Active Sampling for Planetary Rover Exploration

Suhit Kodgule

CMU-RI-TR-19-66

August 6, 2019

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
David Wettergreen, *chair*
Maxim Likhachev
Alberto Candela

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

To my parents.

Abstract

Planetary Robotics research has expanded beyond simply developing robust navigation strategies for rovers to providing them with the capability of performing intelligent actions so as to develop a better interpretation and understanding of the environment. This will become essential in the future, when rovers explore regions far away from Earth, at distances that would significantly throttle communication with human operators. This research focuses on two problems of interest: Spatio-Spectral Exploration and Active Spectral Reconstruction. For both of these problems we develop a Markov Decision Process framework reliant on using remote sensing measurements that circumvents the partial observability present in these problems. Further, we propose a Monte-Carlo Tree Search based approach for efficiently sampling relevant locations so as to solve these problems. We demonstrate our approach in simulation as well as testing it with the rover Zoë at a Mars-analogous terrain located at Cuprite, NV and highlight its advantages compared to traditional planning strategies.

Acknowledgments

I would like to thank my advisor Prof. David Wettergreen for his incredible support throughout these two years. His expertise and enthusiasm in planetary robotics have always helped immensely in my research.

I would also like to thank Alberto Candela who collaborated with me and has contributed significantly in the latter parts of this work. Further, I would like to thank the Cuprite team members Srinivasan Vijayarangan and Kevin Edelson without whom the field experiments would not have been successful. I would also like to thank my lab mate Himanshi Yadav for her interest in my ideas and providing constructive criticism for them.

Lastly, I would like to thank my family, for their unwavering support over the last twenty years and my group of friends at CMU who made this experience even more wonderful.

This work was supported by the National Science Foundation National Robotics Initiative Grant #IIS-1526667. I gratefully acknowledge the assistance of Dr. David Thompson at Jet Propulsion Laboratory, California Institute of Technology.

Contents

When this dissertation is viewed as a PDF, the page header is a link to this Table of Contents.

List of Figures

List of Tables

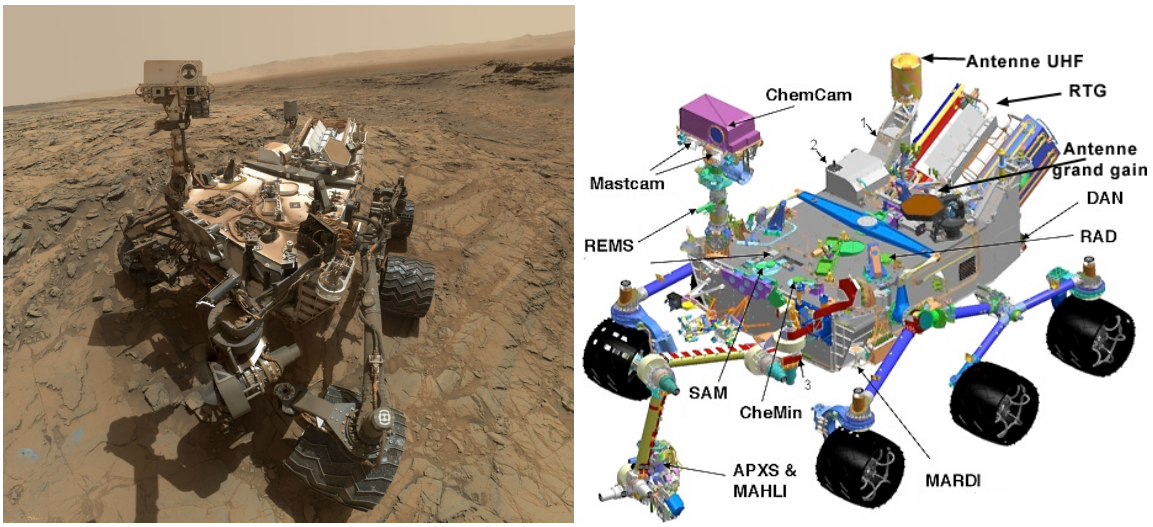
Chapter 1

Introduction

The era of robotic planetary exploration began with the Soviet Lunokhod 1 [9] landing on the Moon. Since then, numerous rovers such as Opportunity and Curiosity [22] have successfully explored extraterrestrial regions. As space robotics research has progressed over the years, focus has shifted from merely improving the mobility of the rovers to employing rovers to independently conduct scientific investigations with minimal human supervision. This is evident from the additional payload Curiosity rover handles compared to Opportunity, possessing twice as many scientific instruments as its counterpart (See Fig 1.1).

