_{avg}\cos(\theta_{avg})D} + \frac{m_R g}{{}^{UGV}\Omega_{Man}} \times \frac{\int_{Path} k(x)_{terr}[C(x)_{rr}\cos(\gamma+\theta(x)) + \sin(\gamma+\theta(x))]dx}{d}\right\}, \\
\Rightarrow d &= \frac{\tilde{E}}{\left\{\frac{P_{anc}}{v_{avg}\cos(\theta_{avg})D} + \frac{m_R g}{{}^{UGV}\Omega_{Man}} \times \frac{\int_{Path} k(x)_{terr}[C(x)_{rr}\cos(\gamma+\theta(x)) + \sin(\gamma+\theta(x))]dx}{d}\right\}}
\end{aligned}
\tag{5.22}
$$

The maximum attainable distance d_{max} is a function of the optimal velocity v_{opt}. Cruising at speeds higher/lower than v_{opt} would results in operational ranges lesser than d_{max}. So, Eq. (5.22) can now be written as:

$$ d_{max} = \frac{\tilde{E}}{\frac{P_{anc}}{v_{opt}\cos(\theta_{avg})D} + \left\{\frac{m_R g}{{}^{UGV}\Omega_{Man}} \times \frac{\int_{Path} k(x)_{terr}[C(x)_{rr}\cos(\gamma+\theta(x)) + \sin(\gamma+\theta(x))]dx}{d_{max}}\right\}}\,. \tag{5.23} $$

In Eq. (5.23), the factor $\left\{\frac{m_R g}{{}^{UGV}\Omega_{Man}} \times \frac{\int_{Path} k(x)_{terr}[C(x)_{rr}\cos(\gamma+\theta(x)) + \sin(\gamma+\theta(x))]dx}{d_{max}}\right\}$ denotes the average resistive force which acts on the robot on the path QP as shown in Fig. 5.10. Thus, replacing this factor by the expected average resistive force, the maximum traversal range (d_{max}) can be inferred and the estimation accuracy will be as good as the perfection in the estimation of the expected average resistive force. As the mechanical efficacy of the actuators ($\neg\eta_2$) varies with operational speed, v_{opt} is the velocity at which the net losses of ancillary and maneuvering branches are minimal. Also, as v_{opt} is a rather complex function of robot/actuators, exact trajectory traversed

or path taken and the mission characteristics, no further comments or profiling of v_{opt} is possible in the scope of current work. Thus, the target velocity set by the operator was considered as the v_{opt}.

In realistic scenarios, the *offline* model needs the values of $k(x)_{terr}, C(x)_{rr}$ to be defined for each x to find average expected resistive force. Or equivalently, the integral over the path can be eliminated by replacing it with average expected resistive force which can be done by replacing $k(x)_{terr}, C(x)_{rr}, \theta(x)$ by their averages $\bar{k}_{terr}, \bar{C}_{rr}, \bar{\theta}$, respectively. These values for the *offline* model can be estimated using any of the following methods: *1.)* Using the data and experience acquired over the previous missions; *2.)* carrying out a trial mission and then using the acquired information as prior knowledge for the actual mission; *3.)* using the expertise of the operators (system/environment experts) to provide realistic/good estimates. For this work, the approach 2.) mentioned above was considered.

5.5.4.1.2 Case 2 : Multi-rotor UAV operating in presence of external disturbances

Albeit the energy distribution for a UAV is quite similar to that of a UGV as mentioned previously in Section 5.5.3, but there are slight variations. The difference with respect to the latter being that, during the mission, a UGV may have phases of negligible maneuvering energy requirements whilst a UAV continuously needs to hover and maintain flight stability. As opposed to [15], the author not only considers the hovering and aerodynamic drag losses but also accounts for flight adjustments required due to unpredictable environmental factors (like strong wind gusts *etc.*).

Analogous to the UGVs, the energy for hovering, drag losses and flight adjustments in UAVs are comparable to energy requirements of motion over flats, slopes and rubble respectively. This is owing to the fact that, in case of hovering, the UAV experiences a constant environment dependent force required to stay aloft. Similarly, to maintain motion for a UGV, it must constantly overcome the resistive frictional forces. Identical analogues can also be drawn for the remaining cases.

1. For *hovering*, the energy consumption model is motivated by the works of [16]. For a UAV with N_R propellers each of radius r_p with a figure of merit Γ and rotor thrust T_{hover}, the P_{hover} can be defined as:

$$\begin{aligned} P_{hover} &= \frac{(T_{hover})^{\frac{3}{2}}}{\Gamma r_p \sqrt{2 N_R \rho \pi}}, \\ &= \frac{(m_R g)^{\frac{3}{2}}}{\Gamma r_p \sqrt{2 N_R \rho \pi}} \end{aligned} \tag{5.24}$$

2. Also accounting for *flight adjustments* [3], the instantaneous power $(P(t)_{fa})$ is given by:

$$P(t)_{fa} = \frac{[T(t)_{fa}]^{\frac{3}{2}}}{\Gamma r_p \sqrt{2 N_R \rho \pi}} \tag{5.25}$$

[3]The term *flight adjustments* takes into account all adjustments the UAV needs to make in order to maintain its course in presence of external disturbance or otherwise.

where the instantaneous thrust with flight adjustments ($T(t)_{fa}$) is defined as:

$$\begin{aligned} T(t)_{fa} &\triangleq T_{hover} + T(t)_{adjust}\,, \\ T(t)_{adjust} &= f(t)T_{controller} \end{aligned} \tag{5.26}$$

In Eq. (5.26), $T_{fa}(t)$ refers to the net thrust required for hovering with adjustments. This is defined in terms of hovering thrust (T_{hover}) and adjustment thrust ($T_{adjust}(t)$). The term $T_{controller}$ refers to the thrust required to follow the acceleration profile generated by the chosen flight controller (*e.g.,* PID controller or Neural Networks *etc.*) and $f(t)$ is a time dependent constant of proportionality. In order to expand the outreach of the model and remove the dependence on any particular flight controller, T_{adjust} was modeled as a time dependent function of T_{hover} as:

$$T(t)_{adjust} \triangleq k(t)_{env} T_{hover}\,. \tag{5.27}$$

Thus,

$$\begin{aligned} T(t)_{fa} &= T_{hover} + k(t)_{env} T_{hover}\,, \\ &= m_R g + k(t)_{env} m_R g\,. \end{aligned} \tag{5.28}$$

$$P(t)_{fa} = \frac{[m_R g + k(t)_{env} m_R g]^{\frac{3}{2}}}{\Gamma r_p \sqrt{2 N_R \rho \pi}} \tag{5.29}$$

Thus, the average power for *flight adjustments* over the entire time of flight (TOF) is given by:

$$P_{fa} \triangleq \frac{\int_{TOF} [\{m_R g + k(t)_{env} m_R g\} dt]^{\frac{3}{2}}}{TOF\ \Gamma r_p \sqrt{2 N_R \rho \pi}} \tag{5.30}$$

3. Finally, *drag losses* acting on rotor blades need to be incorporated. The drag force on N_R propellers is estimated from fluid mechanics as:

$$\begin{aligned} F_D &= \frac{N_R \rho C_D A v^2}{2}\,, \\ &= \frac{N_R \rho C_D A (r_p \omega)^2}{2}\,. \end{aligned} \tag{5.31}$$

The drag torque (τ_D) is given by:

$$\begin{aligned} \tau_D &= \int_0^{r_p} F_D dr\,, \\ &= \frac{N_R \rho C_D A r_p^3 \omega^2}{6} \end{aligned} \tag{5.32}$$

Since the power required to overcome the drag losses is given by $\tau_D\omega$ and the drag thrust $T(t)_{fa}(= N_R k_r \omega^2)$ for propeller constant k_r, the instantaneous power for drag losses ($P_D(t)$) is computed as:

$$P_D(t) = \frac{\rho C_D A r_p^3 [T(t)_{fa}]^{\frac{3}{2}}}{6k_r\sqrt{N_R}} \tag{5.33}$$

Substituting Eq. 5.28 into Eq. 5.33 to get the average power for drag losses as:

$$P_D = \frac{\rho C_D A r_p^3 [\int_{TOF}\{m_R g + k(t)_{env} m_R g\}dt]^{\frac{3}{2}}}{6k_r\sqrt{N_R}\ TOF} \tag{5.34}$$

The net energy required for navigation of a UAV is now given based on Eq. (5.10) as:

$$\begin{aligned}
\tilde{E} &= AE + TE\,, \\
&= Ancillary\ Power \times TOF + \frac{[P_D + P_{fa}]TOF}{{}^{UAV}\Omega_{Man}}\,, \\
&= P_{anc} \times TOF + \frac{[\frac{\rho C_D A r_p^3 [\int_{TOF}\{m_R g + k(t)_{env} m_R g\}dt]^{\frac{3}{2}}}{6k_r\ TOF\sqrt{N_R}} + \frac{\int_{TOF}[\{m_R g + k(t)_{env} m_R g\}dt]^{\frac{3}{2}}}{TOF\ \Gamma r_p\sqrt{2N_R\rho\pi}}]}{{}^{UAV}\Omega_{Man}}\,, \\
&= P_{anc} \times TOF + \left\{\frac{\left[\frac{\rho C_D A r_p^3}{6k_r\sqrt{N_R}}\right] + \left[\frac{1}{\Gamma r_p\sqrt{2N_R\rho\pi}}\right]}{{}^{UAV}\Omega_{Man}}\right\}\left\{\frac{\int_{TOF}[\{m_R g + k(t)_{env} m_R g\}dt]^{\frac{3}{2}}}{TOF}\right\}TOF
\end{aligned} \tag{5.35}$$

$$\Rightarrow TOF = \frac{\tilde{E}}{P_{anc} + \left\{\frac{\left[\frac{\rho C_D A r_p^3}{6k_r\sqrt{N_R}}\right] + \left[\frac{1}{\Gamma r_p\sqrt{2N_R\rho\pi}}\right]}{{}^{UAV}\Omega_{Man}}\right\}\left\{\frac{\int_{TOF}[\{m_R g + k(t)_{env} m_R g\}dt]^{\frac{3}{2}}}{TOF}\right\}} \tag{5.36}$$

Replacing $TOF = \dfrac{d}{v}$ in Eq. (5.36), the revised equation becomes:

$$\begin{aligned}
\frac{d}{v} &= \frac{\tilde{E}}{P_{anc} + \left\{\frac{\left[\frac{\rho C_D A r_p^3}{6k_r\sqrt{N_R}}\right] + \left[\frac{1}{\Gamma r_p\sqrt{2N_R\rho\pi}}\right]}{{}^{UAV}\Omega_{Man}}\right\}\left\{\frac{\int_{TOF}[\{m_R g + k(t)_{env} m_R g\}dt]^{\frac{3}{2}}}{TOF}\right\}}\,, \\
\Rightarrow d &= \frac{\tilde{E}}{\frac{P_{anc}}{vD} + \left\{\frac{\left[\frac{\rho C_D A r_p^3}{6k_r\sqrt{N_R}}\right] + \left[\frac{1}{r_p\sqrt{2N_R\rho\pi}}\right]}{{}^{UAV}\Omega_{Man}v}\right\}\left\{\frac{\int_{TOF}[\{m_R g + k(t)_{env} m_R g\}dt]^{\frac{3}{2}}}{TOF}\right\}}
\end{aligned} \tag{5.37}$$

The theoretical maximal operational is only attainable when the UAV operates at v_{opt} such that minimal losses are accrued. Thus,

$$d_{max} = \frac{\tilde{E}}{\frac{P_{anc}}{v_{opt}D} + \left\{ \frac{\left[\frac{\rho C_D A r_p^3}{6k_r\sqrt{N_R}}\right] + \left[\frac{1}{r_p\sqrt{2N_R\rho\pi}}\right]}{^{UAV}\Omega_{Man} v_{opt}} \right\} \left\{ \frac{\int_{TOF}[\{m_R g + k(t)_{env} m_R g\}dt]^{\frac{3}{2}}}{TOF} \right\}},$$

$$\because \text{D=100\% for UAV}, \tag{5.38}$$

$$d_{max} = \frac{\tilde{E}}{\frac{P_{anc}}{v_{opt}} + \left\{ \frac{\left[\frac{\rho C_D A r_p^3}{6k_r\sqrt{N_R}}\right] + \left[\frac{1}{r_p\sqrt{2N_R\rho\pi}}\right]}{^{UAV}\Omega_{Man}} \right\} \left\{ \frac{\int_{TOF}[\{m_R g + k(t)_{env} m_R g\}dt]^{\frac{3}{2}}}{TOF\ v_{opt}} \right\}},$$

In Eq. (5.38), the factor,

$$\left\{ \frac{\left[\frac{\rho C_D A r_p^3}{6k_r\sqrt{N_R}}\right] + \left[\frac{1}{\Gamma r_p\sqrt{2N_R\rho\pi}}\right]}{^{UAV}\Omega_{Man}} \right\} \left\{ \frac{\int_{TOF}[\{m_R g + k(t)_{env} m_R g\}dt]^{\frac{3}{2}}}{TOF\ v_{opt}} \right\},$$

represents the average resistive force experienced by the UAV over the entire time of flight. This is akin to Eq. (5.23) which serves to satiate the requirement for developing a *Unified framework*. Apt replacement of this parameter by using the expected average resistive force can help to estimate the maximum operational range for the UAV. The error in estimation of this factor directly translates to the error in expected operational range.

5.5.4.2 Approach 2: *Online* operational range estimation for diverse mobile robot platforms

In Section 5.5.4.1, the author presented an *offline* range estimation model whereby, based on a priori known mission characteristics, the maximum attainable range for mobile robots was estimated. However, in reality, it might be rather challenging to strictly follow the mission characteristics or to even obtain a priori mission information. Furthermore, the approximation of factors in Eqs. (5.23) and (5.38) are largely dependent on the expertise of the human operator supervising the mission. In order to adapt to unforeseen and unavoidable variations in the mission profile, the author now proposes an *online* variant of the operational range estimation framework. In this method, based on all available real-time data (historic and current), the operational range is recursively updated. As the mission progresses and more data is acquired about the mission characteristics, the range estimation model updates the estimate of the maximum operational range in real-time. This is crucial for real-world missions as it allows for flexibility in missions themselves and can be coupled with energy efficiency

path planners [17] to dynamic adapt to situations as they present themselves. Similar to Section 5.5.4.1, UGV and multi-rotor case studies are re-introduced in an *online* data acquisition setting.

5.5.4.2.1 Case 1: UGV operating in uneven terrain

In Eq. (5.23), the terms $k(x)_{terr}$ and $\theta(x)$ can either be set by a human operator (offline model) or can be deduced from prior missions carried out in that terrain. However, the former introduces human error and the latter is usually not available. Thus, as an alternative, $k(x)_{terr}, \theta(x)$ can instead be replaced by their respective estimates, $\hat{k}(x)_{terr}, \hat{\theta}(x)$. Additionally, the *offline* model used instantaneous rolling resistance $C_{rr}(x)$ while here it is being approximated by a constant C_{rr}. So, the estimated maximum range for the remaining mission is now given by:

$$\hat{d}_{Max}^{[t:end]} \triangleq \frac{\tilde{E}_{rem}}{\frac{P_{anc}}{v_{opt}D} + \frac{\hat{k}(x)_{terr}[C_{rr}\cos\hat{\theta}(x) + \sin\hat{\theta}(x)]m_R g}{^{UGV}\Omega_{Man}}}, \tag{5.39}$$

where

$$\tilde{E}_{rem} = \tilde{E}^{[0:end]} - \tilde{E}^{[0:t]} \tag{5.40}$$

Here, $\tilde{E}_{rem}$ is the useful energy remaining in the battery and $\tilde{E}^{[0:end]}$ is the usable energy present in the battery at the start of the mission,i.e, at $t = 0$. Similarly, $\tilde{E}^{[0:t]}$ is the energy spent from $t = 0$ to time instance t. Now, the total estimated maximum operational range over the entire mission is given by:

$$\hat{d}_{max}^{[0:end]} = d^{[0:t]} + \hat{d}_{max}^{[t:end]} \tag{5.41}$$

In Eq. (5.41), to estimate the net operational range for the entire mission ($\hat{d}_{max}^{[0:end}]$), the distance that has already been covered ($d^{[0:t]}$) is utilized to estimate the maximum distance that may be covered ($\hat{d}_{max}^{[t:end]}$) based on available residual energy. The value of $\hat{k}(x)_{Terr}$ that is required for estimating $\hat{d}_{max}^{[t:end]}$ can be estimated using Eq. (5.39). During the mission, after every time-step (t), the robot will have the knowledge of the distance that it has covered in that time-step, energy it has spent to cover that distance and terrain elevation θ for that time step. Substituting the value of $d^{[t-1:t]}$ for d_{max} and $E^{[t-1:t]}$ for $\tilde{E}_{rem}$ in Eq. (5.39), the value of $k(x)_{terr}^{[t]}$ for the given time-step t can be calculated. Now, these set of values of $k(x)_{terr}^{[t]}$ can be used to estimate $\hat{k}(x)_{terr}$ which in turn can be used to make predictions about the distance the robot can still cover using the remaining energy. Since the estimate for the remaining distance depends upon the value on the estimation of $\hat{k}(x)_{terr}$ and $\hat{\theta}(x)$, which needs to be recursively updated as new data is being collected, we utilize a recursive average filter. For ease of notation, let $\hat{\mathbf{X}}^{[t]} = [\hat{k}(x)_{Terr}^{[t]}, \hat{\theta}(x)^{[t]}]$ and $\mathbf{Z}^{[t-1]} = [k(x)_{terr}^{[t-1]}, \theta(x)^{[t-1]}]$ which represents the set of actual (noisy) measurements up till the last time step $(t-1)$.

Then, given a noisy set of measurements, $\mathbf{Z}^{[0:t-1]}$, and no additional information about the impact of environmental factors on the system dynamics, a reasonable estimate for the system state at the current time-step, t can be obtained as:

$$\hat{\mathbf{X}}^{[t]} = P^{[t-1]} \sum_{i=0}^{t-1} \mathbf{Z}^{[i]} \tag{5.42}$$

where $P^{[t-1]} = \dfrac{1}{t-1}$ represents the responsiveness of the filter, *i.e.,* the filter is very responsive (making a lot of corrections) in the beginning since limited data is available. As time passes and more data becomes available, the filter becomes more certain about its estimates, and thus, reduces the relative importance of the measurements. However, being a fixed response model (true values of k_{terr} and θ_{terr} are fixed) with response rate decreasing with time, it cannot always adapt to sudden changes in the values of $k(x)_{terr}$ and $\theta(x)_{terr}$ as $\frac{1}{t-1}$ can be very small. These sudden changes can occur when there is a change in terrain type or weather conditions but their impact will diminish with the passage of time. Manipulating Eq. (5.42) to obtain the recursive update rule as follows:

$$\begin{aligned}
\hat{\mathbf{X}}^{[t]} &= P^{[t-1]} \sum_{i=0}^{t-1} \mathbf{Z}^{[i]} \\
&= P^{[t-1]} \sum_{i=0}^{t-2} \mathbf{Z}^{[i]} + P^{[t-1]} \mathbf{Z}^{[t-1]} \\
&= \frac{t-2}{t-1} \times \underbrace{\frac{1}{t-2} \sum_{i=0}^{t-2} \mathbf{Z}^{[i]}}_{\hat{\mathbf{X}}^{[t-1]}} + P^{[t-1]} \mathbf{Z}^{[t-1]} \\
&= \hat{\mathbf{X}}^{[t-1]} + P^{[t-1]} \left(\mathbf{Z}^{[t-1]} - \hat{\mathbf{X}}^{[t-1]} \right)
\end{aligned} \tag{5.43}$$

Eq. (5.43) represents the recursive state update rule wherein the term $P^{[t-1]} \left(\mathbf{Z}^{[t-1]} - \hat{\mathbf{X}}^{[t-1]} \right)$ represents the *measurement innovation i.e.,* the new information acquired via the new observation. Similarly, the recursive update rule for the filter response can also be derived as:

$$P^{[t]} = P^{[t-1]} - P^{[t-1]} (P^{[t-1]} + 1)^{-1} P^{[t-1]} \tag{5.44}$$

From Eqs. (5.43)-(5.44), it is clear that the filter is a modified moving average filter [18] with increasing window size, that accommodates all the data available. The predictions begin at $t = 2$, and $\hat{\mathbf{X}}^{[1]} = \mathbf{Z}^{[1]}$. Here, we can see that $\mathbf{X}^{[t]}$ is a function of $\mathbf{Z}^{[0:t-1]}$ which is a series of points indexed in time order *i.e.,* a time-series. So we can use common time-series forecasting method to estimate the value of $\mathbf{X}^{[t]}$ such as various variants of Autoregressive moving average (ARMA) model [19]. In our case,

we have used a modified moving average model (ARMA(0,0,1)), that computes the average of all the data points available to estimate the value of $\mathbf{X}^{[t]}$.

5.5.4.2.2 Case 2: Multi-rotor UAV operating in presence of external disturbances

Similar to the case of UGVs, the term $k(t)_{env}$ in Eq. (5.38) is now replaced by its estimated value $\hat{k}(t)_{env}$ based on autoregressive model presented in Eq. (5.43). Thus, the estimated maximum range for UAV for the remainder of the mission is now given by:

$$\hat{d}_{max}^{[t:]} \triangleq \frac{\tilde{E}_{rem}}{\frac{P_{anc}}{v_{opt}} + \left\{ \frac{\left[\frac{^{UAV}\Omega_{Man}\rho C_D A r_p^3}{6k_r\sqrt{N_R}}\right] + \left[\frac{1}{r_p\sqrt{2N_R\rho\pi}}\right]}{^{UAV}\Omega_{Man}v_{opt}} \right\} [m_R g + \hat{k}(t)_{env} m_R g]^{\frac{3}{2}}} \tag{5.45}$$

Now, the maximum operational range estimation can be done similar to Eq. (5.41) and Eq. (5.43). However, in case of UAV, let $\hat{X}^{[t]} = \hat{k}(t)_{env}$ and $Z^{[t-1]} = k(t-1)_{env}$ define the estimated and observed values of the environmental variable which are required for operational range estimation.

5.6 Experiments

In Sections 5.5.4.1 and 5.5.4.2, the author presented *offline* and *online* range estimation models for the unified framework. A precursor to the *offline* range estimation model was also discussed in Section 5.4.2. All 3 range estimation models were empirically evaluated in the following settings:

- **Simplified Range Estimation Model:** was tested in indoor setting which usually present smooth surfaces $\theta \approx 0$ with constant average gradient of slope $\gamma \approx$ constant. The test-bed used for indoor experiments was *Rusti V1.0* as shown in Fig. 5.11.
- **Unified Framework:** was designed to account for outdoor environment conditions along with unforeseen and unavoidable variations to missions that may have to be tackled in real missions. Thus, this framework was tested in natural outdoor settings. The test-bed for outdoor experiments were *Rusti V2.0* shown in Fig. 5.12 and *ArDrone 2.0* shown in Fig. 5.13.
  - *Offline Range Estimation Model:* The parameters in Eqs. (5.23) and (5.23) were set based on few prior experiments that had to be carried out independent of the actual field trials.

– *Online Range Estimation Model:* The parameters in Eqs. (5.23) and (5.23) were updated in real-time using the autoregressive moving average filter.

In what follows, firstly, the indoor experiments are presented for the simplified model followed by outdoor experiments for the unified framework.

Figure 5.11: Rusti V1.0 with omnidirection wheels, ultrasonic sensor and short range wireless communication module (XBee) for indoor navigation.

Figure 5.12: Rusti V2.0 with all-terrain wheels and external 3-axis accelerometer sensor for outdoor field experiments carried out on asphalt, grass and tiles.

5.6.1 System Identification

Before presenting the empirical analysis for indoor and outdoor field trials, a summary of system identification parameters is given in the table below:

Table 5.1: System Efficiency Calibration for Rusti V1.0, Rusti V2.0 and ArDrone 2.0

Robot	$\neg\boldsymbol{\eta}_1$	$\neg\boldsymbol{\eta}_2$	$\neg\boldsymbol{\eta}_3$	$\neg\boldsymbol{\eta}_4$	$\Omega = \Pi_{i=1}^{4}\eta_i$
Rusti V1.0	99.5%	94.2%	9.2%	99.9%	8.615%
Rusti V2.0	99.5%	**	**	99.9%	**
ArDrone V2.0	99.5%	27%		99.9%	26.84%

In Table 5.1, the items marked with ** were not directly observable/measurable. Such terms were instead accounted for by clustering of variables in the equations and

Figure 5.13: ArDrone 2.0 with GPS used for outdoor experiments in parking lot and public park.

considering multiple terms in unison.Additionally, for the drone, average propulsion efficency was jointly obtained from a closely related work [6]. Further details can be found in [13].