Historically, planetary robotics research has focused on risk-aware planning: in generating safe trajectories with an emphasis on hazard detection and avoidance and formulating risk-bounded temporal plans. However, with the advancement of hardware capabilities and planetary rovers expected to explore terrains with a greater degree of autonomy, research on incorporating planetary rovers with the ability to function without constant oversight has risen to prominence. This field of research is termed Science Autonomy which focuses on developing robotic technologies that improve a planetary rover's interpretation and understanding of the environment as well as providing them with the capability of undertaking abstract decisions that maximize certain science objectives.

Incorporating science information in the planning process however, leads to a large state-space that suffers from the *curse of dimensionality* [27]. Effort has been made from both planning and the reinforcement learning (RL) community to address



(a) Self-portrait of Curiosity on Mars

(b) Instruments on Curiosity

Figure 1.1: (a): Self-portrait captured by Curiosity during its traverse on Mars. (b): Instruments on-board utilized for scientific analysis.

this. For instance, Hollinger et al. [25] extended sampling based methods such as RRT* for information gathering scenarios with modular objective functions. From the RL perspective, employing function approximators or using *Monte-Carlo sampling* proved to be beneficial in breaking this curse [42]. Banking on the success of RL methods for large state-spaces, we propose a Monte-Carlo Tree Search (MCTS) based method for science autonomy applications.

In particular, we focus on two problems in science autonomy: Spatio-Spectral Exploration (SSE) and Active Spectral Reconstruction. Both these problems are of significant importance in the fields of Imaging Spectroscopy and Space Robotics. While SSE focuses on interpreting the orbital sensing measurements, Active Spectral Reconstruction attempts to provide a compact representation of these measurements. A commonality between these two problems is that they both rely on a generative model of the environment. We demonstrate how this generative model can be incorporated into the MCTS framework for improved sampling strategies.

Formally, this thesis makes the following contributions. We propose Markov Decision Process (MDP) frameworks for two applications of science autonomy, i.e., SSE and Active Spectral Reconstruction. We describe how solving these problems involves accounting for a non-stationary environment with partial observability, but

how utilizing orbital sensing measurements allows us to approximate this into a fully observable MDP. Further, we propose a MCTS based planner for solving this MDP and provide rigorous comparative analysis with other planning strategies.

This thesis is organized as follows. Chapter 2 provides background on RL and some of the techniques developed to address the *curse of dimensionality*. Chapter 3 details Science Autonomy, its development over the years and explains SSE and Active Spectral Reconstruction in detail. Further, it develops the MDP frameworks and the MCTS solvers required for solving these problems. Chapter 4 details the simulation environment and performs analysis of the solvers for both the problems. Chapter 5 validates the conclusions derived in the previous chapter for Active Spectral Reconstruction at a test site in Cuprite, NV using the planetary rover Zoë. Finally Chapter 6 delves on concluding thoughts as well the scope for future work in this domain.

Chapter 2

Reinforcement Learning Background

Consider an agent navigating in an uncertain environment. The agent can interact with the environment and observe some rewards on account of its interactions. The planning problem in this scenario is to choose actions that maximize the cumulative reward observed by the agent. Reinforcement learning tries to solve this problem by mapping states to actions so as to gain the maximum reward. The optimal action to choose is not the one that provides the maximum reward at the next time-step but the one that would provide the highest cumulative reward over a finite or infinite horizon.

Reinforcement learning consists of four main subelements: a *policy*, a *reward signal*, a *value function* and, sometimes a *model* of the environment [42]. A *policy* defines the behavior of an agent. Specifically, it is a mapping from states of the environment to the actions of an agent. A *reward signal* characterizes a state of the environment. Based on whether the state is good or bad, a positive or negative reward is observed by the agent. The agent's sole objective is to maximize the cumulative reward over its lifetime. A *value function* characterizes the long-term effects of taking an action from a particular state. It can roughly be thought of as the expected cumulative reward the agent is expected to observe from the current time-step. Finally, a *model* of the environment defines what states the agent will transition to next from its current state after following an action. Models are used by the agent in the planning process:

in choosing the next action by rolling out the future states and observing the *value*.

Some background concepts pertaining to reinforcement learning are useful for understanding the problem and we begin with a review of the building block of reinforcement learning: MDP; followed by elaborating on some of the solvers developed for the MDP framework.

2.1 Markov Decision Process

An MDP can be defined as a four-tuple consisting of set of states, a set of actions, a transition function and a reward function denoted by (S, A, T, R). MDPs rely on the Markov assumption which states that the distribution of the future states only depends on the current state. This implies that the state representation must contain all components required for the agent to learn the optimal policy.

The state-space may be discrete or continuous depending on the problem, although finding an optimal policy for continuous states requires a different set of solvers. This thesis focuses solely on discrete state-space. The action space is the set of discrete or continuous actions the agent can take in an environment. The actions are a way for the agent to interact with the environment and observe rewards. The transition function is defined as $T : S \times A \times S' \to \mathbb{R}$. It defines a distribution over states the agent will transition to from the current state after applying an action. The reward function: $R : S \times A \to \mathbb{R}$ assigns a scalar value to each (S, A) pair. A policy is deemed to be the optimal policy if following it provides the optimal value function. Formally, this is defined as:

$$\pi^*(s) = \underset{\pi}{\operatorname{argmax}}\, V^{\pi}(s) \tag{2.1}$$

2.2 MDP Solvers

There have been numerous methods of classifying RL algorithms. Some of them include whether model of the environment is used, whether a policy is learnt by parameterizing it in the form of a neural network or whether function approximators or tabular representations are used for the RL environment. However, RL algorithms

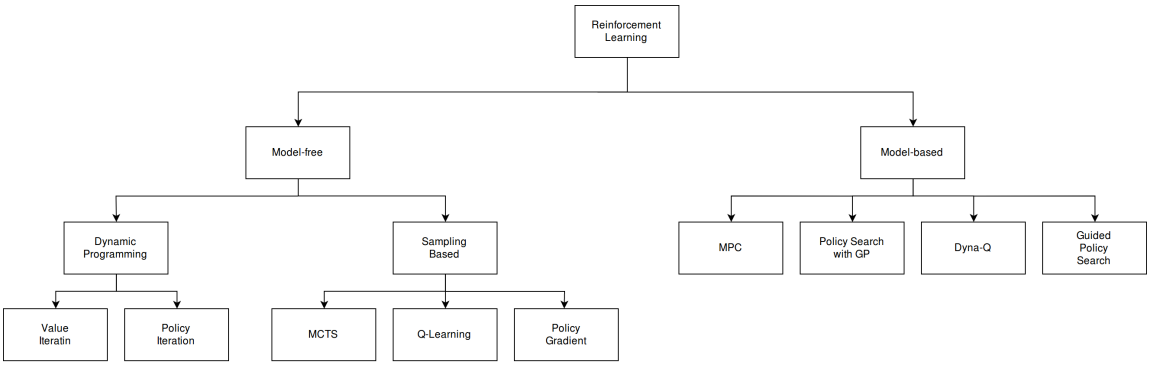


Figure 2.1: Classification of Reinforcement Learning Algorithms as model-based and model-free methods.

can be broadly classified into model-based and model-free algorithms as shown in Figure 2.1. Model-based methods explicitly learn the transition dynamics or the reward functions by sampling the environment and then conduct planning using the learnt models to obtain the optimal policy. Model-free methods on the other hand, directly learn the value function or even the policy itself by sampling states from the environment. Model-free methods can be further categorized as Dynamic programming methods and sampling methods. Dynamic Programming methods require access to the entire state space for computing the values, whereas sampling based methods sample a small portion of the state space to compute the values for required states.

2.2.1 Value Iteration

Value Iteration (VI) is a dynamic programming method which requires information of all the states in the environment. VI iteratively computes the value function using the Bellman optimality equation and has been proven to converge to the optimal value function [3], [4].

$$V_{k+1}(s) \leftarrow \max_{a} \sum_{s'} P\left(s'|s,a\right)\left(R\left(s,a,s'\right)+\gamma V_k\left(s'\right)\right) \tag{2.2}$$

Here, γ is the discount factor that provides higher preference for more immediate states, $P(s'|s,a)$ the probability of transitioning into state s' after taking action a

from state s, $R(s, a, s')$ is the reward observed after the transition and finally $V_k(s')$ is the value of state s' from the previous iteration. This method does not explicitly learn a policy but only focuses on learning the optimal value function. The optimal policy is then simply the deterministic policy that chooses the action with the maximum value for the current state.