5.6.2 Indoor experiments

The indoor experiments were performed using Rusti V1.0 equipped with $HC-SR04$ Ultrasonic ranging module. Box shaped trajectories were executed for planar surfaces and oscillating trajectories were executed for inclined surfaces as illustrated via Fig. 5.14. In order to validate the ancillary power consumption model given by P_{anc} in Eq. (5.6), the ultrasonic sensor was operated at various operational frequencies and the power consumed was plotted as shown in Fig. 5.15. The model for ancillary power consumption was verified empirically. The power consumption model obtained is:

$$P_{sensing} = 5.7318e^{-5}\ f_s + 0.0293 \tag{5.46}$$

From Eq. (5.46), it becomes clear that the power consumed by the sensor array in idling state is $0.0293W$. The worst case power consumption at a sampling rate of $100Hz$ was found to be $0.0348W$. The power consumption of the micro-controller unit (MCU) which controls the ultrasonic ranging sensor and the wireless communication (XBee) was found to be quite stable at $0.3928W$ with XBee consuming $0.166W$ irrespective of the size of data being transmitted. Thus, the overall ancillary power consumption model now becomes:

$$P_{anc} = \underbrace{\{5.7318e^{-5}\ f_s + 0.0293\}}_{P_{Sensing}} + \underbrace{\{\underbrace{0.166}_{P_{XBee}} + \underbrace{0.3928}_{P_{MCU}}\}}_{P_C} \tag{5.47}$$

In Fig. 5.16, the energy utilized as the robot covers more ground is showcased. Localization was disregarded for these experiments so that the computational energy

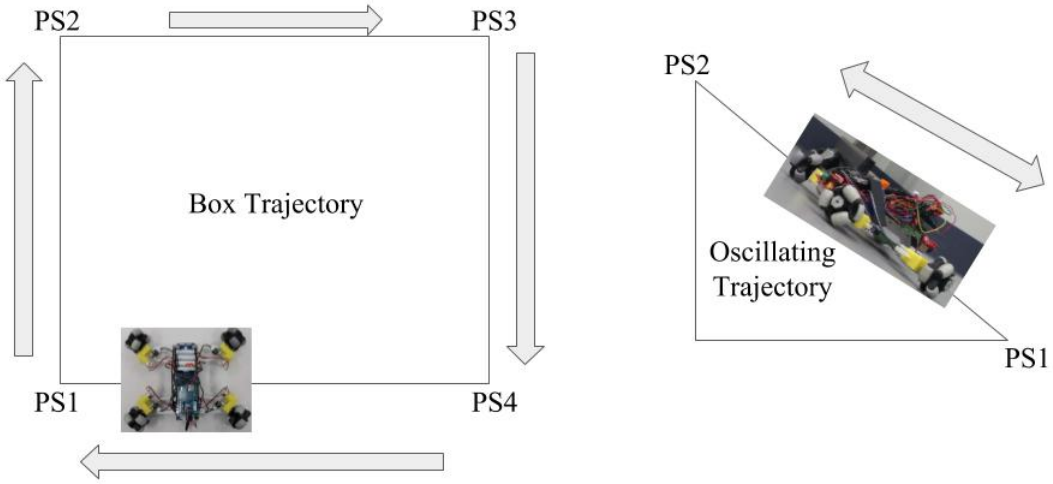


Figure 5.14: Indoor trajectories. Box type trajectory on planar grounds and oscillating trajectory on inclined plane. PS_i represents pit stops where the robot was made to stop for pre-determined time period. This was done to emulate scenarios where a robot may need to stop and process data during a real mission.

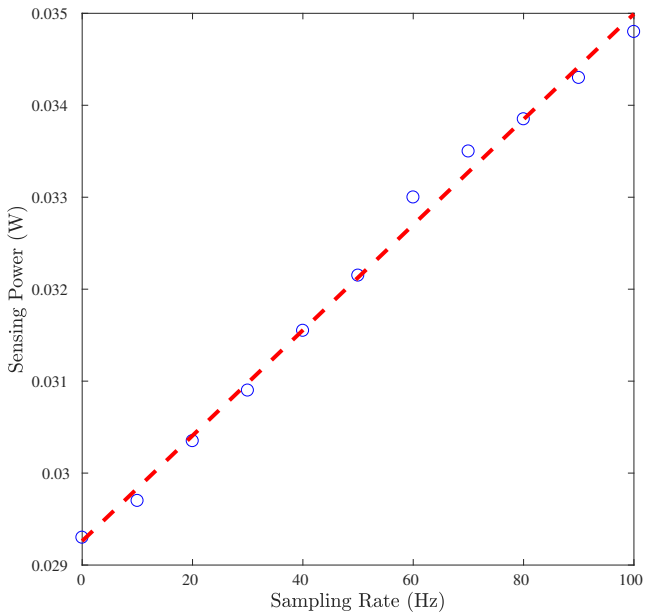


Figure 5.15: Energy consumed by sonar. Illustrating modulating in ancillary power consumed by $HC-SR04$ Ultrasonic ranging module when operating at various frequencies f_s.

could be quantified. Quantifying the energy consumed by the robot for performing SLAM, Localization, *etc.,* is beyond the scope of this work as they require low level cache management for the on-board processors and also depend on the nature of the source code used. Besides, no matter how accurately these factors can be accounted for, the energy that goes into these components does not contribute to the maneuvering energy and hence, the author chose to ignore them. An interesting fact to note in this figure is that, owing to lack of localization, the robot drifted from its assigned path especially at high duty cycles which was even more pronounced on the inclined plane. If the robot were to precisely follow the box trajectories and the oscillating trajectories, then the trends would have been quite linear. Nonetheless, this was not the objective of the experiments. The trajectories were pre-coded since the robot was not meant to be fully autonomous but in real world scenarios a robot usually does not follow repetitive trajectories. Also, the distance covered is maximum when the duty cycle is 100% whilst it decreases as the robot spends more time stopping for gathering and processing information. This is in accord with the definition of the duty cycle.

Fig. 5.17 shows the estimation error for the simplified range estimation model for both planar and incline environments. In this figure, negative values indicate underestimation *i.e.,* achieved range was larger than the estimated values whilst the positive error represent vice versa. The accuracy of this model was evaluated to be $66\% \sim 91\%$. However, as the duty cycle is reduced further, the model sometimes has trouble to precisely estimate the power consumption for ancillary functions which leads to erroneous estimates for the achievable range. The reason for this could be attributed to the fact that at lower duty cycles, the robot transmits larger amounts of data thus, the ancillary power must also account for the data transmission rate which is beyond the scope of current model but was accounted for in the unified framework. Additionally, on-board vibrations introduce noise in sensor data being acquired which also affects the model performance. Thus, the performance was logged for very noisy data.

The component wise power consumption was also evaluated for which a detailed breakdown is shown in Table 5.2. The maneuvering power was estimated for fixed average speed of $\approx 1m/s$ and the sensing power was estimated for sampling rates ranging from $0Hz \sim 100Hz$. For the case of $0Hz$, we implemented a low power sleep model for our sensor to significantly reduce the power consumption. This could come in very handy later on, when designing a controller/ real time scheduler, that can divert the power from ancillary branch to maneuvering branch to further enhance the achievable range by cutting down unnecessary power consumption.

The minimum and maximum percentage of a component's power to the total system power are shown in Table 5.2. To find the minimum, the component is assumed to be running at its minimum power whilst the rest of the system consumes the maximum power and vice versa for the maximum [20]. From this table, it can be show that when on-board computation is not required, 90% of the total power is consumed by the motors for maneuvering. However, when adding an embedded computer to the ancillary branch (which consumes $8W \sim 15W$ power [20]), this composition will go down significantly (30.5%∼36.5%). This composition will be further affected by varying velocities for traversal and type of sensors. Thus, accurately estimating the

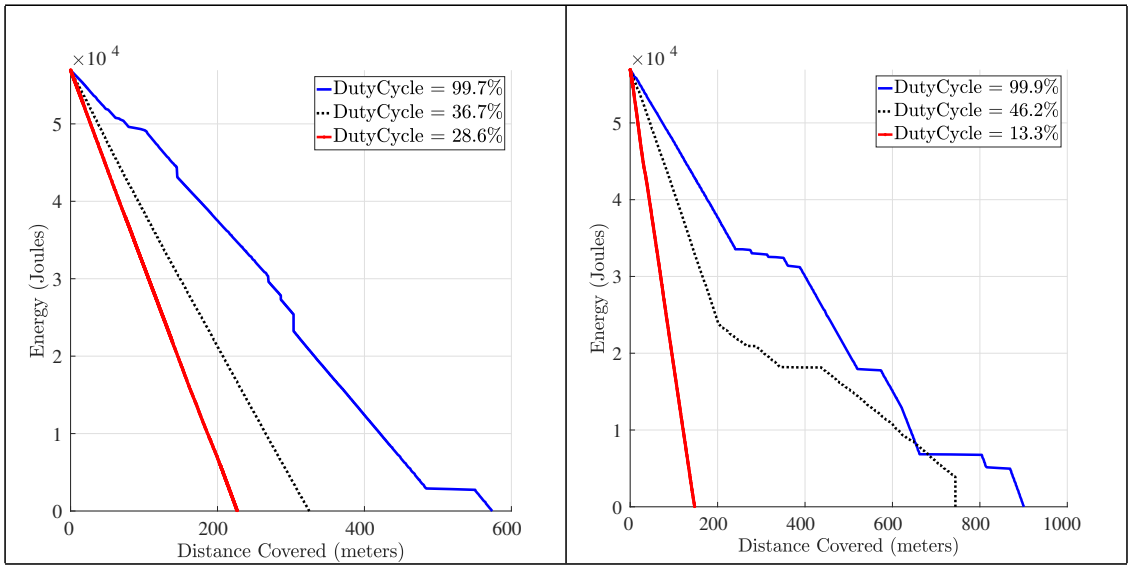


Figure 5.16: Energy utilization for flat plane and incline slope experiments.

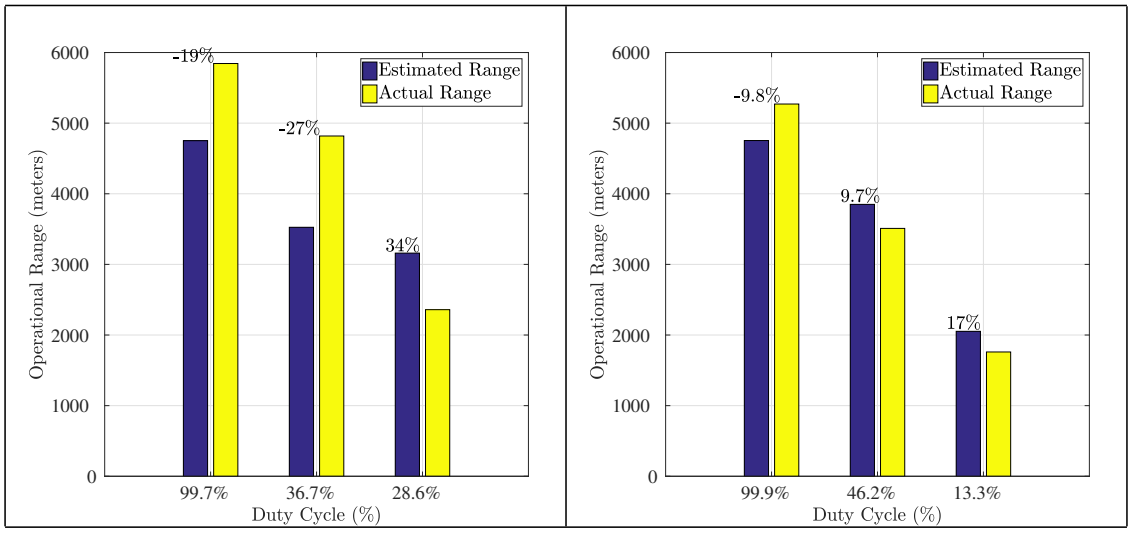


Figure 5.17: Operational range estimation for flat plane and incline slope experiments.

range is all the more critical in such cases.

Table 5.2: Power Consumption Breakdown

Component	Power (W)	Composition(%)
Maneuvering	$4.8158W \sim 6.8456W$	$89.13\% \sim 92.02\%$
Sensing (Ultrasonic Senor)	$0.0293W \sim 0.0348W$	$0.46\% \sim 0.54\%$
Wireless Communication	$0.165W \sim 0.166W$	$2.23\% \sim 3.05\%$
Micro-controller unit (MCU)	$0.3928W$	$5.29\% \sim 7.26\%$

5.6.3 Outdoor experiments

In this section, the author explains the outdoor experimental conditions in which the unified framework from Section 5.5.4 was evaluated. For the UGV, 36 experiments were carried out on various terrains comprising of either grass, tiles, or asphalt with varying elevations and wheel rpm of 80 and 140[4]. The reason for considering these terrain types individually was the lack of capable hardware to determine change in terrain types on-the-fly and accordingly adjust the coefficient of rolling friction for making online prediction. This could be achieved if a camera and ladar were to be used to subtract the background information and match the features of the foreground with pre-selected images of the terrains that can be seen in real field trials. Such ideas have been explored in works like [21] that use ladar and camera to complement each other to detect obstacles and classify the terrain. However, this procedure fails in lack of proper lighting conditions and additionally, estimating the rolling friction coefficient was beyond the scope of this work.

For system efficiency calibrations, 6 *minimal load* tests at 100 and 200 rpm were performed. The average rolling coefficients for terrain resistance offered by grass, tiles, and asphalt were set as $0.099, 0.066$, and 0.062, respectively and the prior information of γ to be used in the *offline* estimation was set based on Table 1 of [7]. In order to deduce the average values of the parameters, we used $\overline{k}_{terr}^{[0:t]} \leftarrow \frac{\tilde{E}^{[0:t]} - AE^{[0:t]}}{F_{slopes}}$. Similarly, the equations for $\overline{\gamma}^{[0:t]}, \overline{\theta}^{[0:t]}$ can be deduced.

As for the UAV, 30 field trials were executed which were split into two different sets, *viz., hovering* and *motion*. For *hovering*, only the altitude of the UAV was varied and the human operator occasionally had to send control commands to maintain the position of the drone within a set perimeter. As opposed to this, in *motion* case, the human operator constantly fed linear motion commands to the drone whilst occasionally commanding the drone to hover (in cases when wind gusts lead to dangerously high velocity gains). This not only helped ensure the safety of the drone

[4]Given the wheel radius of $65mm$, these translate to $v = 0.544, 0.952$ m/s respectively. The velocities were pre-set at the beginning of the field trial and were not monitored during the field trial. For the *offline* and *online* models, the heading velocity remained constant and for turning, while one side of motors were slowed by δ, the other side was sped up by the same factor. This ensured that the average velocity of the center of mass of the robot remained constant.

and its operator but also helped emulate the real life scenarios in which the drone may loose connection to the base station (Fig. 5.7b) or corruption of mission critical information (Fig. 5.7a). The control commands were sent at operator-defined data transmission rate such that the $P_{communication}$ in Eq. (5.15) remained constant. 10 experiments for *hovering* at different altitudes varying between $[1, 10]$ meters were carried out. Furthermore, 20 tests at 5 different operational velocities[5] for *motion* to account for a mix of wind gusts, altitude adjustments, variable mission speeds, and trajectories were also considered. Experiments were performed at intervals of 2 hrs so as to account for changing environmental factors like wind and weather conditions.

In case of UAV, the wind compensation angle of the UAV was constantly monitored to evaluate the adjustments in the thrust that the robot needs to make to maintain flight stability. Through simple geometry, this was then used to calculated real-time values of $k(t)_{env}$ as explained in Fig. 5.18. From this figure, it can be seen that the net stabilization required on the part of the rotorcraft will be $CB = OB(1-\cos(\theta))$. When the rotorcraft was maintaining a constant altitude, the value of $OB = m_R g$. So here CB represents $k(t)_{env} m_R g$ as explained in Eq. (5.27). Therefore, $k(t)_{env} = 1 - \cos(\theta)$.

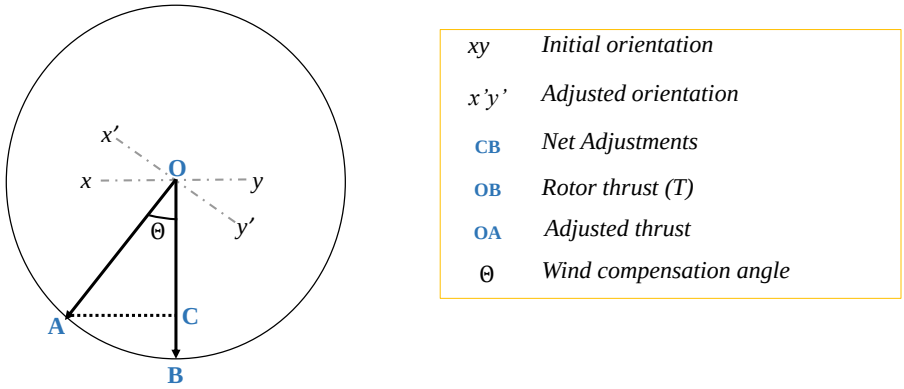


Figure 5.18: Geometric analysis of wind compensation angle to deduce the value of parameter $k(t)_{env}$. Suppose the UAV is stable, then the orientation is represented by xy and OB represents the thrust (T) exerted by the UAV for maintaining flight. Now, assume that because of sudden wind action, the rotorcraft is displaced by an angle θ (which can also be interpreted as wind compensation angle) and the new orientation is $x'y'$. So, OA will represent the same thrust under the sudden influence of the wind at an angle θ to the previous direction. Thus, the net altitude destabilization effect of the wind is given by BC.

For Eq. (5.23) and Eq. (5.38), as was discussed earlier on, to make predictions of d_{max}, the average resistive forces need to be approximated, which in turn are factors of $k(x)_{terr}$ and $k(t)_{env}$. So, to estimate these parmeters in real-time, the ARMA filter is fed with real-time mission data.

[5]The translation velocities were chosen from $[0.1, 0.2, 0.4, 0.6, 0.8]$ m/s which are subject to brief change upon change in heading direction. *E.g.,* consider operational velocity of 0.1 m/s along $+X$ direction. Upon request to change direction to $-X$, the velocity during this brief transition period will vary from 0.1 m/s in $+X$ to 0 m/s in $+X$ followed by 0.1 m/s in $-X$ direction. This velocity profile cannot be feasibly estimated for *offline* model, so we directly used the operational velocity (0.1 m/s for this *e.g.*), while for the *online* model, velocity was continuously observed, so the average velocity till current time step was used.

5.6.4 Batteries used for field experiments

Since the *ArDrone* comes factory fitted with a mini-tamiya connector, the stock battery *i.e.,* $11.1V@1500mAh$ high density LiPo battery was used for it. However, having custom built the *Rusti V2.0*, the following 2 LiPo batteries were considered for field trials:

- $11.1V@2200mAh$
- $11.1V@1500mAh$ (also used for *ArDrone*)

Subsequent sections discuss the results obtained for the UGV followed by UAV during the outdoor field trials.

5.6.5 Case 1 : UGV

In Figs. 5.19 - 5.21, both the *offline* and *online* models are pegged against the true achieved range during real field trials on grass, asphalt, and tiles, respectively. For *offline* estimation,based on Eq. (5.23), it was previously explained that for estimating the operational range, the model needs prior information about θ, γ, and k_{terr}. Also, the mechanical efficiency[6] is unascertained. The value for γ was obtained based on Table 1 of [7], and that of θ_{avg} was empirically set to 5°. Estimating the values of k_{terr} and ${}^{Rusti}\Omega_{Man} = \Pi_{i=2}^{3} \neg\eta_i$ are rather challenging and require some prior field experience. Owing to lack of any such experience, the author rather chose to modify Eq. (5.23) such that the maneuvering efficiency term *i.e.,* ${}^{Rusti}\Omega_{Man}$ is now considered within the integral and the new term $\frac{k_{terr}}{{}^{Rusti}\Omega_{Man}}$ was treated a single, terrain dependent variable. The average value of this terrain dependent factor was then determined through a series of field trials as $\frac{k_{terr}}{{}^{Rusti}\Omega_{Man}} = [3.09, 2.81, 2.69]$ for grass, asphalt, and tiles, respectively. For *online* estimation, the belief of the model over the net operational range achievable is updated in real-time based on cumulative performance characteristics. Effectively, on an average, the net true distance covered by the robot at $0.544m/s$ and $0.952m/s$ are almost the same on all types of terrain. This can be attributed to the fact that because of the use of high torque DC motors in *Rusti V2.0*, the net ancillary energy requirements are negligible compared to maneuvering energy (which is independent of operational speed). From Eq. (5.23) we see that this in fact will be the case if $P_{anc} \ll P_{man}$. Also, the true distance covered using the $2200mAh$ battery is greater than that covered using the $1500mAh$ battery but they are not in proportion of the battery capacities *i.e.,* the ratio of battery capacities is $\frac{22}{15} = 1.47$ but the ratio of achieved true range is $\frac{6.67}{5.17} = 1.29$. This difference can be attributed to the fact that the mass of the robot is slightly higher when using the $2200mAh$ battery which dilutes the effect of extra charge capacity. However, these proportions will be drastically affected so much so that the maneuvering energy consumption could account for less than 50% of the net energy supplied by the battery [20]. Having said this, it must also be pointed out here that this proportion depends on the mission profile and the

[6] ${}^{Rusti}\Omega_{Man}$ is a factor of only motors and will account for both frictional losses as well as heat losses in motor

load borne by the sensor array being using. These are beyond the scope of this work as the robot with minimal sensor array was used for analysis of operational range. Nonetheless, if the proportions were to change as was mentioned earlier, then the need for high accuracy in operational range estimation becomes ever more pronounced.

Then in Fig. 5.22 a bar plot is shown to evaluate the average accuracy of both the proposed *offline* and *online* models along with their respective variances. As was expected, the *offline* model tends to over-shoot or under-shoot the true operational range incurring extreme errors with high variance whilst the *online* models tends to attain the true operational data with a very high accuracy and low variance. It must be pointed out here that while traversing on grass using the following settings: $1500mAh@0.952m/s$, both the models show comparable average performance while for $2200mAh@0.952m/s$ the *offline* model performs slightly better. Despite this, the variance of the *offline* model remains higher which can also be confirmed from Fig. 5.19. Overall, the *online* model shows $\approx 60\%$ enhanced accuracy as compared to its adversary for operational range estimation of Rusti V2.0.

5.6.6 Case 2 : UAV

Fig. 5.23 demonstrates the real world performance of the *offline* and *online* estimator models for the case of a UAV.

For *offline* estimation, prior information regarding the operational environment of the robot is required to make meaningful predictions of its operational range. Since there is no prior research explaining how the values of the parameter $k(t)_{env}$ vary, an additional set of 5 experiments (each in varying conditions) were performed and their data was averaged for estimating the value of the parameter $k(t)_{env}$ which was found to be ≈ 0.01. Using this prior information, the maximum operational range was obtained using the *offline* model for the 20 experiments presented here. As the mean estimation error for the *offline* model is about 30 meters in each experiments, with even lower errors at slower speeds, the value of $k(t)_{env} = 0.01$ is claimed to be optimal.

For *online* estimation, the estimate of the net operational range is updated using the auto regressive average of the data acquired in real time. Thus, no prior mission information was deemed necessary to deduce the value of the parameters $k(t)_{env}$ and v. Instead, they are deduced based on real-time mission information. Upon take-off, the drone is initially quite unstable owing to significantly high rotor rpm required for lift-off which then settles to a stable rpm for hovering. During this time, not only the in-house electronics of the drone are unreliable, but also the GPS sensors used needs time to sync with satellite information. This initial instability in the ArDrone and the sensor's data, just after take-off, is what the author calls as the *burn-in* phase. The data of the burn in phase is discarded and the *online* estimation frameworks is activated only upon ArDrone's stabilization. For the purpose of continuous representation in graphs, the data was interpolated during this phase resulting in the initial straight line trends observed during the burn-in phase of the plots. Also, as is evident from the plots, the *online* estimator converges to the true distance as the mission progresses. As more and more mission data becomes available, the estimation performance of the *online*

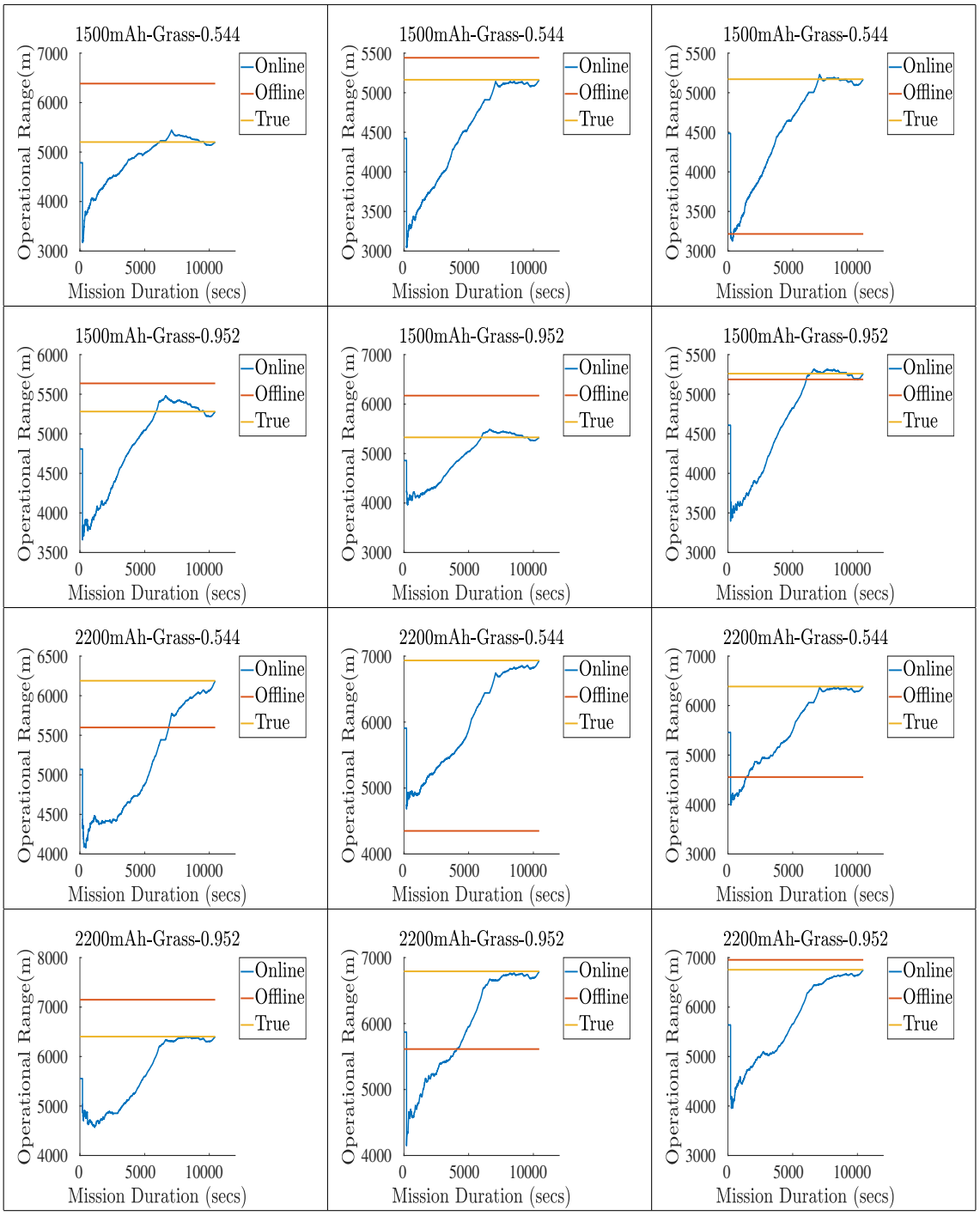


Figure 5.19: Rusti's operational range estimation for grass. Rows $1-2$ *represent experiments performed using* $11.1V$@$1500mAh$ *battery* @$80rpm$ *followed by* @$140rpm$. *Rows* $3-4$ *represents the similar pattern for* $11.1V$@$2200mAh$ *battery respectively.*

Figure 5.20: Rusti's operational range estimation for asphalt. Rows $1-2$ *represent experiments performed using* $11.1V$@$1500mAh$ *battery* @$80rpm$ *followed by* @$140rpm$*. Rows* $3-4$ *represents the similar pattern for* $11.1V$@$2200mAh$ *battery respectively.*

Figure 5.21: Rusti's operational range estimation for tiles. Rows $1-2$ *represent experiments performed using* $11.1V$@$1500mAh$ *battery* @$80rpm$ *followed by* @$140rpm$. *Rows* $3-4$ *represents the similar pattern for* $11.1V$@$2200mAh$ *battery respectively.*

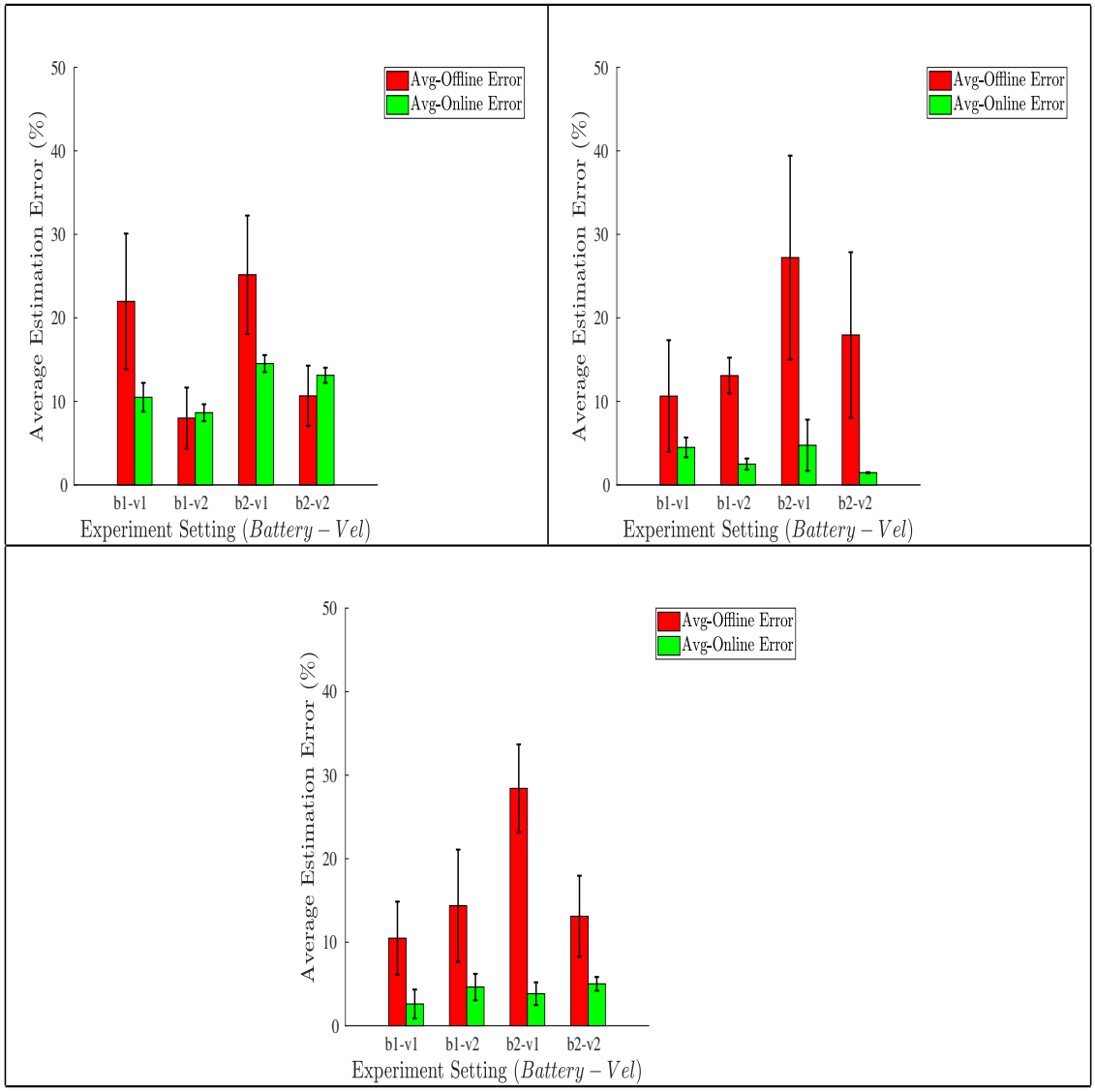


Figure 5.22: Range estimation error for Rusti while traversing on grass, asphalt and tiles respectively. Here $b1, b2$ refer to the 1500mAh and 2200mAh batteries and $v1, v2$ refers to 0.544,0.952 m/sec velocities respectively.

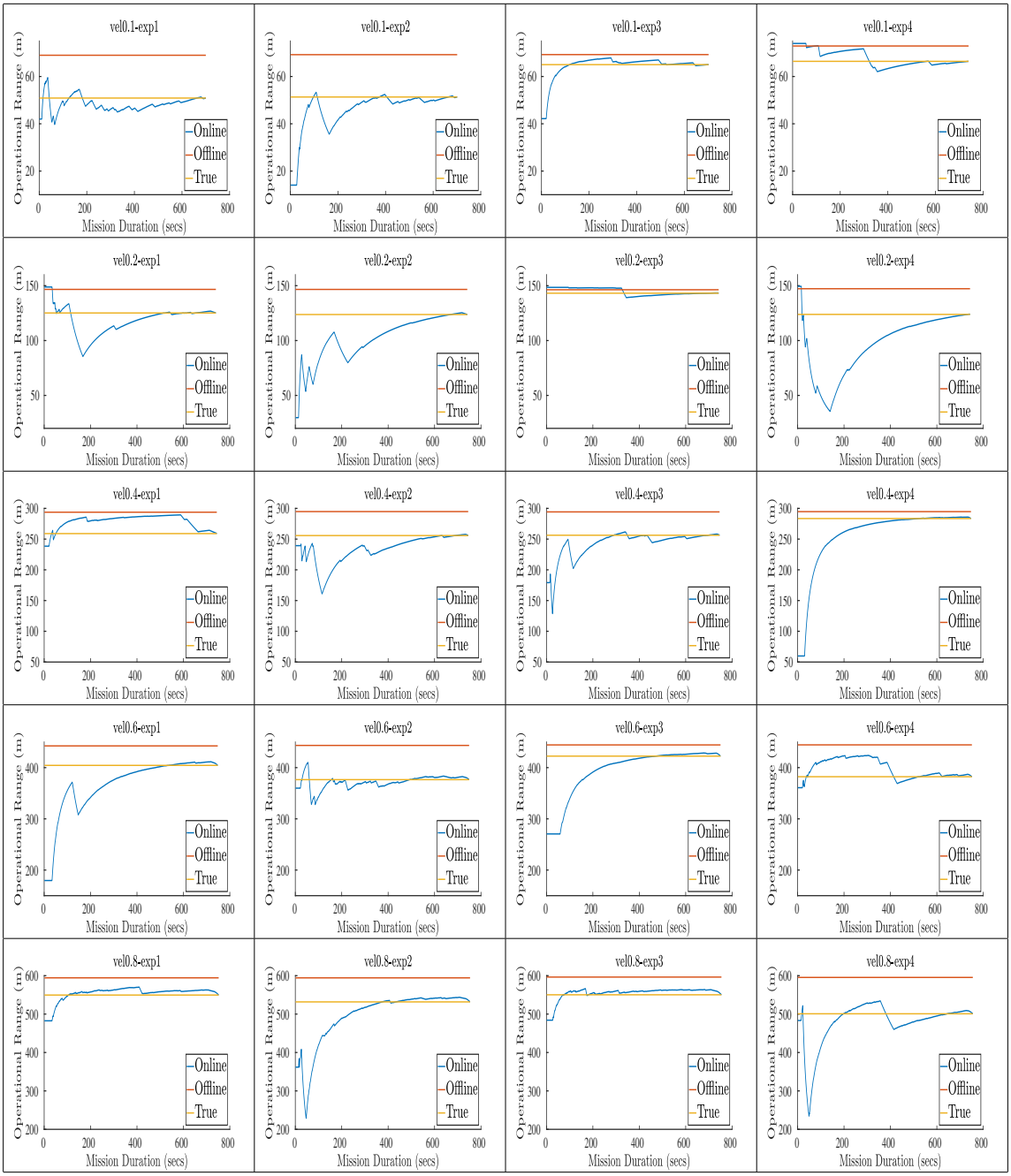


Figure 5.23: ArDrone's Operational Range Estimation. First row represents experiments carried for $v = 0.1$, followed by $v = 0.2, 0.4, 0.6$ and 0.8 m/s respectively. Columns represents multiple experiments at corresponding v. ***N.B.: Results are only comparable across columns owing to different scales of plots across rows.***

model becomes significantly better than its *offline* counterpart.

It must also be pointed out here that the variations in the *online* estimator are quite profound during the early stages of the mission which can be attributed to the fact that the estimator is trying to update its belief with sparse and limited amount of data, but it quickly stabilizes as the amount of data grows. Also, it might seem that increasing the operational velocity (v) always leads to an increase in the operational range (d). However, when the ArDrone attains an operational velocity v_{opt} which is high enough, so much so, that the aerodynamic drag forces acting on the body of the drone is higher than that on the propeller, the theoretical maximum operational range (d_{max}) will be attained and any further increase in the velocity will result in decrease in the operational range. Besides, such high velocities (v_{opt}) are not attainable by current multi-rotor UAVs.

The author would also like to highlight that the variance in the input data (wind compensation angle) being very low, results in low variance in $k(t)_{env}$. This low variance in $k(t)_{env}$, coupled with its low absolute value (1% on average) results in very low variance in predicted distance, often less than a meter. So, for the sake of clear understanding of the readers and legible visual representation of the field trials, the variance in predicted distance was omitted.

Additionally, Fig. 5.24 showcases how the energy stored in the battery is consumed as the mission progresses. An interesting fact to note here is that the trends for both the *hovering* case and *motion* case are quite similar. The reason for this can be attributed to the fact that the value of $k(t)_{env}$ which represents the average excess percentage of thrust that needs to be exerted to maintain stability and velocity, owing to changing environmental conditions, remains below 2%[7]. So, the major component of maneuvering energy is utilized to maintain flight instead of stabilizing the rotorcraft and maintaining its velocity.

Finally, Fig. 5.25 shows bar plots for the operational range estimation performance for both the discussed *online* and *offline* models. For this,the average estimation error of both frameworks for each operational velocity is shown. It can be clearly seen from the graph that the *online* model is $\approx 58\%$ more efficient than its counterpart. Also, to clarify the high *offline* estimation error at $v = 0.1\ m/s$, the author would like to point out that even for small amounts of *hovering* time, the percentage difference in the average velocity and operational velocity is considerably higher than in cases of higher velocities which translates to higher percentage error in prediction using the *offline* model.

5.7 Summary

This chapter addresses one very critical aspect of autonomous navigation ignoring which, could lead to mission failures and render the robots strangulated amidst the field. The problem being addressed can be summed up under the keyword “operational range estimation” for which the author discussed 2 framework in the chapter:

[7]Analyzed from the data obtained from *motion* experiments at varying velocities as shown in Fig. 5.23.

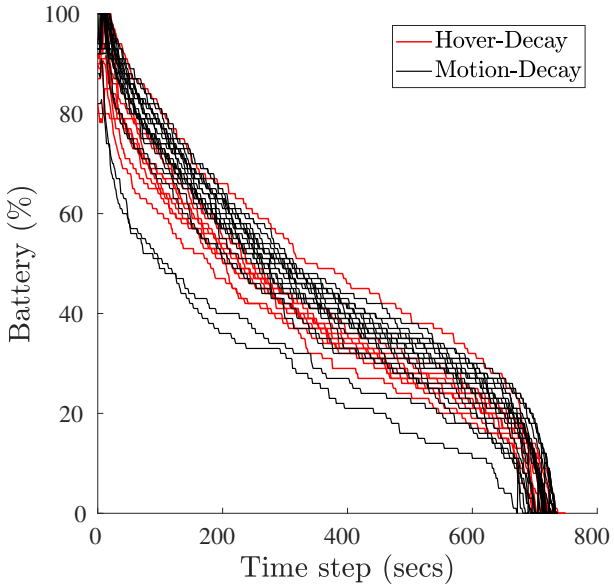


Figure 5.24: Battery Decay for UAV while hovering and motion.

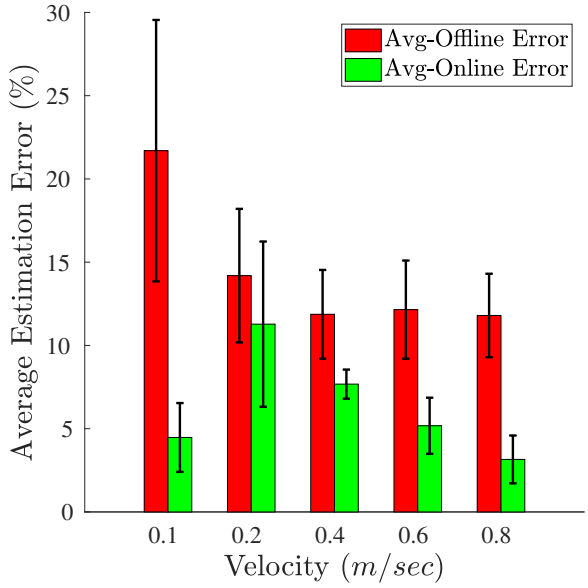


Figure 5.25: Range estimation error. Plot showing error in operational range calculated using the offline and online models along with corresponding standard deviation.

- **Simplified Framework:** This framework was designed explicitly for ground robots operating in smooth terrains with fixed gradient which is usually the case for indoor environments. This framework comprises of two components *viz.,*
  - *Simplified Energy Distribution Model:* explains how the energy is distributed througout the robot and all its components. The maneuvering energy model accounts for planar and elevated terrains while the ancillary energy model includes energy consumed by sensors along with the unwarranted losses. This model can be used to deduce the net energy available for traversal.
  - *Simplified (Offline) Range Estimation Model:* transforms the net traversal energy into operational range and also proposes a theoretical upper bound for it.
- **Generic (Unified) Framework:** This framework was presented as a further enhancement over the simplified variant and encompasses variety of robots operating in myriad environmental conditions (harsh and otherwise). This framework generalizes the models of the simplified models as follows:
  - *Generic Energy Distribution Model:* extends the previous variant of maneuvering energy model to various classes of robots. Additionally, the ancillary energy model now accounts for data transmission rate for short range wireless communications.
  - *Generic Range Estimation Model:* as opposed to previous *offline* model for smooth terrains, this model now has an *offline* variant capable of handling uneven terrains and unforeseen environmental disturbances. Not only this, an *online* model is also proposed to account for sudden changes in the mission profile as they present themselves. Both the extensions were studied in-depth for UGVs and UAVs.

The strengths of the **Generic (Unified) Framework** are highlighted below:

- The unified framework is equally applicable to both commercial and custom-built robots alike, provided, additional sensors can be incorporated to log the appropriate data.
- The concept of duty cycle proposed herewith, brings this model really close to real-world scenarios making the framework applicable without hassles.
- Having obtained average accuracy of almost 93.87% with the online variant and 82.97% with the offline variant, it is safe to conclude that framework is by far the state-of-the-art operational range estimation framework for all robots that may be considered for field trials.

All that remains now, is to couple this framework with energy efficient path planners and then the robots can be guaranteed to return to base station by the end of their mission (not accounting for impromptu hardware failures).

Bibliography

[1] D. Panigrahi, C. Chiasserini, S. Dey, R. Rao, A. Raghunathan, K. Lahiri, *et al.*, "Battery life estimation of mobile embedded systems," in *VLSI Design, 2001. Fourteenth International Conference on*, pp. 57–63, IEEE, 2001.

[2] F. Zhang, G. Liu, and L. Fang, "Battery state estimation using unscented kalman filter," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pp. 1863–1868, IEEE, 2009.

[3] M.-H. Chang, H.-P. Huang, and S.-W. Chang, "A new state of charge estimation method for lifepo4 battery packs used in robots," *Energies*, vol. 6, no. 4, pp. 2007–2030, 2013.

[4] Q. Miao, L. Xie, H. Cui, W. Liang, and M. Pecht, "Remaining useful life prediction of lithium-ion battery with unscented particle filter technique," *Microelectronics Reliability*, vol. 53, no. 6, pp. 805–810, 2013.

[5] L. Liao and F. Köttig, "Review of hybrid prognostics approaches for remaining useful life prediction of engineered systems, and an application to battery life prediction," *IEEE Transactions on Reliability*, vol. 63, no. 1, pp. 191–207, 2014.

[6] A. Abdilla, A. Richards, and S. Burrow, "Endurance optimisation of battery-powered rotorcraft," in *Conference Towards Autonomous Robotic Systems*, pp. 1–12, Springer, 2015.

[7] A. Sadrpour, J. J. Jin, and A. G. Ulsoy, "Mission energy prediction for unmanned ground vehicles using real-time measurements and prior knowledge," *Journal of Field Robotics*, vol. 30, no. 3, pp. 399–414, 2013.

[8] K. Tiwari, X. Xiao, and N. Y. Chong, "Estimating achievable range of ground robots operating on single battery discharge for operational efficacy amelioration," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3991–3998, IEEE, Oct 2018.

[9] Y. Mei, Y.-H. Lu, Y. C. Hu, and C. G. Lee, "Energy-efficient motion planning for mobile robots," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 5, pp. 4344–4349, IEEE, 2004.

[10] D. Brooks, V. Tiwari, and M. Martonosi, *Wattch: A framework for architectural-level power analysis and optimizations*, vol. 28. ACM, 2000.

[11] O. Tremblay, L.-A. Dessaint, and A.-I. Dekkiche, "A generic battery model for the dynamic simulation of hybrid electric vehicles," in *Vehicle Power and Propulsion Conference, 2007. VPPC 2007. IEEE*, pp. 284–289, IEEE, 2007.

[12] R. Mur-Artal, J. M. M. Montiel, and J. D. Tards, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, pp. 1147–1163, Oct 2015.

[13] K. Tiwari, X. Xiao, A. Malik, and N. Y. Chong, "A unified framework for operational range estimation of mobile robots operating on a single discharge to avoid complete immobilization," *Mechatronics*, Oct. 2018. In Press.

[14] S. W. Kim and Y. H. Lee, "Combined rate and power adaptation in ds/cdma communications over nakagami fading channels," *IEEE Transactions on Communications*, vol. 48, no. 1, pp. 162–168, 2000.

[15] M. Gatti, F. Giulietti, and M. Turci, "Maximum endurance for battery-powered rotary-wing aircraft," *Aerospace Science and Technology*, vol. 45, pp. 174–179, 2015.

[16] A. Abdilla, A. Richards, and S. Burrow, "Power and endurance modelling of battery-powered rotorcraft," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 675–680, Sept 2015.

[17] Y. Mei, Y.-H. Lu, Y. C. Hu, and C. G. Lee, "Energy-efficient motion planning for mobile robots," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 5, pp. 4344–4349, IEEE, 2004.

[18] H. Sato, "Moving average filter," Oct 2001. US Patent 6,304,133.

[19] D. Graupe, A. A. Beex, and G. D. Causey, "Arma filter and method for designing the same," Feb 1980. US Patent 4,188,667.

[20] Y. Mei, Y.-H. Lu, Y. C. Hu, and C. G. Lee, "A case study of mobile robot's energy consumption and conservation techniques," in *Advanced Robotics, 2005. ICAR'05. Proceedings., 12th International Conference on*, pp. 492–497, IEEE, 2005.

[21] R. Manduchi, A. Castano, A. Talukder, and L. Matthies, "Obstacle detection and terrain classification for autonomous off-road navigation," *Autonomous robots*, vol. 18, no. 1, pp. 81–102, 2005.

[22] S. Thrun, W. Burgard, and D. Fox, "Probabilistic robotics. 2005," *Massachusetts Institute of Technology, USA*, 2005.

Part III

Reduce Phase: Model Fusion

Chapter 6

Fusion of Distributed Gaussian Process Experts *(FuDGE)*

> *“ Information never hurts, but whom do we trust ? ”*
>
> Kshitij Tiwari, *2017*

In Chapter 4, the fully decentralized active sensing framework called *RC-DAS* was discussed, which is suitable to disconnected multi-robots teams. In doing so, multiple models of the environment were obtained which may have slightly conflicting estimates about the internal dynamics of the environment. This is due to the fact that every robot could only observe part of the field which may not provide enough training samples to generalize the dynamics over those regions that are far away. In order to resolve such conflicting local models, the author now discusses a novel fusion technique to fuse all local models into one globally consistent model which can now be inferred as the representation of the overall dynamics of the environment. The objective now is:

> *Given multiple models of environmental dynamics, which model should be trusted?*

6.1 Various Notions of Fusion

The problem stated above is referring to a *many-to-one* mapping dilemma wherein each robot tries to generate a model which it thinks is accurate but having obtained M models from M robots, should one or all of them be selected? If one had to be chosen, then the information acquired by the others would go in vain, but if all were retained, then the underlying environmental dynamics cannot be represented until one global model is contructed. To solve this problem, a pointwise fusion of distributed GP experts or *FuDGE* [1] is discussed in this chapter.

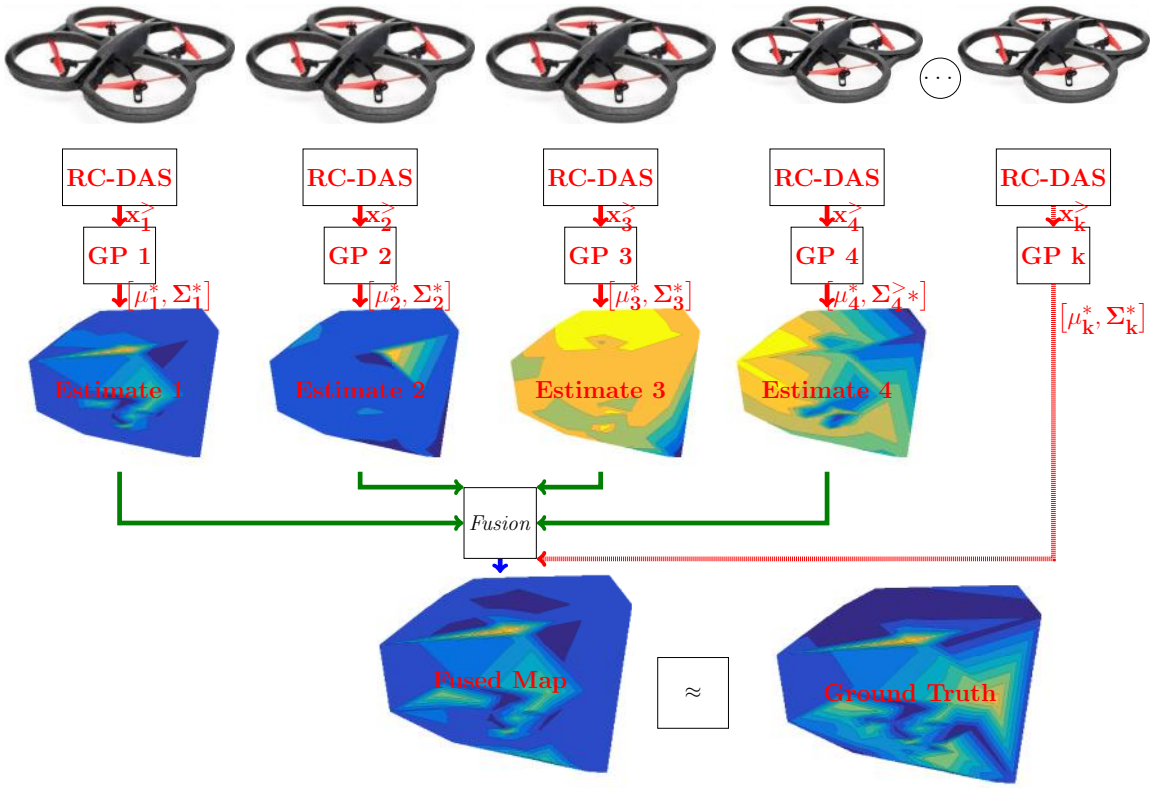


Figure 6.1: Sensing Scenario. Illustration of the sensing scenario in which the team of mobile robots operates under resource constraints. The aim is to gather optimal observations to make a prediction for the environment defined by posterior mean μ_m^* and posterior covariance Σ_m^*. Estimate 1 – Estimate 4 represent the 4 individualistic prediction maps made by the 4 robots based on their training samples. $x_m^>$ represents the next-best-location chosen by the RC-DAS active sensing for the m^{th} expert. Fused Map is the globally consistent fused prediction map generated by using the proposed fusion framework. The objective is to make the Fused Map as similar to the Ground Truth as possible. These maps have been interpolated for ease of visualization. In reality, we just have a discrete collection of predicted measurements at pre-determined locations. Figure based on [1].

Similar works in the domain of applied machine learning use the term *"fusion"* to combine multiple sets of heterogeneous sensor data using GPs as discussed in [2–5]. In the context of multiple sensors mounted on robots, the state estimation can be done effectively by *"fusion"* of noisy information provided by various sensors using Kalman filters [6–8]. Alternatively, the term *"fusion"* in the machine learning literature is used to define an ensemble of probabilistically fused prediction estimators [9–14], which is the notion that this work will be adopting. This work can be positioned at the junction of machine learning and robotics literature and the author intends to use the term *"fusion"* to refer to a probabilistic amalgamation of various individually trained unbiased estimators wherein each robot itself behaves as such.

6.2 Existing Fusion Approaches

Existing *model fusion* techniques from the literature can be broadly classified into two main categories. *First* category is models can be called the *Product of Expert (PoE)* models like the *Bayesian Committee Machine (BCM)* [15] and *generalized Product of Experts (gPoE)* [11]. In the BCM framework, multiple independent GP experts are trained on subsets of the whole training dataset, and their confidence is evaluated based on the reduction in uncertainty over the test points. Although this approach is promising in terms of distributing the computational load of a single GP over multiple GP experts, it is not feasible for a real robot implementation. The reason for this shortcoming is that, this approach works only under the assumption that all GP experts are "jointly trained" such that they "share the same set of hyper-parameters" [9]. Doing so, from a machine learning perspective *i.e.,* implementing on a work station with sufficient computation power is feasible and can be realized *e.g.,* by deploying multiple threads, each training a local GP model over the subset of training data in a synchronized fashion and sharing the same set of hyper-parameters. The global model can be hierarchically combined or the same can be done in one pass. On the other hand, for a real robot team this would require precise time synchronization between all memebers and an all-to-all synchronized communication (as was used in the recent work [16]) in order to ensure joint training over subsets of dataset. This problem can be easily tackled using the *generalized Product of Experts (gPoE)* models from [11], wherein the fusion is carried out over independent GP experts while their contributions are determined, *e.g.,* by their respective differential entropy scores *c.f. Definition 6.1.* Both of the above models are *log opinion pool models* but BCM model ensures consistency in the sense that predictions are guaranteed to fall back to the prior when the testing data points fall significantly far away from the training data.

Second category can be called as the *Mixture of Expert (MoE)* models [12–14] wherein each GP expert specializes in different partitions of the state space and the mixture ensemble automatically allocates the expert its corresponding specialist zone. This model is a *linear opinion pool* of experts where the weights are given by input-dependent gating functions. In order to design apt gating functions, some hints can be taken from the neural network literature [17–19], which introduced a point-wise

locally weighted fusion (LWF) technique to evaluate the performance of a predictor over a neighborhood around the probe-point. However, these approaches require a sufficiently dense training dataset with access to ground truth. Hence, they cannot be applied directly in a real robotic setup wherein the robot never knows the ground truth. Even after visiting an observing a certain location, the robot only acquires a noisy variant of the ground truth.

Beyond the above mentioned solutions, there are other solutions in the literature that deal with multi-agent decentralized exploration like [20] wherein a Dirichlet Process Mixture of GP experts is used to model a decentralized ensemble of GP experts. In this approach, the requirement of a control parameter α that manages the addition of a new cluster, enforces the need of supervision (by base node or human opertors) that can control and instruct a new member to be added to the team when a new cluster is created.

Aside from the above two category of model fusion, there are other stand alone researches which do not fall under any category. For instance, an alternative solution was proposed for multiple GP experts for decentralized data fusion by the authors in [10]. This work does not belong to any of the two categories summarized above, and, in this work, the robots share the measurements gathered with their nearest neighbors using consensus filtering.

6.3 Limitations of Existing Works

Robots may need to operate in harsh environments where peer-to-peer and peer-to-base communication channels are unrealiable or sometimes even costly in terms of transmission costs (power consumption, latency *etc.*). In such scenarios, sharing the information with the peers is infeasible. Besides, if the sensing area to be monitored is significantly large, then there is a high likelihood that the peers may never meet each other and in such settings the solutions proposed by [10] would not suffice. Similarly, when a multi-robot team is tasked with observing a target phenomenon of interest, members are usually not swapped or added / removed dynamically while the team is actively exploring. This renders the works of [20] ineffective. As opposed to these, the author suggests an iterative weighted fusion technique suitable for GPs that allows us to evaluate the proximity of a probe (test) point to the training samples of GP experts while evaluating the confidence of each expert.

Definition 6.1 (Differential Entropy Score)**.** *Let $\sigma^2_{**}(x)$ represent the prior variance and $\sigma^2_{*}(x)$ represent the posterior variance over a certain location of interest x. The the differential entropy score is given by the difference in the differential entropies :*

$$\text{Score} = 0.5 \times (\log \sigma^2_{**}(x) - \log \sigma^2_{*}(x)) \tag{6.1}$$

6.4 Predictive Model Fusion for Distributed GP Experts *(FuDGE)*

At the end of exploration (mission time) of all members of the mobile robot team, M diverse GP experts are acquired, which were each trained on their respective subsets of training data and have generated a predictive map over the entire target phenomenon. To fuse the predictions from multiple models into one globally consistent model, first, a *consistency check* is performed, which involves finding the probe (test) locations that are shared by all the GP experts.

For this, let $U_{global} \triangleq \{U_1 \cap U_2 \cap \ldots \cap U_M\}$ represent the super set of all unobserved nodes that were never visited by any robot. Similarly, let $O_{global} \triangleq \{O_1 \cup O_2 \cup \ldots \cup O_M\}$ define the super set of all observed nodes that were visited by all robots.

6.4.1 Fusion Strategy

Let a probe point be represented by $Q \in U_{global}$ and defined as a point of interest for which the predictions from multiple GP experts must be fused. The fusion algorithm is defined next.

6.4.1.1 Pointwise Mixture of Experts using GMM

In what follows, first, a premise of the fusion algorithm is outlined followed by the detailed description of the fusion algorithm itself.

Premise: GPs are kernel based methods as was explained in Chapter 3, the author utilizes isotropic squared exponential kernels as shown in Eq. (3.1). By definition, the correlation between locations decays exponentially as the spatial separation increases. Thus, the predictions are made with highest confidence nearby the observed locations and the confidence drops gradually as the distance increases [21]. This is also supported by Tobler's first law of geography which states that: "Everything is related to everything else, but near things are more related than distant things" [22].

Model Description: Having laid down the premise of the model, the author believes that the reader(s) have a good intuition about the nature of correlations (given by Eq. (3.1)) in environment monitoring phenomenon. Thus, now is the right time to introduce the model fusion technique hereby referred to as *FuDGE*. For this, first, independently[1] trained GP experts are obtained by utilizing the distributed GP framework from [23] and running the RC-DAS information acquisition function. Then, during the test phase, the expert predictions need to be combined based on the proximity of a test (probe) point to the experts' training samples. Thus, on the lower level independent prediction models are deduced and on the higher level a fused globally consistent model is obtained making this a $2-$layer model.

The length scales inferred by the GP experts represent the standard deviation in the spatial variation of measurements along the i^{th} input dimension σ_i. A probe point

[1] not the same as conditional independence. Just refers to individual models maintained by each expert

Q lying too far[2] away from the training points of the m^{th} expert will not be predicted confidently by the m^{th} GP. This is attributed to the fact that a stationary squared exponential covariance kernel from Eq. (3.1) was used to model the environment. Using this covariance structure, it was inferred that correlation in measurements at two locations x and x^* will decay as the spatial separation between them increases as was also explained previously in the premise of the model. Thus, a multivariate Gaussian distribution can be placed over the $O_m \sim \mathcal{N}\left(Q|O_m^j, \Sigma_m\right)$ where j represents the j^{th} training sample of the m^{th} expert and $\Sigma_m \triangleq \text{diag}(l_{lat}^2, l_{long}^2)$. The spread of the multivariate normal distribution is defined in terms of length scales along the *Latitude* and *Longitude* of the corresponding GP expert. This gives rise to one Gaussian mixture model (GMM) over the training data points of each GP expert. The responsibilities of this hierarchical GMM are then defined as:

$$\log p(m|Q, O_{global}) \triangleq \Sigma_{x_i} \log p(Q|x_i, \Sigma_m) \tag{6.2}$$

In (6.2), x_i refers to $[O_{global}]_i$, Σ_m refers to the covariance of the Gaussian distribution for the m^{th} GP expert and $p(Q|x_i, \Sigma_m) = \mathcal{N}(x_i, \Sigma_m)$. This is illustrated in Fig. 6.2 where all the locations that were unvisited by the robot (m) during its exploration are referred to as the test set for that robot U_m. The responsibilities of a hierarchical Gaussian mixture model in Eq. (6.2) are such that $\log p(m|Q, O_{global}) \in [0, 1]$ and $\Sigma_{m=1}^{M} \log p(m|Q, O_{global}) = 1$. Then, the fused prediction at probe point Q can be represented as the weighted fusion of predictions from all models as:

$$\mu_{Q|O_m,\boldsymbol{\theta}_m} \triangleq \Sigma_{m=1}^{M} \left(\log p(m|Q, O_{global})\mu_m^Q\right) \tag{6.3}$$

In (6.3), the fused prediction at probe point Q is defined as the sum of predictions (μ_m^Q) weighted by the sum of log-responsibilities of a GMM $(\log p(m|Q, O_{global}))$ for each expert $m \in \{1, \ldots, M\}$.

Additionally, the confidence of the fused estimator at the probe point Q can be explained by the net variance at the probe point Q as follows:

$$\begin{aligned}
\sigma_{Q|O_m,\boldsymbol{\theta}_m} \triangleq & \Sigma_{m=1}^{M} \left\{\log p(m|Q, O_{global})[(\sigma_m^Q)^2 + (\mu_m^Q)^2]\right\} - \\
& \left(\mu_{Q|O_m,\boldsymbol{\theta}_m}\right)^2, \\
= & \Sigma_{m=1}^{M} \left(\log p(m|Q, O_{global})(\sigma_m^Q)^2\right) + \\
& \Sigma_{m=1}^{M} \left(\log p(m|Q, O_{global})(\mu_m^Q)^2\right) - \\
& \left(\mu_{Q|O_m,\boldsymbol{\theta}_m}\right)^2.
\end{aligned} \tag{6.4}$$

Since $(\cdot)^2$ is a convex operator, using the Jensen's inequality [24], we know that $\Sigma_{m=1}^{M} \left(\log p(m|Q, O_{global})(\mu_m^Q)^2\right) \geq (\mu_{Q|O_m,\boldsymbol{\theta}_m})^2$. Now, Eq. (6.4) can be interpreted as the weighted combination of variances of the components plus a correction term which is always positive. The correction term accounts for the divergence of respective component means (μ_m^Q) from the mean of the mixture $(\mu_{Q|O_m,\boldsymbol{\theta}_m})$ for the probe point Q.

[2]outside the 99.5% confidence bound

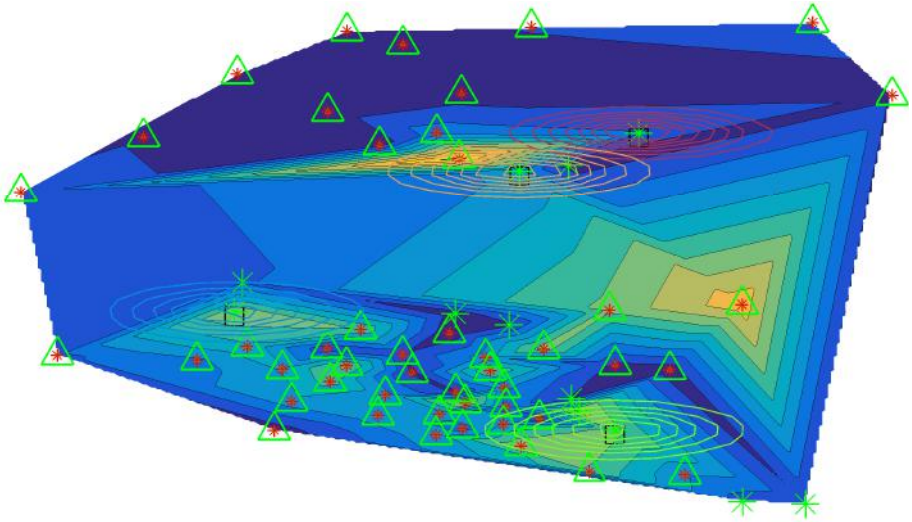


Figure 6.2: FuDGE. Illustration of weighted fusion performed using FuDGE by positioning a $2D$ *Gaussian distribution* $\mathcal{N}(x_i, \Sigma_m)$ *to evaluate the responsibility of a GP expert over a probe point. In this Figure, locations marked in green asterisk (*) represent the training locations that were visited by the robots during their respective missions, while those highlighted by red asterisk (*) represent the probe points over which the predictions are to be fused and black squares (□) represent the start location of each of the* 4 *robots. For ease of visualization, only the first training sample of each of the GP expert is shown and the process is iteratively carried out over all query points. For this illustration,* 4 *experts were considered, each of which are represented by a Gaussian contour plot centered around their first training sample respectively.*

Algorithm 3 FuDGE ($\forall_{m\in M}\mu_m, \forall_{m\in M}\theta_m, U_{global}, \forall_{m\in M}O_m$)

1: **Input**:
- $\forall_{m\in M}\mu_m$: predictions from all robots
- $\forall_{m\in M}\boldsymbol{\theta}_m$: hyper-parameters from all robots
- $U_{global} \triangleq \{U_1 \cap U_2 \cap \ldots \cap U_M\}$: locations for fusing predictions
- $\forall_{m\in M}O_m$: Observations from all robots

2: **Output**:$\forall_{Q\in U_{global}}\ \boldsymbol{\mu}^Q$: *FuDGE Predictions*
3: **for** $\forall Q \in U_{global}$ **do**
4: **for** each robot m **do**
5: Σ_m =DIAGONAL($\theta_m^2[1], \theta_m^2[2]$) ▷ Construct variance matrix using spatial length scales
6: $L_m^Q = \sum_{\forall x_m \in O_m} \log(p(Q|x_m, \Sigma_m))$ ▷ Compute the *Responsibility*
7: **end for**
8: Normalize L_m^Q such that $\sum_{\forall m\in M} L_m^Q = 1$
9: $\mu_Q = \sum_{\forall m\in M} L_m^Q \mu_m^Q$ ▷ Fuse the weighted predictions from all robots
10: **end for**
11: **return** $\boldsymbol{\mu}^Q$

The *FuDGE* approach is summarized in Alg. 3 and the details are as follows: The algorithm requires posterior estimates from all experts ($\forall_{m\in M}\mu_m$) along with their respective hyper-parameters ($\forall_{m\in M}\theta_m$) generated based on corresponding observations ($\forall_{m\in M}O_m$). Upon performing consistency check, the set U_{global} is obtained. Then, for each probe point $Q \in U_{global}$ (line 3), all experts are queried to obtain their learnt hyper-parameters and the covariance matrix is generated in line 5. Similarly, the responsibility of each expert is obtained in line 6 by using Eq. (6.2). The responsibilities are then normalized to transform them into weights in line 8. Finally, a weighted summation is performed to obtain the fused prediction for the current probe point Q as shown in line 9. After iterating over all probe points, a list of fused predictions $\boldsymbol{\mu}^Q$ is returned in line 11.

6.4.1.2 Generalized Product of Experts Model [11]

This ensemble predicts the measurement at a test point as a weighted product of predictions from all the experts for the said test point. The gPoE model allows flexibility in the definition of weights (confidence) of each expert which are adjusted based on the importance of an expert [9]. In the original work, a differential entropy score was used to define the weight of the experts based on the improvement in information gain between the prior and the posterior *c.f. Definition 6.1.* Following this definition, this work also defines the weights (β_m) of the m^{th} expert and fused predictions generated by an ensemble of M GP experts are obtained as follows:

$$\beta_m = \frac{1}{2}(\log(\sigma^2_{m_{**}}) - \log(\sigma^2_m(x^*))) \tag{6.5}$$

$$\hat{\beta_m} = \frac{\beta_m}{\Sigma_m \beta_m} \tag{6.6}$$

$$\mu^{gPoE}_{U|O,\Theta} \triangleq \Sigma^{gPoE}_{UU|O,\theta_m} \sum_{m=1}^{M} \hat{\beta_m} \Sigma^{-1}_{UU|O_m,\theta_m} \mu_{U|O_m} \tag{6.7}$$

$$(\Sigma^{gPoE}_{UU|O,\Theta})^{-1} \triangleq \sum_{m=1}^{M} \hat{\beta_m} \Sigma^{-1}_{UU|O_m,\theta_m} \tag{6.8}$$

In (6.5), the differential entropy score is defined based on Definition 6.1 and in (6.6) the confidence weight per probe point x^* is evaluated by finding the differential entropy between the prior variance $\sigma^2_{m_{**}}$ and posterior variance $\sigma^2_m(x^*)$ for the probe point x^* such that $\Sigma_m \hat{\beta}_m = 1$. The limitation however is that this model is over-conservative and often over-estimates the variance. Additionally, there is no correction term in the variance to rectify over-estimation and thus, the author proposed a novel fusion technique discussed above which caters to such limitations.

6.4.1.3 Multiple mobile sensor nodes generating single GP

Given the availability of a fusion center with sufficient processing capabilities that can fuse the models of all the robots, an obvious question then arises:

> *What happens if the multiple robots were simply considered as mobile sensor nodes, each tasked with just gathering observations while the fusion center acquires all observations and makes a single GP directly?*

Using the mobile robot team simply as sensor nodes gathering observations for the sink node (fusion center) instead of modeling the environmental dynamics has the following limitations:

- This would lead to a paralelly connected topology wherein all agents are in direct contact with the fusion center (base station). This transforms the exploration phase itself from disconnected- decentralized to paralelly connected-centralized architecture. As such, such the base station fail at some point, the entire team will get strangulated and the model will be completely lost.
- Acquiring data from all agents in real-time would be an additional challenge and would increase the computational time of the map phase.
- If the single GP is directly created at the end of the missions of all agents then the robots would be performing active sensing based on fixed hyper-parameters which cannot be updated as more data is being acquired.
- The computational complexity is again cubic in the size of the data acquired by the entire team which is much larger than the current setting.
- Potential congestion
- Vulnerability in case of failure of sink node

6.5 Map-Reduce Gaussian Process *(MR-GP)* Framework

The active sensing architecture along with the fusion mechanism can all be assimilated into one sequential framework as shown in Fig. 6.3. Thus, first the *RC-DAS* objective function can be used to acquire training samples and make one model per robot. Similar procedure is repeated for all M robots each behaving like a self-sustaining GP expert in a fully decentralized setting. This is referred to as *Map* phase. Upon termination of the mission of all robots and successful retrieval of all M robots at the base station, a one-shot fusion of all models can be performed to obtain a globally consistent model. This stage is referred to as the *Reduce* phase and hence the overall framework is called Map-Reduce Gaussian Process *(MR-GP)*. The requirement for the robots to return to the base station for the fusion procedure necessitates the need for operational range estimation (discussed in Chapter 5) and information acquisition mechanisms like *RC-DAS*† (discussed in Chapter 4). This not only allows models to be fused at the end, but also ensures that no information acquired by any agent goes in vain.

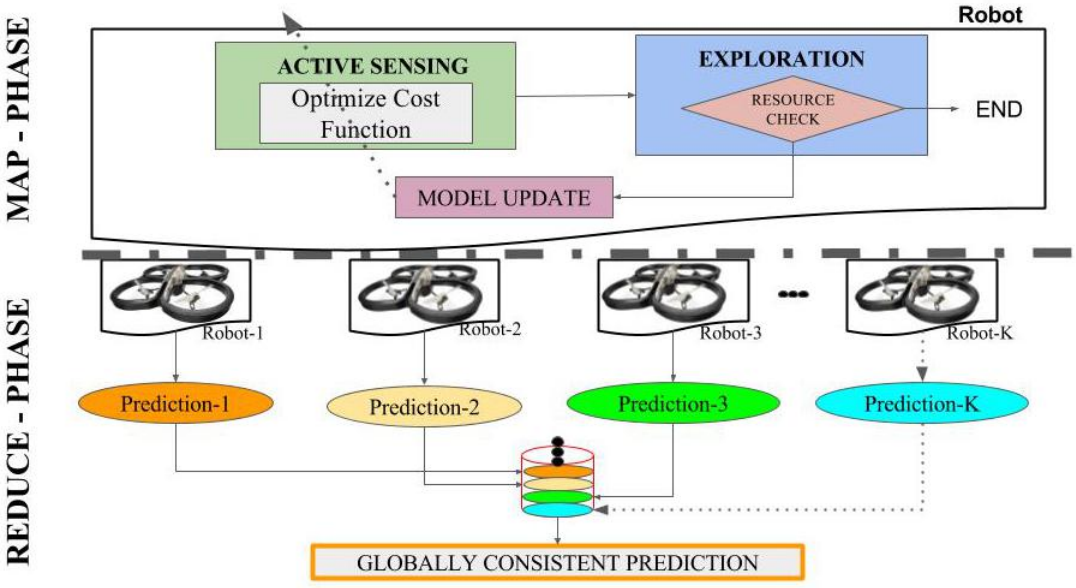


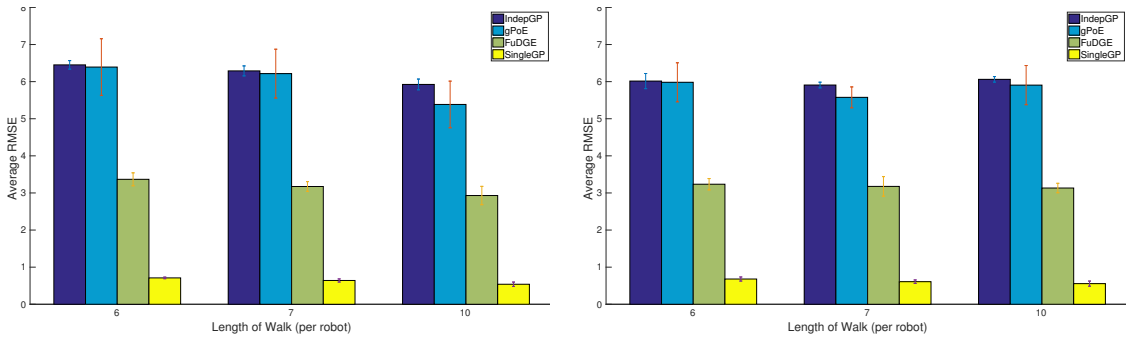
Figure 6.3: MR-GP framework. Our sequential architecture for Map phase and Reduce phase. During the Map phase each robot (GP-expert) generates is individual model and tries to optimize it as far as possible. Upon mission termination from all members of the team, during the Reduce phase, the base station, performs a point-wise weighted fusion of all models to obtain a single globally consistent model. The performance of the fused model is directly influence by the quality of each individual model.

6.6 Experiments

For empirical analysis, the **USA Ozone dataset** is used like earlier. Based on the dataset, the model fusion quality is evaluated in unison with the variety of active sensing schemes discussed thus far. **N.B.: Since NN gathers only correlated observations, the author omitted the analysis with respect to nearest neighbors. Only *RC-DAS* and *full-DAS* are considered here.**

6.6.1 Fusion quality

In this section, the average RMSE is assesssed which represents the average of the errors of all robots between the estimated model and the ground truth evaluated over each element of U_{global}.



(a) Performance of FuDGE using full-DAS *(b) Performance of FuDGE using RC-DAS*

Figure 6.4: Fusion Performance. Evaluating average fusion performance for full-DAS and RC-DAS v/s Length of Walk [Ozone Dataset]

Fig. 6.4a shows the fusion performance of *full-DAS* against the average performance of independent robots labeled as *IndepGP*, the state-of-the-art *gPoE* and the single GP case evaluated over U_{global}. *IndepGP* refers to the average of individual performances of all robots as evaluated over U_{global}. As explained earlier on, this does not mean that the M GPs are conditionally independent of each other, since they might have had shared training samples owing to uncoordinated exploration but independence here is used in the sense of uncoordinated individual GP expert models. It can be seen that the independent robots tend to incur higher (average) performance error owing to limited exploration. This error tends to go down as more observations become available. However, fusion strategies outperform the independent robot models. By comparing Fig. 6.4a and Fig. 6.4b, it can be observed that *full-DAS* tends to perform better than *RC-DAS* since in this case, the GPs had access to the most uncertain and hence the most informative training samples. This also helps the fusion model perform better as some of the experts tend to know slightly more information about a region as compared to the others, and hence not all experts can be assigned equal weights. From Fig. 6.4a and Fig. 6.4b, it can be observed that the average fusion performance of our proposed model is always the best. In essence, the *FuDGE* can be considered a

Simple Averaging when the GP experts are equally good (or bad) at predictions for a probe point while at other times, *FuDGE* assigns the weights to GP experts based on the log-likelihood (responsibility) of the GP for the probe point. Another interesting fact to note here is that, while the error of all fusion techniques for *full-DAS* tends to reduce with the increase in the number of observations (length of walk) of each robot, the error does not follow a monotonically decreasing trend for *RC-DAS* owing to the choice of training samples as explained earlier on. Moreover, the *FuDGE* and *gPoE* are approximations of a single GP utilized to assist with efficient robot exploration. Thus, they incur a slight compromise in accuracy.

In order to guarantee the statistical significance of the author's claims, the *p-values* [25] were calculated for the experiments. For this, the null hypothesis, H_0:*FuDGE* does not perform better than *gPoE* and the alternative hypothesis, H_a: *FuDGE* performs better than *gPoE* were defined. Then, for *full-DAS* and *RC-DAS*, the *p-values* of the *z-statistic* for the right-tailed test were evaluated as 0.0294 and 0.0090 respectively. The significance level of $\alpha = 0.05$ was selected and since $p < \alpha$ for both active sensing techniques, strong evidence against the null hypothesis allows the null hypothesis to be rejected. Thus, the performance of *FuDGE* is significantly better when compared to that of *gPoE*.

6.6.2 Path length

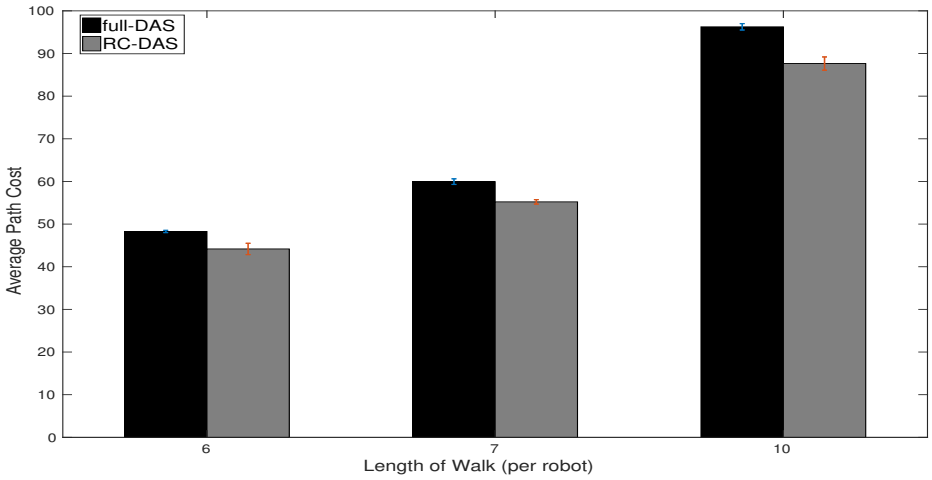


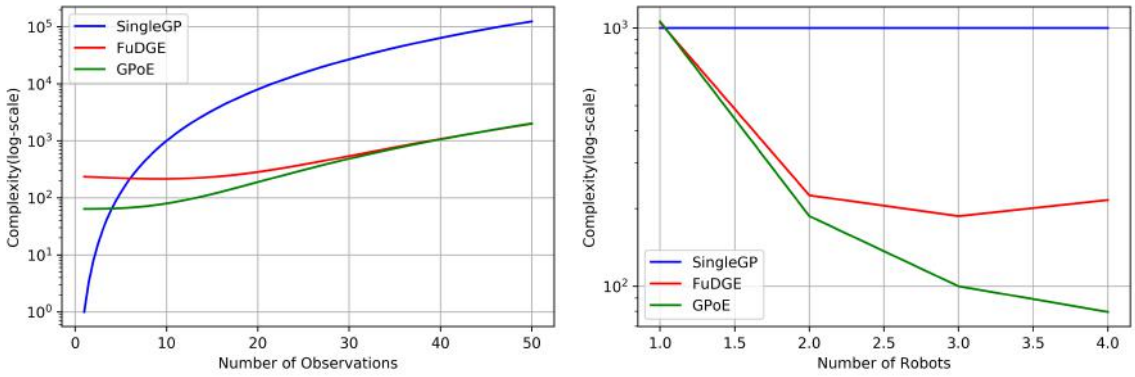
Figure 6.5: Path cost. Evaluating path cost for full-DAS and RC-DAS v/s Length of Walk [Ozone Dataset]

Since the active sensing schemes (informative path planning schemes discussed in Chapter 4) assist the data collection that eventually leads to map fusion, it is essential to also analyse the path cost incurred by both active sensing schemes, whereby the path cost refers to the net *sensing cost* and *traveling cost* incurred by a robot during its exploration. The results are summarized in Fig. 6.5 which shows the average past cost representing the average of total path costs incurred by all robots. It can be seen

that the costs incurred by *full-DAS* are consistently higher than that of *RC-DAS*. This goes to satisfy the author's claim that model performance can be successfully traded-off to efficient resource utilization without having to drastically compromise on any one of them.

In conclusion, it is apparent that both *full-DAS* and *RC-DAS* attain similar performance in terms of fusion quality but *RC-DAS* does so at lower path costs.

6.6.3 Computational Complexity



(a) Complexity v/s Length of Walk *(b) Complexity v/s Number of Robots*

Figure 6.6: Computational Complexity. Illustrating the computational complexity of singleGP, FuDGE and GPoE models.

From Fig. 6.4, it is explicit that *FuDGE* outperforms existing state-of-art models but it is also essential to analyze that the performance was not obtained at the cost of extensive computations. In order to perform a fair comparison between the referred models, the trajectories of all robots were stored a priori using *RC-DAS* information acquisition function. Then, the same trajectories are fed to all models to simulate real robot exploration and this cost of exploration was accounted as if the model was performing exploration in real time. Let M represent the size of the team operating in the field whose domain is D as before. The results hence obtained are summarized in Table 6.1 and the instance description used herewith are summarized in Table 6.2.

Table 6.1: Computational complexity analysis for FuDGE, GPoE and SingleGP.

	Inference Complexity	**Exploration Complexity**	**Fusion Complexity**
SingleGP	$I_S = \mathcal{O}(S)^3$	$E_S = \mathcal{O}(R_S)$	$F_S = \emptyset$
FuDGE	$I_F = \mathcal{O}(\frac{S}{M})^3$	$E_F = \mathcal{O}(R_F M)$	$F_F = \mathcal{O}(FM)$
GPoE	$I_G = \mathcal{O}(\frac{S}{M})^3$	$E_G = \mathcal{O}(R_G M)$	$F_G = \mathcal{O}(G + M)$

In Table 6.1, the computational cost is categorized using 3 components: all costs referenced with I_* refer to the cost for performing GP inference, E_* refer to the computational cost for active sensing and F_* refer to the computational cost for fusion.

Table 6.2: Instances used for computational complexity analysis.

Instance	Description
S	$\#(O_{global})$
R_S	$\#(D \setminus O_{global})$
F	$\#(U_{global})$
R_F	$\#(D \setminus O_m)$
G	$\#(D)$
R_G	$\#(D \setminus O_m)$

To ease the understanding of the readers, a visual representation of Table 6.1 is also shown in Fig. 6.6 wherein the model complexities are analyzed with the growing number of observations for a fixed size of team (Fig. 6.6a) and also the impact of variable size of team (Fig. 6.6b). From Fig. 6.6a, it is clear that both *FuDGE* and *GPoE* are computationally lighter than *singleGP* and as the number of observations grow, *FuDGE* and *GPoE* are computationally equivalent but *FuDGE* is more accurate. In Fig. 6.6b, it is shown that *FuDGE* and *GPoE* are better off as opposed to their *SingleGP* counterpart as they can efficiently distribute the computational load over the entire fleet. Thus, based on Fig. 6.6, the 3 fusion models can now be arranged in decreasing order of complexity as: $SingleGP > FuDGE \geq GPoE$. From this, it was concluded that not only the *FuDGE* can generate significantly better fused maps as compared to existing state-of-the-art models, but this is also done at equivalent or nominally higher computation costs. Consequently, the *FuDGE* qualifies as a state-of-the-art fusion model best suited for multi-robot teams operating under resource constraints in communication devoid environments.

6.7 Summary

This chapter presents and validates a novel fusion approach wherein multiple locally generated models are fused into a globally consistent model. This approach is scalable with the size of the team and can even handle heterogenous teams wherein the heterogeneity could either be in terms of the nature of the robots or the acquisition function used by each member. Currently, the model has been empirically validated for scenarios where the number of observations to be gathered by each member of the team was pre-set by the user and all members used the same acquisition function for fair comparison to the state-of-the-art fusion models. The *FuDGE* has proven to significantly reduce the computational time and enhance the model accuracy.

Bibliography

[1] K. Tiwari, S. Jeong, and N. Y. Chong, "Point-wise fusion of distributed gaussian process experts (fudge) using a fully decentralized robot team operating in communication-devoid environment," vol. 34, pp. 820–828, IEEE, 2018.

[2] S. Vasudevan, "Data Fusion with Gaussian Processes," *Robotics and Autonomous Systems*, vol. 60, no. 12, pp. 1528–1544, 2012.

[3] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte, "Heteroscedastic Gaussian Processes for Data Fusion in Large Scale Terrain Modeling," in *ICRA*, pp. 3452–3459, 2010.

[4] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte, "Non-Stationary Dependent Gaussian Processes for Data Fusion in Large-Scale Terrain Modeling," in *ICRA*, pp. 1875–1882, 2011.

[5] S. Vasudevan, A. Melkumyan, and S. Scheding, "Efficacy of Data Fusion using Convolved Multi -output Gaussian Processes," *Journal of Data Science*, pp. 341–367, 2015.

[6] J. Sasiadek and P. Hartana, "Sensor data fusion using kalman filter," in *Information Fusion, 2000. FUSION 2000. Proceedings of the Third International Conference on*, vol. 2, pp. WED5–19, IEEE, 2000.

[7] J. Sasiadek, Q. Wang, and M. Zeremba, "Fuzzy adaptive kalman filtering for ins/gps data fusion," in *Intelligent Control, 2000. Proceedings of the 2000 IEEE International Symposium on*, pp. 181–186, IEEE, 2000.

[8] S.-L. Sun and Z.-L. Deng, "Multi-sensor optimal information fusion kalman filter," *Automatica*, vol. 40, no. 6, pp. 1017–1023, 2004.

[9] M. P. Deisenroth and J. W. Ng, "Distributed Gaussian Processes," in *ICML*, vol. 2, p. 5, 2015.

[10] J. Chen, K. H. Low, Y. Yao, and P. Jaillet, "Gaussian Process Decentralized Data Fusion and Active Sensing for Spatiotemporal Traffic Modeling and Prediction in Mobility-on-Demand Systems," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 901–921, 2015.

[11] Y. Cao and D. J. Fleet, "Generalized Product of Experts for Automatic and Principled Fusion of Gaussian Process Predictions," *Modern Nonparametrics 3: Automating the Learning Pipeline workshop at NIPS, Montreal*, 2014.

[12] E. Meeds and S. Osindero, "An Alternative Infinite Mixture of Gaussian Process Experts," *Advances in Neural Information Processing Systems*, vol. 18, p. 883, 2006.

[13] C. Yuan and C. Neubauer, "Variational Mixture of Gaussian Process Experts," in *Advances in Neural Information Processing Systems*, pp. 1897–1904, 2009.

[14] S. E. Yuksel, J. N. Wilson, and P. D. Gader, "Twenty Years of Mixture of Experts," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, pp. 1177–1193, Aug 2012.

[15] V. Tresp, "A Bayesian Committee Machine," *Neural Computation*, vol. 12, no. 11, pp. 2719–2741, 2000.

[16] A. Viseras, T. Wiedemann, C. Manss, and L. Magel, "Decentralized Multi-Agent Exploration with Online-Learning of Gaussian Process," in *(ICRA)*, pp. 4222–4229, May 2016.

[17] F. Xue, R. Subbu, and P. Bonissone, "Locally Weighted Fusion of Multiple Predictive Models," in *IJCNN*, pp. 2137–2143, 2006.

[18] P. Baraldi, A. Cammi, F. Mangili, and E. E. Zio, "Local Fusion of an Ensemble of Models for the Reconstruction of Faulty Signals," *IEEE Transactions on Nuclear Science*, vol. 57, no. 2, pp. 793–806, 2010.

[19] F. Lavancier and P. Rochet, "A General Procedure to Combine Estimators," *Computational Statistics & Data Analysis*, vol. 94, pp. 175–192, 2016.

[20] R. Ouyang, K. H. Low, J. Chen, and P. Jaillet, "Multi-Robot Active Sensing of Non-Stationary Gaussian Process-based Environmental Phenomena," in *AAMAS*, pp. 573–580, 2014.

[21] C. E. Rasmussen and C. K. Williams, *Gaussian processes for machine learning*, vol. 1. MIT press Cambridge, 2006.

[22] N. Waters, "Tobler's First Law of Geography," *The International Encyclopedia of Geography*, 2017.

[23] K. Tiwari, V. Honoré, S. Jeong, N. Y. Chong, and M. P. Deisenroth, "Resource-constrained decentralized active sensing for multi-robot systems using distributed gaussian processes," in *2016 16th International Conference on Control, Automation and Systems (ICCAS)*, pp. 13–18, Oct 2016.

[24] M. Kuczma, *An Introduction to the Theory of Functional Equations and Inequalities: Cauchy's Equation and Jensen's Inequality*. Springer Science & Business Media, 2009.

[25] E. L. Lehmann and J. P. Romano, *Testing statistical hypotheses*. Springer Science & Business Media, 2006.

[26] K. Tiwari, S. Jeong, and N. Y. Chong, "Map-reduce gaussian process (mr-gp) for multi-uav based environment monitoring with limited battery," in *Society of Instrument and Control Engineers of Japan (SICE), 2017 56th Annual Conference of the*, pp. 760–763, IEEE, 2017.

Part IV

Spatiotemporal Modeling: Temporal Evolution of Spatial Variations

Chapter 7

Towards a Spatiotemporal Environment Monitoring for Continuous Domains

> *Time and space are not conditions of existence, time and space is a model of thinking.*
>
> Albert Einstein,

Former chapters have only considered the (discrete) spatial domain for environment monitoring and modeling. In reality however, the environmental dynamics evolve over time and the measurements are continuous signals. To cater to such requirements, now a continuous spatiotemporal domain will be discussed. The reason for this paradigm shift is that when a robot is set out to venture in a real-world scenario, information and measurements are densely packed in the sensing area *i.e.,* information is continuous and could even be available in overwhelming amounts. In order to be able to extend the MR-GP framework to such scenarios, it is necessary to account for continous domains which showcase temporally evolving spatial variations and hence are truly spatiotemporal in nature.

7.1 Continuous Domain Representation

Most of the previous studies for environment monitoring considered static sensor placements [1]. The data acquired by such setups are now being used for validation of GPs being proposed for environment monitoring aided by mobile sensor nodes which give the freedom to cover more ground at higher spatial resolutions. The only problem that remains is that the static sensor placement data is discrete and sparse while the real-world scenarios require continuous and dense maps. For this, motivated by the works of [2], the author harnessed the strength of GP to interpolate the model acquired

from the discrete dataset. For this, the discrete (raw) spatiotemporal dataset was used to train the GP and a very high spatial resolution map was created to replicate the continous spatiotemporal dynamics. This was then taken as the ground truth and will be referred to as *Ground Truth GP (GTGP)*.

7.1.1 Comparison with Discrete domain

In contrast to the previous MR-GP framework presented in Chapter 4 and Chapter 6, the author now transforms the domain from discrete to continuous. The reason for introducing this domain transformation was that in real world, the robot also faces measurements as continuous signals. In doing so, several additional research challenges are introduced:

- The ground truth for a continuous domain is usually not available and hence, approaches need to be developed to solve this problem.
- The point-sensor model is no longer valid. The sensor receives measurements as continuous signal from all locations within the coverage area. Hence, the point-sensor model must now be replaced with a region-sensor model.
- Earlier, measurements were available only at discrete nodes but now, measurements are available all along the trajectory and also the surrounding areas within coverage range. Thus, all measurements needs to be processed. This increases the computational load.
- Owing to increased spatial resolution of measurements, several trajectories need to be evaluated. Each trajectory would lead to acquisition of different amounts of information and only the most informative trajectory would need to be selected.
- Previously, it was assumed that the robot always traverses at a constant velocity and has sufficient computational power to sustain such speeds whilst generating the model. This assumption may be rendered void owing to additional computational load.

This chapter touches upon the first problem mentioned *i.e.,* generating the ground truth for continuous domain using the discrete raw data available.

7.2 Spatiotemporal GP

Spatiotemporal GPs for continous domains can be obtained by considering the regular GP regression which account for time as an additional input dimension [2]. Another possibility would be to convolve a spatial radial basis function (rbf) kernel with a periodic temporal kernel [3]. The latter approach binds the temporal evolution to be periodic while the former is more liberal in that aspect allowing for temporal evolutions to take any form necessary. Thus, for the scope of this work, temporal domain was considered as an additional input dimension. The inputs can now be represented as $\mathbf{x} \in \mathcal{R}^3 \triangleq [\mathbf{s_{x1}}, \mathbf{s_{x2}}, \mathbf{t}]$.

7.2.1 Spatiotemporal Covariance Kernel

In Chapter 3, a squared exponential (rbf) kernel was introduced in Eq. (3.1). Thus far, the author considered one latent length per dimension *i.e., ARD = True* in space and only spatial domain was considered. Most of the environmental phenomenon are known to be isotropic *i.e.,* uniform spread in either spatial direction and so from hereon, the all spatial dimensions are tied together and thus, one latent length is considered for spatial domain and one for the temporal domain. So, now, *ARD = False* since, two hyper-parameters have been tied together. Consider spatiotemporal inputs of the form $\mathbf{x} \in \mathcal{R}^3 \triangleq [\mathbf{s}, \mathbf{t}]$, where, $\mathbf{s} \in \mathcal{R}^2$ and $\mathbf{t} \in \mathcal{R}$. Then, the new spatiotemporal covariance kernel[1] can be represented by:

$$K^{ST}(\mathbf{x}, \mathbf{x}') = \sigma_{sig}^2 \exp\big(-\tfrac{1}{2}\frac{(\mathbf{s}-\mathbf{s}')^T(\mathbf{s}-\mathbf{s}')}{l_s^2} - \tfrac{1}{2}\frac{(\mathbf{t}-\mathbf{t}')^T(\mathbf{t}-\mathbf{t}')}{l_t^2}\big) \tag{7.1}$$

where the spatial length scale is given by l_s and the temporal length scale is given by l_t. Additionally, correlations between noisy observations can be modeled by:

$$K_\epsilon^{ST}(\mathbf{x}, \mathbf{x}') = K^{ST}(\mathbf{x}, \mathbf{x}') + \sigma_n^2 I \tag{7.2}$$

The hyper-parameters are now given by $\boldsymbol{\theta} \triangleq [\sigma_{sig}, l_s, l_t, \sigma_n]$.

7.2.2 Comparison with Spatial GP

As opposed to spatial GPs which were discussed previously in Chapter 3, the inputs now also account for time as an additional dimension. In doing so, the following additional challenges are introduced:

- When considering the temporal variations along side the spatial ones, it is crucial to utilize spatiotemporal kernels that have closed form solutions which can be obtained in polynomial time.
- The memory required to store a spatial kernel is much less than the memory required to store a spatiotemporal kernel.
- The computational cost for inversion of spatial kernel is lower than that of spatiotemporal kernel where inversion is required for inference.
- When considering only the spatial domain, the locations visited in one time-step do not influence those that will be selected at a later time step.

To this end, the author discusses solutions to the foremost challenge mentioned herewith *i.e.,* tractable spatiotemporal kernel inference.

[1]The readers are hereby cautioned that there is a slight abuse of notation here. $\mathbf{x} - \mathbf{x}'$ does not represent subtraction between vectors. Rather it denotes pair-wise distance operation resulting in a matrix where the i, j element represents the distance between $[\mathbf{x}_i, \mathbf{x}'_j]$.

7.3 Experiments

In order to validate and extend the MR-GP architecture to a fully spatiotemporal setting in a continuous domain, this section considers 3 diverse datasets with distinct spatial and temporal variations. For each of these datasets, optimal spatiotemporal kernels were generated by considering time as an additional dimension and interpolation was performed to obtain GTGP in continuous domain. In what follows, the first part summarizes the datasets used for empirical validation and the second part shows the results hence obtained.

7.3.1 Datasets

3 datasets with variable spatial expanse and distinct temporal variations were used for validation of the MR-GP architecture in a spatiotemporal setting. They are summarized below:

- **USA ozone data**[2]: This dataset is provided by the United States Environment Protection Agency (EPA) and is considerably large, spanning across the continental USA with **hourly** samples dating all the way back to 1987. This dataset was previously used in [2]. The samples gathered are static sensor readings for temperature, humidity, solar radiation, ozone concentrations *etc.* For this work, only the ozone concentration was considered since at lower altitudes, this is a detrimental environmental pollutant.

  The raw data provided mentioned above, reported ozone concentration in *[ppb]* gathered from $N = 80$ static sensors. However, after pruning out the stations with missing data, only measurements from $N = 74$ static sensors were retained. These were then used for training the GP model. To obtain the GTGP for ozone dataset, the 24 hrs. raw data from 4th August, 2009 was chosen as training input for a spatiotemporal GP with Squared Exponential covariance from Eq. (7.1). The GTGP for $t = 0.15$ *day* is shown in Fig. 7.1.

- **Ireland wind data**[3]: The second dataset in consideration is the **daily** average wind speed (in knots = 0.5418 m/s) data collected from year $1961 - 1978$ at $N = 12$ meteorological stations across the Republic of Ireland. This dataset was previously used in [4] whereby the authors utilized a non-separable non-stationary spatiotemporal kernel. Similar to the case of ozone dataset, all missing data were first pruned out and then the spatiotemporal GTGP was trained over the data extracted from 24 days. The GTGP for $t = 2$ *day* is shown in Fig. 7.2.

- **Pacific Northwest precipitation data**[4]: The last data set used for analysis consists of **daily** precipitation data collected during the years $1949 - 1994$ in the states of Washington and Oregon. Overall $N = 167$ regions of equal area,

[2]Dataset web access: http://java.epa.gov/castnet/reportPage.do
[3]Dataset web access: http://lib.stat.cmu.edu/datasets/
[4]Dataset web access: http://research.jisao.washington.edu/data_sets/widmann/

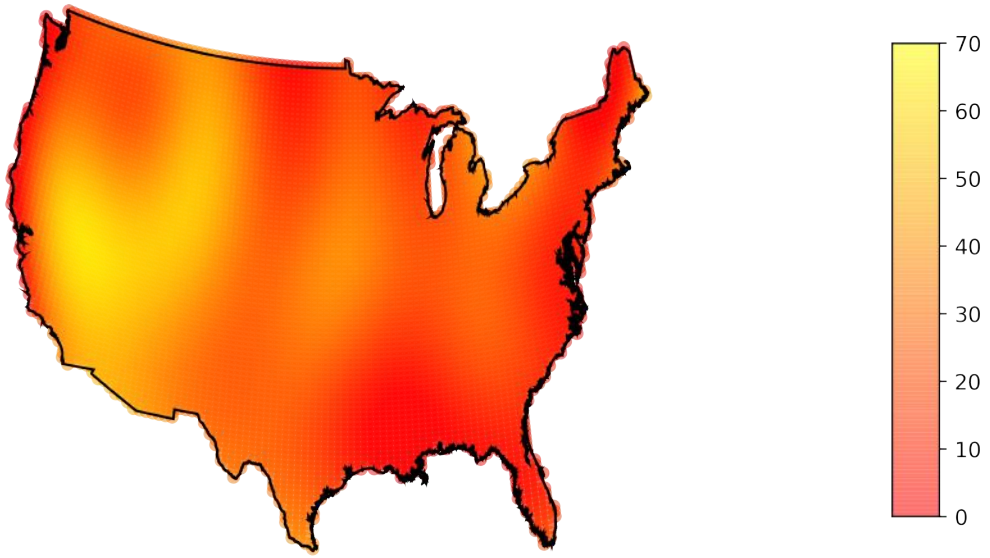


Figure 7.1: GTGP for Ozone. Generating continuous domain map for ozone concentration across continental USA for $t = 0.15$ day.

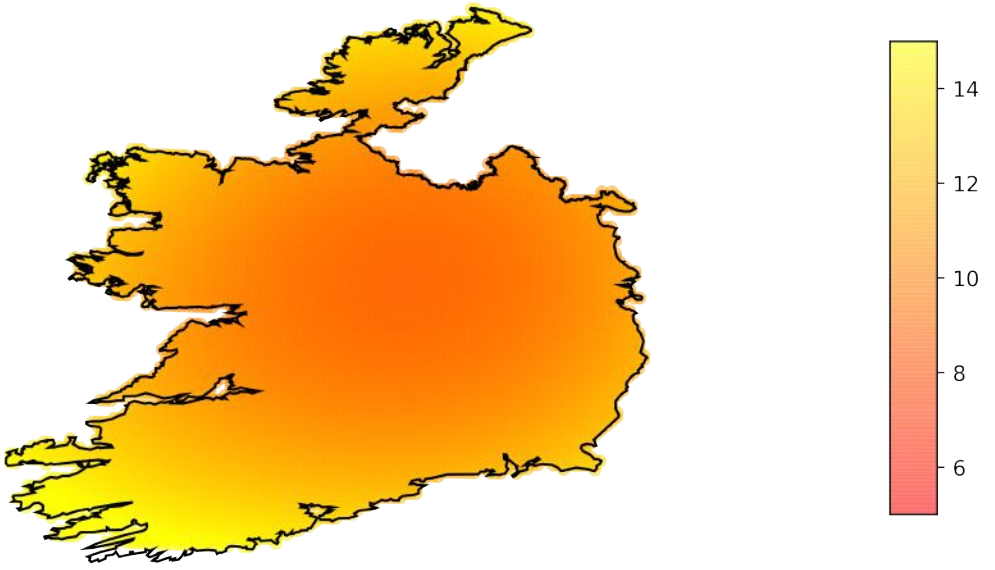


Figure 7.2: GTGP for Wind. Generating continuous domain map for wind speeds across Ireland for $t = 2$ day.

spaced approximately 50 km apart, reported the daily precipitation *[mm]*. To avoid any negative impacts of extreme values on the inference of GTGP, the measurements were transformed in the log-scale and as like earlier, all missing values were filtered out. This dataset was previously used in [1]. Since the dataset was significantly large, to avoid any memory overflow erros, only the first 24 days of data were used to train the GTGP. The results obtained for $t = 2$ *day* is shown in Fig. 7.3.

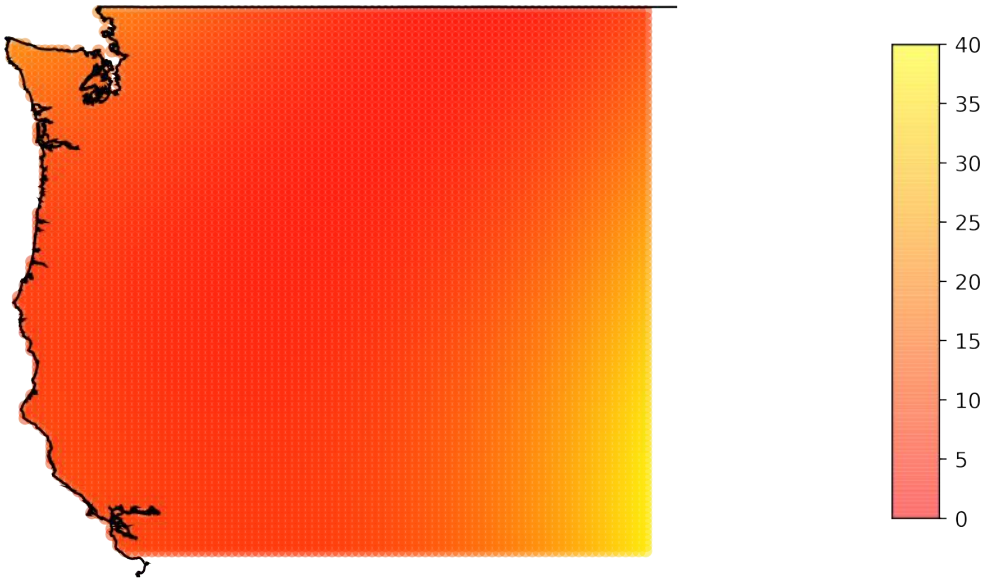


Figure 7.3: GTGP for Precipitation. Generating continuous domain map for precipitation across Pacific Northwest for $t = 2$ day.

7.4 Summary

The first hurdle in generating spatiotemporal models for continuous domains is the lack of the ground truth itself. In this chapter, the author discussed a mechanism to utilize Gaussian Process (GP) interpolation to generate ground truth over continuous domains whilst accounting for both spatial and temporal variations. The previously presented MR-GP architecture is equally applicable in this setting but some additional work needs to be done. For instance, throughout Chapter 4 and Chapter 6, the models were presented for discrete spatial domains. Having a discrete setting, the measurements were only available at the discrete nodes but extending this to a continuous domain means that measurements are now continuous in nature. This presents new challenges in terms of increased computational load for both **Map phase** and **Reduce phase** and will be addressed in further works. This work is meant to serve as a pre-cursor to the extension of MR-GP for continuous spatiotemporal domains but detailed implementations are left for the next phase of development.

Bibliography

[1] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *Journal of Machine Learning Research*, vol. 9, no. Feb, pp. 235–284, 2008.

[2] R. Marchant and F. Ramos, "Bayesian optimisation for intelligent environmental monitoring," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 2242–2249, IEEE, 2012.

[3] R. Marchant and F. Ramos, "Bayesian optimisation for informative continuous path planning," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 6136–6143, IEEE, 2014.

[4] S. Garg, A. Singh, and F. Ramos, "Learning non-stationary space-time models for environmental monitoring.," in *AAAI*, 2012.

[5] C. E. Rasmussen and C. K. Williams, "Gaussian processes in machine learning," *Lecture notes in computer science*, vol. 3176, pp. 63–71, 2004.

[6] C. E. Rasmussen and C. K. Williams, *Gaussian processes for machine learning*, vol. 1. MIT press Cambridge, 2006.

Part V

Epilogue

Chapter 8

Conclusion and Future Works

> *I am not where I need to be, but thank God I am not where I used to be. I am ok and I am on my way !*

Joyce Meyer, *2013*

The aim of this project was to make an attempt to bridge the gap between state-of-the-art machine learning models and cutting edge robots such that machine learning can make the robots fully autonomous but within the limits of current hardware. Making machine learning models work on a high performance hardware is one thing but making the same model work on a real robot hardware is a whole new challenge. With this work, the author intended to propose models that build upon existing state-of-the art machine learning models whilst optimizing them for real robots. The author's vision is to be able to implement GPs on real robot teams and to see them observe and model real world environmental phenomenon in real-time. To this end, several contributions were made as a part of this project to proceed one step closer to realizing this objective.

8.1 Summary of Contributions

This work makes the following contributions:

- Formulation of active sensing as a bi-objective optimization with conflicting objectives to manage resources and simultaneously optimize model quality.
- Dynamic weight deduction for components of bi-objective optimization whilst accounting for residual resources and guaranteeing homing.
- Novel fusion techniques that performs point-wise fusion of predictions from various estimators weighted by their confidence over predictions. Performed as a one-pass procedure by the base station only at the end of mission times of all

robots. Optimized to reduce computation by fusing predictions only over the locations which remain obscure to all models.

- A novel range estimation framework and generalized it to encompass various classes of robots and account for various environmental conditions that a robot may be subjected to during a real mission.
- Extension of MR-GP architecture to account for temporal domain making the architecture suited for spatiotemporal environmental modeling.
- As opposed to other machine learning setup where the training and test dataset are pre-determined, in the spirit of active sensing, the author allows the robots to choose the training and testing sets as deemed necessary to enhance the model accuracy.
- Both the active sensing and range estimation frameworks are meant to work even in the harsh conditions and handle uncertainties in the mission as far as possible.

8.2 Significance of Contributions

The strengths of the contributions made in this work are summarized below:

- It is rather challenging to decide the size of the team required to gather observations from the target phenomenon. To overcome this, the architecture was designed for multi-robot settings that can be easily scaled with the size of the team.
- The architecture can easily accommodate for heterogeneity in the team. This includes different nature of robots involved in a team like UAVs/UGVs *etc.,* along with different active sensing scheme assigned to each agent.
- The MR-GP formulation was designed for robot teams operating in communication devoid environments. This is usually the case in underwater active sensing. Thus, the work is well suited to real-world scenarios and does not bound the robots to be within communication range amongst the team or with the base station.
- The *RC-DAS*† acquisition function was shown to be robust to starting configurations. This is crucial owing to the fact that the environment being monitored is largely unknown and the quality of the model may be affected by the gradient of information followed from the start locations. Independence from starting configuration proves robustness of the architecture.
- The operational range estimation framework can estimate the maximum attainable range with 93% accuracy for the *online* model. This is by far the

state-of-the-art range estimation framework which would prove to be crucial when the MR-GP will be deployed on real robots and will assist *RC-DAS*† to guarantee homing. Not only this, this framework is generic enough to be utilized for any autonomous exploration mission to place upper bounds on the net path lengths that can be incurred by the robot under consideration.

- Erstwhile active sensing schemes have looked into adding resource constraints like [1] but no prior work has tried to solve the homing problem in an information theoretic setting.

8.3 Further Works

Although the proposed framework has been validated extensively in simulations and partially for real world scenarios, there are several ways in which the current architecture can be further enhanced. Some possibilities have been discussed below. Amongst them, some are classified as *necessary* extensions which are essential for the framework to be applied on real robots carrying out missions in real-time as opposed to *sufficient* extensions which are simply suggested to make the framework self-sustaining without relying on external sensor information and/or human supervision. In either of these categories, the extensions are further sub-divided into *Map* and *Reduce* phases respectively.

8.3.1 Necessary Extensions

- **Map Phase:**
  - *Reducing Memory Footprint:* Despite adding resource constraints on active sensing, the memory cost grows as more training data is gathered. This could be solved by truncating the observations like the works of [2] wherein only a set of highly correlated training data is retained. Owing to nautre of the spatiotemporal covariance kernel used in Eq. (7.1), the correlation decays as the temporal separation increases. Thus, not all history is important. But, given that the team was fully disconnected and decentralized, this would also lead to loosing the correlations between obervations across the agents and must be handled appropriately.
  - *Location selection over receeding horizon:* Currently for the *RC-DAS* and *RC-DAS*† acquisition functions, the termination of the exploration is executed if the *next-best-locations* cannot guarantee homing. But instead of doing one-shot termination, the *nth-next-best-location* could be allowed to be selected, if that guarantees homing. This could be useful since information never hurts although the *nth-next-best-location* would provide slightly correlated information where the degree of correlation depends on the magnitude of n. So, even though the quality of information would be sub-optimal but it could still prove to be better than no information at all.

- *Delaying hyper-parameter update:* The active sensing schemes discussed here and those used by peers like [1] all utilize point sensing, *i.e.,* measurments are only obtained from the current location of the robot and the *next-best-location* (when attained). In reality however, the sensor covers a certain region around its location and also measurements are acquired while the robot is executing the trajectories. A cumulative list of all such observations should be considered when updating the hyper-parameters of the GPs like the works of [3, 4].
- *Obstacle Avoidance:* Obstacles (dynamic/static) have not been accounted for in the current active sensing architecture. Since the environment is largely unknown, the robot may encounter obstacles in its path as it is executing its trajectory to reach the *next-best-location* but owing to obstructions, it may need to replan its path.
- *Sequential optimization:* As most of the data acquired by the robots comes in sequentially, instead of inverting the whole kernel per iteration to fit the optimal hyper-parameters, sequential optimization can be utilized. As explained in the Appendix A.1 of [14], Cholesky factors can be updated by recyling previously known factors from memory and only adding new entries corresponding to the new observations. The calculations get a bit simpler when the observations to be appended are stacked at the bottom of the observation list, *i.e.,* new entries in Cholesky factors induces addition of new rows and columns towards the end of the matrix. This allows for a computationally efficient inference for streaming data making the procedure amicable to real-robot setups.

- **Reduce Phase:**
  - *Non-stationary Heteroscedastic fusion:* In the fusion model *FuDGE*, locally stationary homoscedastic models were assumed to be sufficient but sometimes environmental dynamics are highly non-stationary and even have heteroscedastic noise. While there are statistical tests available which can be performed a priori by gathering some prior measurements from static sensors to deduce the heteroscedastic nature, non-stationarity is largely the property of the model itself. In order to make the model suited to all regimes of environmental scenarios, it is advisable to fuse the locally stationary models into one globally non-stationary heteroscedastic model. Inference in such a setting would become rather challenging and hence remains open to further investigation.
  - *Active Fusion for outlier detection:* Currently, the *FuDGE* model is a passive fusion approach which relies solely on the quality of the input models being fused. In such a setting, if during exploration, some agent went rogue owing to sensor failures or improper calculations *etc.,* the performance of fusion model could be affected. In order to handle such outliers, it should rather be transformed to an active fusion mechansim

where the quality of local models and that of the global model are mutually affected.

8.3.2 Sufficient Extensions

- **Map Phase:**
  - *Usage of Area Kernels:* In [5], area kernels have be shown to be useful in merging images of high and low resolution to obtain a fine resolution fused image. This is computationally light and respects the limitations of current hardware. However, another possible application of area kernels is in active sensing. In real world, sensors never gather point observations but a collections of such observations (also referred to as extroceptive sensing). Thus, instead of handling point samples, areas/regions can be handled. Some hints on how to do the inference in such a setting are available in [5]. However, the feasibility to do so in real-time is yet to be investigated.
  - *Addressing Localization uncertainty:* The robot almost never knows its position with 100% certainty. Since, GPs rely on location tagged training samples which are referred to as inputs, the uncertainty in inputs themselves must be accounted for besides the uncertainty in measurements (measurement noise) and the model uncertainty that are currently considered in the framework discussed herewith. To this end, some prior work has been done in [6].
  - *GP-SLAM:* Currently, the environmental monitoring problem is posed as a *mapping* problem where the localization information is assumed to be available from external sensors. Instead, the combined mapping and localization problem *i.e.,* SLAM problem can be solved with GP-LVM based Bayes Filter from [7] to endow the team with co-operative localization techniques and present this as a GP-SLAM problem. SLAM or simultaneous localization and mapping [8] as known from the current literature, is used usually in terms of geometric map generation. However, in the scope of this work, GP-SLAM would be in measurement domain perhaps assisted by geometric domain to allow the robot to navigate the environment.
  - *Team co-ordination for efficient exploration:* In [9], the researchers present an interesting approach to bound the robots within their respective sensing areas and also take into account actuation failures and performance of each agent to shrink the respective voronoi cell. From information theoretic perspective, redundancy (sensing overlaps) could prove to be useful since errors in one model could be rectified by another model. However, in future, we would like to investigate such team coordination approaches to see if it leads to better resource management and accurate models.
  - *Non-myopic path planning:* Currently, the information acquisition functions discussed herewith are strictly *myopic i.e.,* only perform one-step look ahead.

Perhaps, using some non-linear optimizers like Adam [10], the active sensing can be transformed into a globally optimal path planning problem.

- *Continuous time path planning:* The *RC-DAS* and its successor *RC-DAS*† are currently restricted only to spatial domain. Locations observed in one time slice are considered independent of the others. However, as suggested by the works of [11], path planning may be extended to consider the set of locations that are informative across multiple time-steps. However, using a team of multiple-robots, this would involve co-ordinating the team to split the key regions amongst the team and hence will be left to future extensions. Upon successful realization, the MR-GP architecture can also be used for *forecasting* as opposed to its current application to *interpolation* as was shown in this work.
- *Anti-aliasing of observations:* Data associations have been assumed to be perfect. In reality, more than one location within the same locality or even different localities can exhibit similar measurements. Owing to such aliasing, signal strength based localization would become a challenge. Additionally, signal strength based location are known to have an accuracy of the order to a few meters [12, 13] which the author believes, can be improved by harnessing co-operative localization across the team.

- **Reduce Phase:**
  - *Computational Enhancements:* As of now, the fusion architecture utilizes nested for-loops for iterating over all probe points and evaluating confidance per expert. This computational cost will increase significantly when considering the continuous spatiotemporal settings and hence, further optimization techniques need to be investigated.

- **Performance Optimization:**
  - *Parallel processing of Inference and Interpolations:* Generating predictions over the entire spatiotemporal domain with a very high spatial resolution is usually quite slow. Currently, at every iteration of active sensing, both MLE and posterior generation are carried out sequentially. However, in order to optimize performance and distribute computational load over multiple threads, the MLE thread should run at a significantly higher frequency as compared to the posterior generation thread. The frequency of operation of these threads is a function of the data being modeled.

Bibliography

[1] R. Marchant and F. Ramos, "Bayesian optimisation for intelligent environmental monitoring," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 2242–2249, IEEE, 2012.

[2] Y. Xu, J. Choi, and S. Oh, "Mobile sensor network navigation using gaussian processes with truncated observations," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1118–1131, 2011.

[3] H.-L. Choi and J. P. How, "Continuous trajectory planning of mobile sensors for informative forecasting," *Automatica*, vol. 46, no. 8, pp. 1266–1275, 2010.

[4] W. Lu, *Autonomous sensor path planning and control for active information gathering*. PhD thesis, Duke University, 2014.

[5] C. E. Vido and F. Ramos, "From grids to continuous occupancy maps through area kernels," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pp. 1043–1048, IEEE, 2016.

[6] S. Choi, M. Jadaliha, J. Choi, and S. Oh, "Distributed gaussian process regression under localization uncertainty," *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 3, p. 031007, 2015.

[7] J. Ko and D. Fox, "Learning gp-bayesfilters via gaussian process latent variable models," *Autonomous Robots*, vol. 30, no. 1, pp. 3–23, 2011.

[8] S. Thrun and J. J. Leonard, "Simultaneous localization and mapping," in *Springer handbook of robotics*, pp. 871–889, Springer, 2008.

[9] A. Pierson, L. C. Figueiredo, L. C. Pimenta, and M. Schwager, "Adapting to sensing and actuation variations in multi-robot coverage," *The International Journal of Robotics Research*, vol. 36, no. 3, pp. 337–354, 2017.

[10] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[11] R. Marchant and F. Ramos, "Bayesian optimisation for informative continuous path planning," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 6136–6143, IEEE, 2014.

[12] K. H. Low, N. Xu, J. Chen, K. K. Lim, and E. B. Özgül, "Generalized online sparse gaussian processes with application to persistent mobile robot localization," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 499–503, Springer, 2014.

[13] N. Xu, K. H. Low, J. Chen, K. K. Lim, and E. B. Ozgul, "Gp-localize: Persistent mobile robot localization using online sparse gaussian process observation model,"

in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.*, pp. 2585–2593, 2014.

[14] M. A. Osborne, S. J. Roberts, A. Rogers, S. D. Ramchurn, and N. R. Jennings, "Towards real-time information processing of sensor network data using computationally efficient multi-output gaussian processes," in *Information Processing in Sensor Networks, 2008. IPSN'08. International Conference on*, pp. 109–120, IEEE, 2008.

Part VI

Addendum

Chapter 9

Appendix

9.1 Inference using stationary GPs with RBF Kernels

Lemma 9.1 (Inverse of Partitioned Matrix)**.** *If A is non-singular $n \times n$ matrix partitioned as $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$. Then the inverse of this partitioned matrix is given by:*

$$A^{-1} = \begin{bmatrix} (A_{11} - A_{12}A_{22}^{-1}A_{21})^{-1} & -(A_{11} - A_{12}A_{22}^{-1}A_{21})^{-1}A_{12}A_{22}^{-1} \\ -(A_{22} - A_{21}A_{11}^{-1}A_{12})^{-1}A_{21}A_{11}^{-1} & (A_{22} - A_{21}A_{11}^{-1}A_{12})^{-1} \end{bmatrix} \tag{9.1}$$

Proof. For proof sketch, please refer to [1]. □

Theorem 9.2 (Posterior over Exponential Kernels)**.** *Given a column vector $\mathbf{z}_O$ of observed measurements for some set $O \subset D$ of the domain D, a GP model can predict the measurements for any set $U \subset D \setminus O$ using the predictive distribution $p[\mathbf{y}^* \in U|\mathbf{y}] \sim \mathcal{N}(\boldsymbol{\mu}_{U|O}, \Sigma_{UU|O})$. Here,*

$$\begin{aligned} \boldsymbol{\mu}_{U|O} &= \mu_U + \Sigma_{UO}\Sigma_{OO}^{-1}(\mathbf{z}_O - \boldsymbol{\mu}_O) \\ \Sigma_{UU|O} &= \Sigma_{UU} - \Sigma_{UO}\Sigma_{OO}^{-1}\Sigma_{OU} \end{aligned} \tag{9.2}$$

Proof. Our noisy observations $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ for $\mathbf{x} \in \mathbb{R}^D$ and $y \in \mathbb{R}^1$ can be represented using some latent function f as:

$$y_i = f(\mathbf{x}_i) + \epsilon_i \tag{9.3}$$

where $f \sim GP(\mu(\cdot), k_f(\cdot,\cdot))$[1] and $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. Consider a set of observed inputs $\mathbf{x} \in O$ and unobserved inputs $\mathbf{x}^* \in U$. Since, the sum of independent Gaussian random

[1]Here k_f represents the kernel such that $\mathbf{cov}(f(x), f(x')) = k_f(x, x')$

variables is also Gaussian, we have:

$$\begin{aligned}\begin{bmatrix}\mathbf{y}\\ \mathbf{y}^*\end{bmatrix} &= \begin{bmatrix}f(\mathbf{x})\\ f(\mathbf{x}^*)\end{bmatrix} + \begin{bmatrix}\boldsymbol{\epsilon}\\ \boldsymbol{\epsilon}^*\end{bmatrix}\\ &= \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_O\\ \boldsymbol{\mu}_U\end{bmatrix}, \begin{bmatrix}\Sigma_{OO} & \Sigma_{OU}\\ \Sigma_{UO} & \Sigma_{UU}\end{bmatrix}\right) + \mathcal{N}\left(\begin{bmatrix}\vec{0}\\ \vec{0}\end{bmatrix}, \begin{bmatrix}\sigma^2 I & \vec{0}\\ \vec{0} & \sigma^2 I\end{bmatrix}\right)\\ &= \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_O\\ \boldsymbol{\mu}_U\end{bmatrix}, \begin{bmatrix}\Sigma_{OO}+\sigma^2 I & \Sigma_{OU}\\ \Sigma_{UO} & \Sigma_{UU}+\sigma^2 I\end{bmatrix}\right)\end{aligned} \tag{9.4}$$

Let $A = \begin{bmatrix}\Sigma_{OO}+\sigma^2 I & \Sigma_{OU}\\ \Sigma_{UO} & \Sigma_{UU}+\sigma^2 I\end{bmatrix}$ represent the partitioned matrix used above and $V = A^{-1}$ represent the inverse of such matrix. Then from Lemma 9.1, we can obtain the inverse of this partitioned matrix such that

$$\begin{aligned}V = A^{-1} &= \begin{bmatrix}V_{OO} & V_{OU}\\ V_{UO} & V_{UU}\end{bmatrix}\\ &= \begin{bmatrix}(\Sigma_{OO}-\Sigma_{OU}\Sigma_{UU}^{-1}\Sigma_{UO})^{-1} & -(\Sigma_{OO}-\Sigma_{OU}\Sigma_{UU}^{-1}\Sigma_{UO})^{-1}\Sigma_{OU}\Sigma_{UU}^{-1}\\ -(\Sigma_{UU}-\Sigma_{UO}\Sigma_{OO}^{-1}\Sigma_{OU})^{-1}\Sigma_{UO}\Sigma_{OO}^{-1} & (\Sigma_{UU}-\Sigma_{UO}\Sigma_{OO}^{-1}\Sigma_{OU})^{-1}\end{bmatrix}\end{aligned} \tag{9.5}$$

So, the posterior on $\mathbf{y}^*$ is now given by the conditional probability $p[\mathbf{y}^* \in U|\mathbf{y}]$ which can be expanded as:

$$\begin{aligned}&p[\mathbf{y}^* \in U|\mathbf{y}] =\\ &\frac{1}{\zeta_1}\cdot\left[\exp\left\{-\frac{1}{2}\left(\begin{bmatrix}\mathbf{Z}_U\\ \mathbf{z}_O\end{bmatrix}-\begin{bmatrix}\boldsymbol{\mu}_U\\ \boldsymbol{\mu}_O\end{bmatrix}\right)^T\begin{bmatrix}V_{OO} & V_{OU}\\ V_{UO} & V_{UU}\end{bmatrix}\left(\begin{bmatrix}\mathbf{Z}_U\\ \mathbf{z}_O\end{bmatrix}-\begin{bmatrix}\boldsymbol{\mu}_U\\ \boldsymbol{\mu}_O\end{bmatrix}\right)\right\}\right]\end{aligned} \tag{9.6}$$

where, ζ_1 is the normalization constant independent of $\mathbf{Z}_U$.

We will now expand the expression within the $\exp(\cdot)$ in an attempt to simplify it. Thus far, we have:

$$\begin{aligned}&\left(\begin{bmatrix}\mathbf{Z}_U\\ \mathbf{z}_\mathbf{x}\end{bmatrix}-\begin{bmatrix}\boldsymbol{\mu}_U\\ \boldsymbol{\mu}_O\end{bmatrix}\right)^T\begin{bmatrix}V_{OO} & V_{OU}\\ V_{UO} & V_{UU}\end{bmatrix}\left(\begin{bmatrix}\mathbf{Z}_U\\ \mathbf{z}_O\end{bmatrix}-\begin{bmatrix}\boldsymbol{\mu}_U\\ \boldsymbol{\mu}_O\end{bmatrix}\right)\\ =&(\mathbf{Z}_U-\boldsymbol{\mu}_U)^T V_{UU}(\mathbf{Z}_U-\boldsymbol{\mu}_U)+(\mathbf{Z}_U-\boldsymbol{\mu}_U)^T V_{UO}(\mathbf{z}_O-\boldsymbol{\mu}_O)+\\ &(\mathbf{z}_O-\boldsymbol{\mu}_O)^T V_{OU}(\mathbf{Z}_U-\boldsymbol{\mu}_U)+(\mathbf{z}_O-\boldsymbol{\mu}_O)^T V_{OO}(\mathbf{z}_O-\boldsymbol{\mu}_O)^T\end{aligned} \tag{9.7}$$

We will retain only the terms dependent on $\mathbf{Z}_U$ and use the fact that $V_{UO} = V_{OU}^T$ to get:

$$p[f(\mathbf{x}^*)|y(\mathbf{x})] = \frac{1}{\zeta_2}\exp\left(-\frac{1}{2}\left[\mathbf{Z}_U^T V_{UU}\mathbf{Z}_U - 2\mathbf{Z}_U^T V_{UU}\boldsymbol{\mu}_U + 2\mathbf{Z}_U^T V_{UO}(\mathbf{z}_O-\boldsymbol{\mu}_O)\right]\right) \tag{9.8}$$

By completing the square, we can further simplify the expression to:

$$p[f(\mathbf{x}^*)|y(\mathbf{x})] = \frac{1}{\zeta_3}\exp\left(-\frac{1}{2}(\mathbf{Z}_U - \boldsymbol{\mu}_{U|O})^T V_{UU}(\mathbf{Z}_U - \boldsymbol{\mu}_{U|O})\right) \tag{9.9}$$

where $\boldsymbol{\mu}_{U|O} = \boldsymbol{\mu}_U - V_{UU}^{-1}V_{UO}(\mathbf{z}_O - \boldsymbol{\mu}_O)$.

Thus ,

$$p[f(\mathbf{x}^*)|y(\mathbf{x})] \sim \mathcal{N}(\boldsymbol{\mu}_{U|O}, V_{UU}^{-1}) \tag{9.10}$$

Now, plugging Eq. (9.5) above, we get:

$$\begin{aligned}\boldsymbol{\mu}_{U|O} &= \boldsymbol{\mu}_U - V_{UU}^{-1}V_{UO}(\mathbf{z}_O - \boldsymbol{\mu}_O) \\ &= \boldsymbol{\mu}_U + (\Sigma_{UU} - \Sigma_{UO}\Sigma_{OO}^{-1}\Sigma_{OU})((\Sigma_{UU} - \Sigma_{UO}\Sigma_{OO}^{-1}\Sigma_{OU})^{-1}\Sigma_{UO}\Sigma_{OO}^{-1})(\mathbf{z}_O - \boldsymbol{\mu}_O) \\ &= \mu_U + \Sigma_{UO}\Sigma_{OO}^{-1}(\mathbf{z}_O - \boldsymbol{\mu}_O)\end{aligned} \tag{9.11}$$

and

$$V_{UU}^{-1} = \Sigma_{UU} - \Sigma_{UO}\Sigma_{OO}^{-1}\Sigma_{OU} \tag{9.12}$$

□

Lemma 9.3 (MLE using RBF Kernel)**.** *The likelihood of seeing a noisy observation $y = f(x) + \epsilon$ is defined as $p(\mathbf{y}|X, \theta)$ where $X \in O$ represents the observed inputs and θ represents the hyper-parameters of the rbf kernel defined by Eq.* (3.1)*. Then, the log likelihood is given by:*

$$\mathcal{L} = \log(p(\mathbf{y}|X, \boldsymbol{\theta})) = \underbrace{-\frac{1}{2}\log|K|}_{\text{model complexity}}\ \underbrace{-\frac{1}{2}\mathbf{y}^T K^{-1}\mathbf{y}}_{\text{model fit}}\underbrace{-\frac{d}{2}\log(2\pi)}_{\text{normalizer}} \tag{9.13}$$

The optimal hyper-parameters for the rbf kernel are those which maximize the marginal log-likelihood given in Eq. (9.3)*. Thus, we define the partial derivatives of $\mathcal{L}$ with respect to the hyper-parameters $\boldsymbol{\theta}$ as:*

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\theta_i} &= -\frac{1}{2}\frac{\partial\log|K|}{\partial\theta_i} - \frac{1}{2}\frac{\partial\mathbf{y}^T K^{-1}\mathbf{y}}{\partial\theta_i} - \frac{d}{2}\frac{\partial\log(2\pi)}{\partial\theta_i} \\ &= -\frac{1}{2}\operatorname{tr}\left(K^{-1}\frac{\partial K}{\partial\theta_i}\right) - \frac{1}{2}\left(\mathbf{y}^T K^{-1}\frac{\partial K}{\partial\theta_i}K^{-1}\mathbf{y}\right) \\ &= -\frac{1}{2}\operatorname{tr}\left(K^{-1}\frac{\partial K}{\partial\theta_i}\right) - \frac{1}{2}\operatorname{tr}\left(K^{-1}\mathbf{y}\mathbf{y}^T K^{-1}\frac{\partial K}{\partial\theta_i}\right) \\ &= \frac{1}{2}\operatorname{tr}\left((K^{-1}\mathbf{y}\mathbf{y}^T K^{-1} - K^{-1})\frac{\partial K}{\partial\theta_i}\right)\end{aligned} \tag{9.14}$$

Lemma 9.4 (Derivatives of RBF Kernel w.r.t. $\boldsymbol{\theta} \triangleq [\sigma_s, l_s, l_t, \sigma_n]$)**.** *The derivatives of the RBF kernel with respect to its parameters are given by:*

$$\frac{\partial K}{\partial \sigma_s^2} = \exp\left(\frac{-||\mathbf{x}-\mathbf{x}'||}{2l^2}\right) \tag{9.15}$$

$$\begin{aligned}\frac{\partial K}{\partial l_s} &= \left(\frac{-||\mathbf{x}-\mathbf{x}'||^2}{l_s^3}\right)\frac{\partial K}{\partial \sigma_s^2} \\ \frac{\partial K}{\partial l_t} &= \left(\frac{-||\mathbf{x}-\mathbf{x}'||^2}{l_t^3}\right)\frac{\partial K}{\partial \sigma_s^2}\end{aligned} \tag{9.16}$$

$$\frac{\partial K}{\partial \sigma_n^2} = I \tag{9.17}$$

9.2 Entropy of GP

Lemma 9.5 (Symmetry of trace of product of 2 matrices)**.** *Suppose $P \in \mathcal{R}^{m\times n}$ and $Q \in \mathcal{R}^{n\times m}$, then* $\mathrm{tr}(PQ) = \mathrm{tr}(QP)$.

Proof. By the definition of $\mathrm{tr}(\cdot)$, we know that
$tr(PQ) = \Sigma_{i=1}^{m}(PQ)_{ii} = \Sigma_{i=1}^{m}\Sigma_{j=1}^{n}P_{ij}Q_{ji} = \Sigma_{i=1}^{n}\Sigma_{j=1}^{m}Q_{ji}P_{ij} = \Sigma_{j=1}^{m}(QP)_{jj} = \mathrm{tr}(QP)$. □

Corollary 9.6 (Symmetry of trace for 3 matrices)**.** *Suppose $P \in \mathcal{R}^{m\times n}$, $Q \in \mathcal{R}^{n\times o}$ and $R \in \mathcal{R}^{o\times m}$ then* $\mathrm{tr}(PQR) = \mathrm{tr}(RQP)$.

Proof. Let $S = PQ$ and $T = R$. Then from Lemma (9.5), we already know that:

$$\begin{aligned}&\mathrm{tr}(ST) = \mathrm{tr}(TS) \\ \implies &\mathrm{tr}(PQR) = \mathrm{tr}(RQP)\end{aligned} \tag{9.18}$$

□

Theorem 9.7 (Entropy of GP)**.** *Let $\Sigma_{UU|O}$ represent the posterior covariance of a GP for set $O \subset D$ representing the observed inputs and set $U \subset D$ standing for unobserved inputs. Let Z_{**} represent the random measurements for $**$ sampled from either set O or U. Then, the conditional entropy $\mathbb{H}[\mathbf{Z}_U|\mathbf{Z}_D]$ is denoted by:*

$$\mathbb{H}[\mathbf{Z}_U|\mathbf{Z}_D] = \frac{1}{2}\log[(2\pi e)^{\#(U)}|\Sigma_{UU|O}|] \tag{9.19}$$

Proof. Consider a column vector of random measurements $\mathbf{Z}_U$ for inputs belonging to set U. We know that $\mathbf{Z}_U \sim \mathcal{N}(\boldsymbol{\mu}_{U|O}, \Sigma_{UU|O})$ with a pdf given by:

$$\psi(\mathbf{Z}_U) = \frac{1}{\sqrt{(2\pi)^{\#(U)}|\Sigma_{UU|O}|}}\exp\left(-\frac{1}{2}(\mathbf{Z}_U - \boldsymbol{\mu}_{U|O})^T{\Sigma_{UU|O}}^{-1}(\mathbf{Z}_U - \boldsymbol{\mu}_{U|O})\right) \tag{9.20}$$

Then, by the definition of Shannon entropy over the continuous domain, we have

$$
\begin{aligned}
&\mathbb{H}[\mathbf{Z}_U|\mathbf{Z}_D] \\
&= -\int \{\psi(\mathbf{Z}_U)\log(\psi(\mathbf{Z}_U))\}\, d\mathbf{Z}_U \\
&= -\int \left\{\psi(\mathbf{Z}_U)\left[-\frac{1}{2}(\mathbf{Z}_U-\boldsymbol{\mu}_{U|O})^T\Sigma_{UU|O}^{-1}(\mathbf{Z}_U-\boldsymbol{\mu}_{U|O}) - \log(\sqrt{(2\pi)^{\#(U)}|\Sigma_{UU|O}|})\right]\right\} d\mathbf{Z}_U \\
&= \frac{1}{2}\,\mathbb{E}_\psi[\mathrm{tr}\left\{(\mathbf{Z}_U-\boldsymbol{\mu}_{U|O})^T\Sigma_{UU|O}^{-1}(\mathbf{Z}_U-\boldsymbol{\mu}_{U|O})\right\}] + \frac{1}{2}\log\left[(2\pi)^{\#(U)}|\Sigma_{UU|O}|\right]
\end{aligned}
$$

using Corollary (9.6), we get:

$$
\begin{aligned}
&= \frac{1}{2}\,\mathbb{E}_\psi[\mathrm{tr}\left\{(\mathbf{Z}_U-\boldsymbol{\mu}_{U|O})(\mathbf{Z}_U-\boldsymbol{\mu}_{U|O})^T\Sigma_{UU|O}^{-1}\right\}] + \frac{1}{2}\log\left[(2\pi)^{\#(U)}|\Sigma_{UU|O}|\right] \\
&= \frac{1}{2}\,\mathbb{E}_\psi[\mathrm{tr}\left\{(\mathbf{Z}_U-\boldsymbol{\mu}_{U|O})(\mathbf{Z}_U-\boldsymbol{\mu}_{U|O})^T\right\}]\Sigma_{UU|O}^{-1} + \frac{1}{2}\log\left[(2\pi)^{\#(U)}|\Sigma_{UU|O}|\right] \\
&= \frac{1}{2}\,\mathrm{tr}\left\{\Sigma_{UU|O}\Sigma_{UU|O}^{-1}\right\} + \frac{1}{2}\log\left[(2\pi)^{\#(U)}|\Sigma_{UU|O}|\right] \\
&= \frac{1}{2}\,\mathrm{tr}\left\{\mathcal{I}\right\} + \frac{1}{2}\log\left[(2\pi)^{\#(U)}|\Sigma_{UU|O}|\right] \\
&= \frac{1}{2}\#(U) + \frac{1}{2}\log\left[(2\pi)^{\#(U)}|\Sigma_{UU|O}|\right] \\
&= \frac{1}{2}\log\left[(2\pi e)^{\#(U)}|\Sigma_{UU|O}|\right]
\end{aligned}
\tag{9.21}
$$

□

9.3 Rusti V2.0

9.3.1 Circuit Diagram for Rusti V2.0

The circuit diagram used for the assembling the electronic components of Rusti V2.0 is shown in Fig. 9.1.

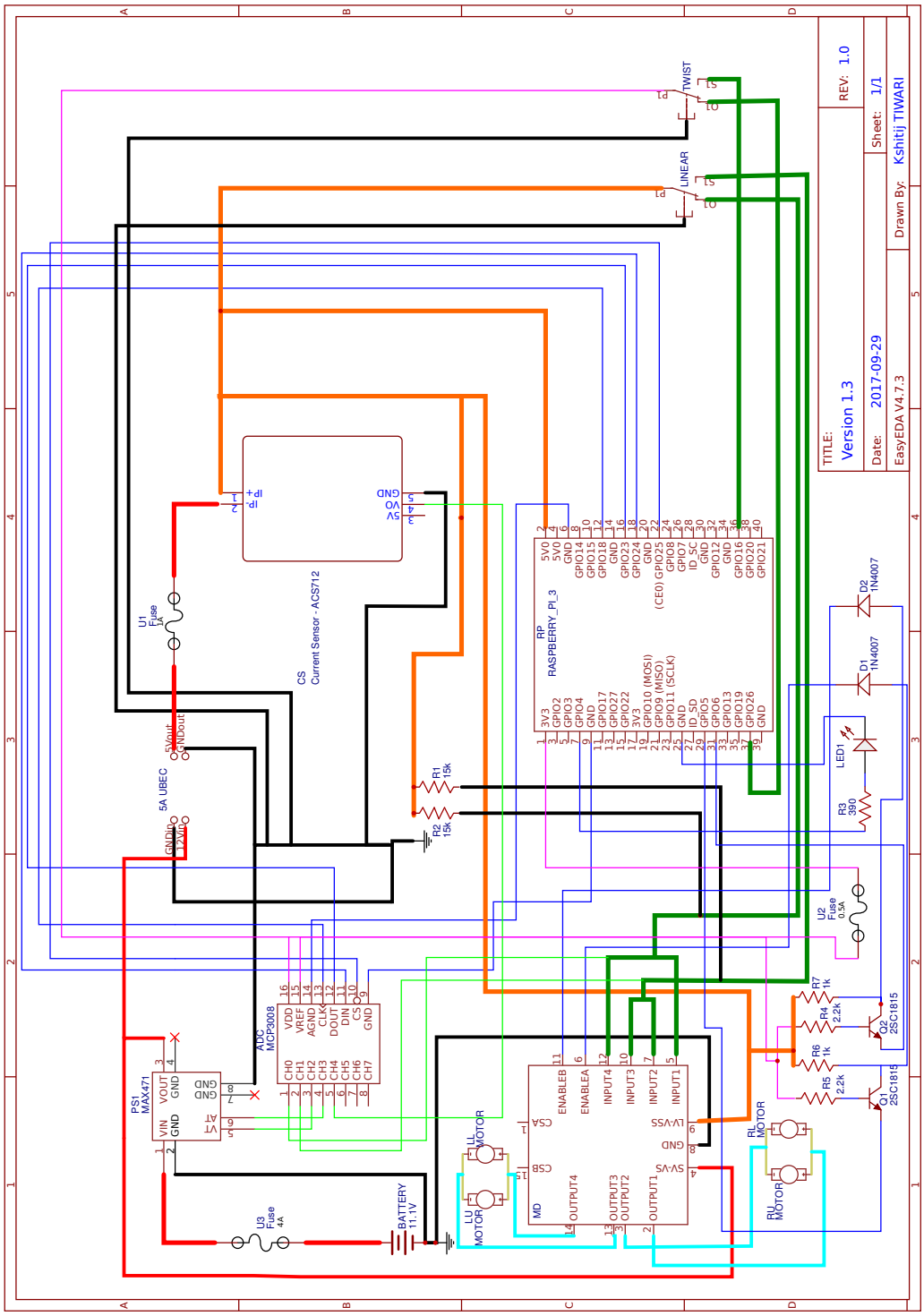


Figure 9.1: Circuit Diagram for Rusti V2.0

9.3.2 Level Shifter for Rusti V2.0

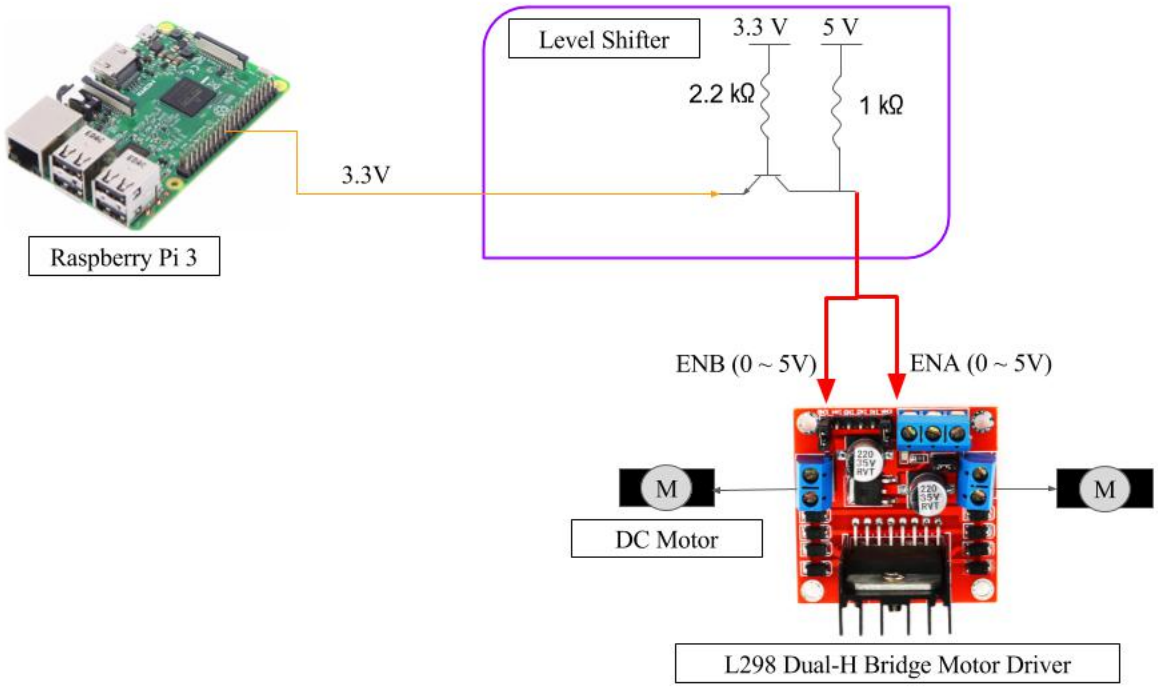


Figure 9.2: Level shifter

Bibliography

[1] H. J. Bierens, "The inverse of a partitioned matrix." `http://www.math.chalmers.se/~rootzen/highdimensional/blockmatrixinverse.pdf`.

[2] D. Duvenaud, J. Lloyd, R. Grosse, J. Tenenbaum, and G. Zoubin, "Structure discovery in nonparametric regression through compositional kernel search," in *Proceedings of the 30th International Conference on Machine Learning* (S. Dasgupta and D. McAllester, eds.), vol. 28 of *Proceedings of Machine Learning Research*, (Atlanta, Georgia, USA), pp. 1166–1174, PMLR, 17–19 Jun 2013.

References

[1] K. Deb, "Multi-objective optimization," in *Search methodologies*, pp. 403–449, Springer, 2014.

[2] C. Pirri, M. Cocuzza, L. Scaltrito, S. Ferrero, P. Rivolo, *et al.*, "Evolved sensing platform for the offshore sites sea water environmental monitoring," in *Offshore Mediterranean Conference and Exhibition*, Offshore Mediterranean Conference, 2017.

[3] M. Tavakol, R. Arjmandi, M. Shayeghi, S. M. Monavari, and A. Karbassi, "Developing an environmental water quality monitoring program for haraz river in northern iran," *Environmental Monitoring and Assessment*, vol. 189, no. 8, p. 410, 2017.

[4] F. Weiss, H. Wey, C. Stamm, C. Ruepert, C. Zurbrügg, and R. Eggen, "Passive sampling approaches used for time-integrated environmental monitoring and risk assessment in the tropical río tapezco catchment in costa rica," in *EGU General Assembly Conference Abstracts*, vol. 19, p. 13660, 2017.

[5] J. T. James, T. F. Limero, S. W. Beck, M. Martin, P. A. Covington, L. Yang, D. Lind, and J. F. Boyd, "Environmental monitoring air quality," *Helen W. Lane*, p. 177, 2002.

[6] N. Domínguez-Morueco, S. Augusto, L. Trabalón, E. Pocurull, F. Borrull, M. Schuhmacher, J. L. Domingo, and M. Nadal, "Monitoring pahs in the petrochemical area of tarragona county, spain: comparing passive air samplers with lichen transplants," *Environmental Science and Pollution Research*, vol. 24, no. 13, pp. 11890–11900, 2017.

[7] D. Mage, G. Ozolins, P. Peterson, A. Webster, R. Orthofer, V. Vandeweerd, and M. Gwynne, "Urban air pollution in megacities of the world," *Atmospheric Environment*, vol. 30, no. 5, pp. 681–686, 1996.

[8] I. Krupnik and D. Jolly, *The Earth Is Faster Now: Indigenous Observations of Arctic Environmental Change. Frontiers in Polar Social Science*. ERIC, 2002.

[9] I. P. on Climate Change, *Climate Change 2014–Impacts, Adaptation and Vulnerability: Regional Aspects*. Cambridge University Press, 2014.

[10] G. T. Pecl, M. B. Araújo, J. D. Bell, J. Blanchard, T. C. Bonebrake, I.-C. Chen, T. D. Clark, R. K. Colwell, F. Danielsen, B. Evengård, *et al.*, "Biodiversity redistribution under climate change: Impacts on ecosystems and human well-being," *Science*, vol. 355, no. 6332, p. eaai9214, 2017.

[11] I. Dincer, "Renewable energy and sustainable development: a crucial review," *Renewable and Sustainable Energy Reviews*, vol. 4, no. 2, pp. 157–175, 2000.

[12] H. K. White, P.-Y. Hsing, W. Cho, T. M. Shank, E. E. Cordes, A. M. Quattrini, R. K. Nelson, R. Camilli, A. W. Demopoulos, C. R. German, *et al.*, "Impact of the deepwater horizon oil spill on a deep-water coral community in the gulf of mexico," *Proceedings of the National Academy of Sciences*, vol. 109, no. 50, pp. 20303–20308, 2012.

[13] B. Bayat, N. Crasta, A. Crespi, A. M. Pascoal, and A. Ijspeert, "Environmental monitoring using autonomous vehicles: a survey of recent searching techniques," *Current Opinion in Biotechnology*, vol. 45, pp. 76–84, 2017.

[14] M. Egerstedt, "Persistent environmental monitoring: Robots that seemingly do nothing most of the time," 2017.

[15] M. Dunbabin and L. Marques, "Robots for environmental monitoring: Significant advancements and applications," *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 24–39, 2012.

[16] H. Messer, A. Zinevich, and P. Alpert, "Environmental monitoring by wireless communication networks," *Science*, vol. 312, no. 5774, pp. 713–713, 2006.

[17] M. Trincavelli, M. Reggente, S. Coradeschi, A. Loutfi, H. Ishida, and A. J. Lilienthal, "Towards environmental monitoring with mobile robots," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pp. 2210–2215, IEEE, 2008.

[18] L. Arnold, "Stochastic differential equations," *New York*, 1974.

[19] M. Vidyasagar, *Nonlinear systems analysis*. SIAM, 2002.

[20] J. Binney, A. Krause, and G. S. Sukhatme, "Informative path planning for an autonomous underwater vehicle," in *Robotics and automation (icra), 2010 IEEE international conference on*, pp. 4791–4796, IEEE, 2010.

[21] R. Murphy, *Introduction to AI robotics*. MIT press, 2000.

[22] M. S. Grewal, "Kalman filtering," in *International Encyclopedia of Statistical Science*, pp. 705–708, Springer, 2011.

[23] J. Ko and D. Fox, "Gp-bayesfilters: Bayesian filtering using gaussian process prediction and observation models," *Autonomous Robots*, vol. 27, no. 1, pp. 75–90, 2009.

[24] A. Antoniou, *Digital signal processing*. McGraw-Hill, 2016.

[25] N. Seltenrich, "Keeping tabs on habs: new tools for detecting, monitoring, and preventing harmful algal blooms," *Environmental health perspectives*, vol. 122, no. 8, p. A206, 2014.

[26] S. McNEIL, "Growing algae bloom in Arabian Sea tied to climate change." https://tinyurl.com/l3et3oy, 2017. [Online; accessed 6-Dec-2017].

[27] S. Koperski, "Experimental drone used for controlled burn." https://tinyurl.com/y72bulxw, 2016. [Online; accessed 6-Dec-2017].

[28] S. D. of Agriculture. https://tinyurl.com/y922wnel. [Online; accessed 6-Dec-2017].

[29] N. THABIT and J. BERTUCCI, "AWAKENING FROM THE DREAM: THE REAL FACTS ON FUKUSHIMA." https://tinyurl.com/y7qkr4f4, 2012. [Online; accessed 6-Dec-2017].

[30] I. R. I. for Nuclear Decommissioning (IRID), "Robots working inside the buildings at Fukushima Daiichi NPS (Part II) MHI-MEISTeR - Images taken inside the Unit 2 Reactor building by the robot -AWAKENING FROM THE DREAM: THE REAL FACTS ON FUKUSHIMA." http://irid.or.jp/en/research/meister/. [Online; accessed 6-Dec-2017].

[31] N. Oceanic and A. A. (NOAA), "New Seaglider Collects Data along Gulf Coast." https://tinyurl.com/y7s9dt98, 2012. [Online; accessed 6-Dec-2017].

[32] S. Arismendez, "For Submerged Oil Pollution in Western Gulf of Mexico, Restoration Is Coming After 2005 DBL 152 Oil Spill." https://tinyurl.com/y8jzuto7, 2013. [Online; accessed 6-Dec-2017].

[33] Z. Ma, K. Yin, L. Liu, and G. S. Sukhatme, "A spatio-temporal representation for the orienteering problem with time-varying profits," *arXiv preprint arXiv:1611.08037*, 2016.

[34] J. Yu, M. Schwager, and D. Rus, "Correlated orienteering problem and its application to persistent monitoring tasks," *IEEE Transactions on Robotics*, vol. 32, pp. 1106–1118, Oct 2016.

[35] "Pokémon Go and Traveling Salesman Problem." http://www.math.uwaterloo.ca/tsp/poke/index.html, 2016.

[36] R. S. Olson, "Computing the optimal road trip across the U.S.." http://www.randalolson.com/2015/03/08/computing-the-optimal-road-trip-across-the-u-s/, 2017.

[37] R. S. Garfinkel, G. L. Nemhauser, *et al.*, *Integer programming*, vol. 4. Wiley New York, 1972.

[38] A. Gunawan, H. C. Lau, and P. Vansteenwegen, "Orienteering problem: A survey of recent variants, solution approaches and applications," *European Journal of Operational Research*, vol. 255, no. 2, pp. 315–332, 2016.

[39] P. Toth and D. Vigo, *The vehicle routing problem*. SIAM, 2002.

[40] I.-M. Chao, B. L. Golden, and E. A. Wasil, "A fast and effective heuristic for the orienteering problem," *European journal of operational research*, vol. 88, no. 3, pp. 475–489, 1996.

[41] L. Ke, C. Archetti, and Z. Feng, "Ants can solve the team orienteering problem," *Computers & Industrial Engineering*, vol. 54, no. 3, pp. 648–665, 2008.

[42] C. Chekuri and S. Khanna, "A polynomial time approximation scheme for the multiple knapsack problem," *SIAM Journal on Computing*, vol. 35, no. 3, pp. 713–728, 2005.

[43] A. A. Bertossi, "The edge hamiltonian path problem is np-complete," *Information Processing Letters*, vol. 13, no. 4-5, pp. 157–159, 1981.

[44] T. Bektas, "The multiple traveling salesman problem: an overview of formulations and solution procedures," *Omega*, vol. 34, no. 3, pp. 209–219, 2006.

[45] S. Trigui, O. Cheikhrouhou, A. Koubaa, U. Baroudi, and H. Youssef, "Fl-mtsp: a fuzzy logic approach to solve the multi-objective multiple traveling salesman problem for multi-robot systems," *Soft Computing*, vol. 21, no. 24, pp. 7351–7362, 2017.

[46] S. Ozden, A. E. Smith, and K. R. Gue, "Solving large batches of traveling salesman problems with parallel and distributed computing," *Computers & Operations Research*, vol. 85, pp. 87–96, 2017.

[47] D. Golovin and A. Krause, "Adaptive submodularity: Theory and applications in active learning and stochastic optimization," *Journal of Artificial Intelligence Research*, vol. 42, pp. 427–486, 2011.

[48] A. Krause and C. Guestrin, "Near-optimal observation selection using submodular functions," in *AAAI*, vol. 7, pp. 1650–1654, 2007.

[49] A. Singh, A. Krause, and W. J. Kaiser, "Nonmyopic adaptive informative path planning for multiple robots.," in *IJCAI*, vol. 3, p. 2, 2009.

[50] A. Singh, A. Krause, C. Guestrin, W. J. Kaiser, and M. A. Batalin, "Efficient planning of informative paths for multiple robots.," in *IJCAI*, vol. 7, pp. 2204–2211, 2007.

[51] R. Marchant and F. Ramos, "Bayesian optimisation for intelligent environmental monitoring," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 2242–2249, IEEE, 2012.

[52] D.-H. Cho, J.-S. Ha, S. Lee, S. Moon, and H.-L. Choi, "Informative path planning and mapping with multiple uavs in wind fields," *arXiv preprint arXiv:1610.01303*, 2016.

[53] T. Berger, Z. Zhang, and H. Viswanathan, "The ceo problem [multiterminal source coding]," *IEEE Transactions on Information Theory*, vol. 42, pp. 887–902, May 1996.

[54] D. L. Donoho, "Compressed sensing," *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[55] S. Jana and R. Blahut, "Berger-tung problem: A systematic approach based on canonical description," in *Information Theory and Its Applications, 2008. ISITA 2008. International Symposium on*, pp. 1–6, IEEE, 2008.

[56] H. Yin, J. Li, Y. Chai, and S. X. Yang, "A survey on distributed compressed sensing: theory and applications," *Frontiers of Computer Science*, vol. 8, no. 6, pp. 893–904, 2014.

[57] T. T. Do, Y. Chen, D. T. Nguyen, N. Nguyen, L. Gan, and T. D. Tran, "Distributed compressed video sensing," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pp. 1393–1396, IEEE, 2009.

[58] Y. Yang and Z. Xiong, "On the generalized gaussian ceo problem," *IEEE Transactions on Information Theory*, vol. 58, no. 6, pp. 3350–3372, 2012.

[59] K. Tiwari, V. Honoré, S. Jeong, N. Y. Chong, and M. P. Deisenroth, "Resource-constrained decentralized active sensing for multi-robot systems using distributed gaussian processes," in *2016 16th International Conference on Control, Automation and Systems (ICCAS)*, pp. 13–18, Oct 2016.

[60] K. Tiwari, S. Jeong, and N. Y. Chong, "Multi-uav resource constrained online monitoring of large-scale spatio-temporal environment with homing guarantee," in *Industrial Electronics Society, IECON 2017-43rd Annual Conference of the IEEE*, pp. 5893–5900, IEEE, 2017.

[61] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *Journal of Machine Learning Research*, vol. 9, no. Feb, pp. 235–284, 2008.

[62] N. Waters, "Tobler's First Law of Geography," *The International Encyclopedia of Geography*, 2017.

[63] T. Wilson and S. B. Williams, "Active sample selection in scalar fields exhibiting non-stationary noise with parametric heteroscedastic gaussian process regression," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pp. 6455–6462, IEEE, 2017.

[64] A. Singh, F. Ramos, H. D. Whyte, and W. J. Kaiser, "Modeling and decision making in spatio-temporal processes for environmental surveillance," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 5490–5497, IEEE, 2010.

[65] J. Chen, K. H. Low, Y. Yao, and P. Jaillet, "Gaussian process decentralized data fusion and active sensing for spatiotemporal traffic modeling and prediction in mobility-on-demand systems," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 901–921, 2015.

[66] R. Marchant and F. Ramos, "Bayesian optimisation for informative continuous path planning," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 6136–6143, IEEE, 2014.

[67] K. Tiwari, S. Jeong, and N. Y. Chong, "Point-wise fusion of distributed gaussian process experts (fudge) using a fully decentralized robot team operating in communication-devoid environment," vol. 34, pp. 820–828, IEEE, 2018.

[68] K. Tiwari, S. Jeong, and N. Y. Chong, "Map-reduce gaussian process (mr-gp) for multi-uav based environment monitoring with limited battery," in *Society of Instrument and Control Engineers of Japan (SICE), 2017 56th Annual Conference of the*, pp. 760–763, IEEE, 2017.

[69] S. Vasudevan, "Data Fusion with Gaussian Processes," *Robotics and Autonomous Systems*, vol. 60, no. 12, pp. 1528–1544, 2012.

[70] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte, "Heteroscedastic Gaussian Processes for Data Fusion in Large Scale Terrain Modeling," in *ICRA*, pp. 3452–3459, 2010.

[71] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte, "Non-Stationary Dependent Gaussian Processes for Data Fusion in Large-Scale Terrain Modeling," in *ICRA*, pp. 1875–1882, 2011.

[72] S. Vasudevan, A. Melkumyan, and S. Scheding, "Efficacy of Data Fusion using Convolved Multi -output Gaussian Processes," *Journal of Data Science*, pp. 341–367, 2015.

[73] M. P. Deisenroth and J. W. Ng, "Distributed Gaussian Processes," in *ICML*, vol. 2, p. 5, 2015.

[74] J. Chen, K. H. Low, Y. Yao, and P. Jaillet, "Gaussian Process Decentralized Data Fusion and Active Sensing for Spatiotemporal Traffic Modeling and Prediction in Mobility-on-Demand Systems," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 901–921, 2015.

[75] Y. Cao and D. J. Fleet, "Generalized Product of Experts for Automatic and Principled Fusion of Gaussian Process Predictions," *Modern Nonparametrics 3: Automating the Learning Pipeline workshop at NIPS, Montreal*, 2014.

[76] E. Meeds and S. Osindero, "An Alternative Infinite Mixture of Gaussian Process Experts," *Advances in Neural Information Processing Systems*, vol. 18, p. 883, 2006.

[77] C. Yuan and C. Neubauer, "Variational Mixture of Gaussian Process Experts," in *Advances in Neural Information Processing Systems*, pp. 1897–1904, 2009.

[78] S. E. Yuksel, J. N. Wilson, and P. D. Gader, "Twenty Years of Mixture of Experts," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, pp. 1177–1193, Aug 2012.

[79] V. Tresp, "A Bayesian Committee Machine," *Neural Computation*, vol. 12, no. 11, pp. 2719–2741, 2000.

[80] A. Viseras, T. Wiedemann, C. Manss, and L. Magel, "Decentralized Multi-Agent Exploration with Online-Learning of Gaussian Process," in *(ICRA)*, pp. 4222–4229, May 2016.

[81] F. Xue, R. Subbu, and P. Bonissone, "Locally Weighted Fusion of Multiple Predictive Models," in *IJCNN*, pp. 2137–2143, 2006.

[82] F. Lavancier and P. Rochet, "A General Procedure to Combine Estimators," *Computational Statistics & Data Analysis*, vol. 94, pp. 175–192, 2016.

[83] P. Baraldi, A. Cammi, F. Mangili, and E. E. Zio, "Local Fusion of an Ensemble of Models for the Reconstruction of Faulty Signals," *IEEE Transactions on Nuclear Science*, vol. 57, no. 2, pp. 793–806, 2010.

[84] R. Ouyang, K. H. Low, J. Chen, and P. Jaillet, "Multi-Robot Active Sensing of Non-Stationary Gaussian Process-based Environmental Phenomena," in *AAMAS*, pp. 573–580, 2014.

[85] C. E. Rasmussen and C. K. Williams, *Gaussian processes for machine learning*, vol. 1. MIT press Cambridge, 2006.

[86] M. Kuczma, *An Introduction to the Theory of Functional Equations and Inequalities: Cauchy's Equation and Jensen's Inequality*. Springer Science & Business Media, 2009.

[87] J. Sasiadek and P. Hartana, "Sensor data fusion using kalman filter," in *Information Fusion, 2000. FUSION 2000. Proceedings of the Third International Conference on*, vol. 2, pp. WED5–19, IEEE, 2000.

[88] J. Sasiadek, Q. Wang, and M. Zeremba, "Fuzzy adaptive kalman filtering for ins/gps data fusion," in *Intelligent Control, 2000. Proceedings of the 2000 IEEE International Symposium on*, pp. 181–186, IEEE, 2000.

[89] S.-L. Sun and Z.-L. Deng, "Multi-sensor optimal information fusion kalman filter," *Automatica*, vol. 40, no. 6, pp. 1017–1023, 2004.

[90] E. L. Lehmann and J. P. Romano, *Testing statistical hypotheses*. Springer Science & Business Media, 2006.

[91] 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*, pp. 449–456, ACM, 2007.

[92] S. Garg, A. Singh, and F. Ramos, "Learning non-stationary space-time models for environmental monitoring," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 25, p. 45, 2012.

[93] C. E. Rasmussen and C. K. Williams, *Gaussian processes for machine learning*, vol. 1. MIT press Cambridge, 2006.

[94] M. G. Genton, "Classes of kernels for machine learning: a statistics perspective," *Journal of machine learning research*, vol. 2, no. Dec, pp. 299–312, 2001.

[95] Y. Xu and J. Choi, "Adaptive sampling for learning gaussian processes using mobile sensor networks," *Sensors*, vol. 11, no. 3, pp. 3051–3066, 2011.

[96] S. M. Kay, *Fundamentals of statistical signal processing*. Prentice Hall PTR, 1993.

[97] A. Gretton, K. Fukumizu, C. H. Teo, L. Song, B. Schölkopf, and A. J. Smola, "A kernel statistical test of independence," in *Advances in neural information processing systems*, pp. 585–592, 2008.

[98] S. Kim, Z. Yu, R. M. Kil, and M. Lee, "Deep learning of support vector machines with class probability output networks," *Neural Networks*, vol. 64, pp. 19–28, 2015.

[99] F. Scholz, "Maximum likelihood estimation," *Encyclopedia of statistical sciences*, 1985.

[100] M. A. Osborne, S. J. Roberts, A. Rogers, S. D. Ramchurn, and N. R. Jennings, "Towards real-time information processing of sensor network data using computationally efficient multi-output gaussian processes," in *Proceedings of the 7th international conference on Information processing in sensor networks*, pp. 109–120, IEEE Computer Society, 2008.

[101] Empatica, "Real-time physiological signals-E4 EDA/GSR sensor." `https://www.empatica.com/e4-wristband`.

[102] G. Upton and I. Cook, *A dictionary of statistics 3e*. Oxford university press, 2014.

[103] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: experiences from the scikit-learn project," in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108–122, 2013.

[104] W. Lu, *Autonomous sensor path planning and control for active information gathering*. PhD thesis, Duke University, 2014.

[105] D. Levine, B. Luders, and J. P. How, "Information-rich path planning with general constraints using rapidly-exploring random trees," 2010.

[106] D. Ucinski, *Optimal measurement methods for distributed parameter system identification*. CRC Press, 2004.

[107] A. Ly, M. Marsman, J. Verhagen, R. Grasman, and E.-J. Wagenmakers, "A tutorial on fisher information," *arXiv preprint arXiv:1705.01064*, 2017.

[108] P. Inc., "Parrot Flight Recorder GPS for AR.Drone 2.0 Flying Drone." `https://tinyurl.com/yaf7krpv`. [Online; accessed 21-Dec-2017].

[109] G. US, "XBee ZB ZigBee Mesh Module 2.4GHz 2mW with Wire Antenna." `http://www.gravitech.us/xbzbmo22mwwa.html`. [Online; accessed 21-Dec-2017].

[110] K. Tiwari, "Hands-on experience with gaussian processes (gps): Implementing gps in python-i," *arXiv preprint arXiv:1809.01913*, 2018.

[111] K. Tiwari, "Hands-on Experience with Gaussian Processes (GPs)." `https://sites.google.com/view/exposure-to-gp/home`, 2018.

[112] K. Tiwari, X. Xiao, and N. Y. Chong, "Estimating achievable range of ground robots operating on single battery discharge for operational efficacy amelioration," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3991–3998, IEEE, Oct 2018.

[113] K. Tiwari, X. Xiao, A. Malik, and N. Y. Chong, "A unified framework for operational range estimation of mobile robots operating on a single discharge to avoid complete immobilization," *Mechatronics*, Oct. 2018. In Press.

[114] Y. Mei, Y.-H. Lu, Y. C. Hu, and C. G. Lee, "Energy-efficient motion planning for mobile robots," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 5, pp. 4344–4349, IEEE, 2004.

[115] D. Brooks, V. Tiwari, and M. Martonosi, *Wattch: A framework for architectural-level power analysis and optimizations*, vol. 28. ACM, 2000.

[116] O. Tremblay, L.-A. Dessaint, and A.-I. Dekkiche, "A generic battery model for the dynamic simulation of hybrid electric vehicles," in *Vehicle Power and Propulsion Conference, 2007. VPPC 2007. IEEE*, pp. 284–289, IEEE, 2007.

[117] A. Abdilla, A. Richards, and S. Burrow, "Power and endurance modelling of battery-powered rotorcraft," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 675–680, Sept 2015.

[118] A. Abdilla, A. Richards, and S. Burrow, "Endurance optimisation of battery-powered rotorcraft," in *Conference Towards Autonomous Robotic Systems*, pp. 1–12, Springer, 2015.

[119] D. Panigrahi, C. Chiasserini, S. Dey, R. Rao, A. Raghunathan, K. Lahiri, *et al.*, "Battery life estimation of mobile embedded systems," in *VLSI Design, 2001. Fourteenth International Conference on*, pp. 57–63, IEEE, 2001.

[120] F. Zhang, G. Liu, and L. Fang, "Battery state estimation using unscented kalman filter," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pp. 1863–1868, IEEE, 2009.

[121] M.-H. Chang, H.-P. Huang, and S.-W. Chang, "A new state of charge estimation method for lifepo4 battery packs used in robots," *Energies*, vol. 6, no. 4, pp. 2007–2030, 2013.

[122] Q. Miao, L. Xie, H. Cui, W. Liang, and M. Pecht, "Remaining useful life prediction of lithium-ion battery with unscented particle filter technique," *Microelectronics Reliability*, vol. 53, no. 6, pp. 805–810, 2013.

[123] L. Liao and F. Köttig, "Review of hybrid prognostics approaches for remaining useful life prediction of engineered systems, and an application to battery life prediction," *IEEE Transactions on Reliability*, vol. 63, no. 1, pp. 191–207, 2014.

[124] S. W. Kim and Y. H. Lee, "Combined rate and power adaptation in ds/cdma communications over nakagami fading channels," *IEEE Transactions on Communications*, vol. 48, no. 1, pp. 162–168, 2000.

[125] S. Thrun, W. Burgard, and D. Fox, "Probabilistic robotics. 2005," *Massachusetts Institute of Technology, USA*, 2005.

[126] A. Sadrpour, J. J. Jin, and A. G. Ulsoy, "Mission energy prediction for unmanned ground vehicles using real-time measurements and prior knowledge," *Journal of Field Robotics*, vol. 30, no. 3, pp. 399–414, 2013.

[127] M. Gatti, F. Giulietti, and M. Turci, "Maximum endurance for battery-powered rotary-wing aircraft," *Aerospace Science and Technology*, vol. 45, pp. 174–179, 2015.

[128] R. Mur-Artal, J. M. M. Montiel, and J. D. Tards, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, pp. 1147–1163, Oct 2015.

[129] Y. Mei, Y.-H. Lu, Y. C. Hu, and C. G. Lee, "Energy-efficient motion planning for mobile robots," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 5, pp. 4344–4349, IEEE, 2004.

[130] Y. Mei, Y.-H. Lu, Y. C. Hu, and C. G. Lee, "A case study of mobile robot's energy consumption and conservation techniques," in *Advanced Robotics, 2005. ICAR'05. Proceedings., 12th International Conference on*, pp. 492–497, IEEE, 2005.

[131] H. Sato, "Moving average filter," Oct 2001. US Patent 6,304,133.

[132] D. Graupe, A. A. Beex, and G. D. Causey, "Arma filter and method for designing the same," Feb 1980. US Patent 4,188,667.

[133] R. Manduchi, A. Castano, A. Talukder, and L. Matthies, "Obstacle detection and terrain classification for autonomous off-road navigation," *Autonomous robots*, vol. 18, no. 1, pp. 81–102, 2005.

[134] C. E. Vido and F. Ramos, "From grids to continuous occupancy maps through area kernels," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pp. 1043–1048, IEEE, 2016.

[135] Y. Xu, J. Choi, and S. Oh, "Mobile sensor network navigation using gaussian processes with truncated observations," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1118–1131, 2011.

[136] S. Choi, M. Jadaliha, J. Choi, and S. Oh, "Distributed gaussian process regression under localization uncertainty," *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 3, p. 031007, 2015.

[137] J. Ko and D. Fox, "Learning gp-bayesfilters via gaussian process latent variable models," *Autonomous Robots*, vol. 30, no. 1, pp. 3–23, 2011.

[138] S. Thrun and J. J. Leonard, "Simultaneous localization and mapping," in *Springer handbook of robotics*, pp. 871–889, Springer, 2008.

[139] A. Pierson, L. C. Figueiredo, L. C. Pimenta, and M. Schwager, "Adapting to sensing and actuation variations in multi-robot coverage," *The International Journal of Robotics Research*, vol. 36, no. 3, pp. 337–354, 2017.

[140] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[141] K. H. Low, N. Xu, J. Chen, K. K. Lim, and E. B. Özgül, "Generalized online sparse gaussian processes with application to persistent mobile robot localization," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 499–503, Springer, 2014.

[142] N. Xu, K. H. Low, J. Chen, K. K. Lim, and E. B. Ozgul, "Gp-localize: Persistent mobile robot localization using online sparse gaussian process observation model," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.*, pp. 2585–2593, 2014.

[143] H.-L. Choi and J. P. How, "Continuous trajectory planning of mobile sensors for informative forecasting," *Automatica*, vol. 46, no. 8, pp. 1266–1275, 2010.

[144] H. J. Bierens, "The inverse of a partitioned matrix." `http://www.math.chalmers.se/~rootzen/highdimensional/blockmatrixinverse.pdf`.

[145] D. Duvenaud, J. Lloyd, R. Grosse, J. Tenenbaum, and G. Zoubin, "Structure discovery in nonparametric regression through compositional kernel search," in *Proceedings of the 30th International Conference on Machine Learning* (S. Dasgupta and D. McAllester, eds.), vol. 28 of *Proceedings of Machine Learning Research*, (Atlanta, Georgia, USA), pp. 1166–1174, PMLR, 17–19 Jun 2013.

“ The best fighter is not the one who throws the most punches in the first round, but the one who is still standing to throw one in the last. ”

Kshitij Tiwari, *2017*