2.2.2 Policy Iteration

Policy Iteration (PI) is also a dynamic programming. Unlike VI, it directly manipulates the policy in these two stages: (1) Policy Evaluation and (2) Policy Improvement. Policy evaluation consists of applying the Bellman expectation equation until convergence to get the value function for the current policy [26].

$$V_{\pi_k}(s) = \mathbb{E}\left[R_{t+1} + \gamma R_{t+2} + \ldots | S_t = s\right] \tag{2.3}$$

Policy improvement then follows that consists of updating the policy by choosing the action with the maximum Q-value.

$$\pi_{k+1}(s) := \arg\max_{\mathrm{a}} \left(R(s, a) + \gamma \sum_{s' \in S} T\left(s, a, s'\right) V_{\pi_k}\left(s'\right) \right) \tag{2.4}$$

This algorithm also produces a deterministic policy. Both VI and PI are *synchronous* dynamic programming algorithms which update all states in every iteration.

2.2.3 Monte-Carlo Tree Search

One of the major drawbacks of VI and PI is they require updating every state in the environment every iteration. This becomes infeasible in cases of large state-spaces. Moreover, determining the value for states that are rarely, if never to be visited will simply waste valuable computation resources. On account of this, tree-based methods were introduced that compute values only from the current state. MCTS [10] is one such algorithm. This algorithm can only be applied to finite-horizon planning problems. MCTS is a partial tree search algorithm that efficiently balances the exploration versus exploitation trade-off. Each node in the tree corresponds to a

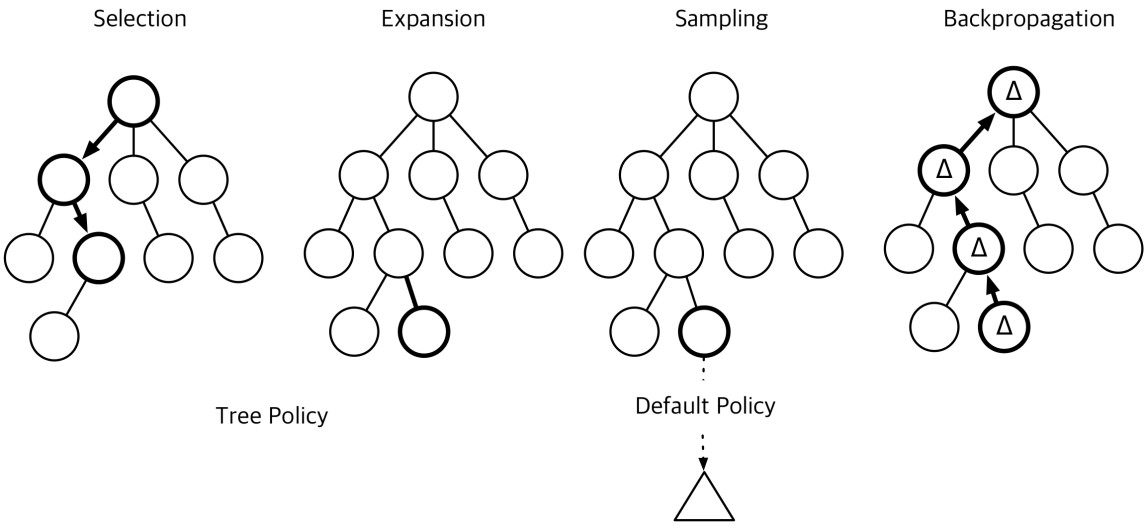


Figure 2.2: Four stages of MCTS. Selection: Selecting a node to expand. Expansion: Expanding the tree with a new node. Sampling: Rolling out till end of horizon for computing values. Backpropagation: Updating the values of parent nodes with observed rewards.

particular state of the MDP. Applying an action from a state results in a successor state and some reward.

If an action of the current node has not been tried before, that action is given precedence. The algorithm then rolls out from the resulting successor state until a fixed depth and returns the cumulative discounted reward. On the other hand, if all feasible actions of a state have been tried,the next action to take is based on the Upper Confidence Bound metric that efficiently trades-off exploration against exploitation. This process is repeated for the successor state until a fixed depth is reached.

The first new successor state resulting from either of the two cases is appended to the tree and the reward is backed up to the root of the tree. Finally, the action leading to the successor with the maximum cumulative reward from the root is returned as shown in Figure 2.2.

2.2.4 Q-Learning

Q-Learning [49] is an off-policy algorithm that learns the optimal Q-value. A Q-value is the cumulative reward gained by following a particular action from a state. An off-policy algorithm follows a different policy (such as ϵ-greedy policy) from the optimal policy while learning the Q-values to promote exploration of the state-space. The Q-values are updated iteratively based on (2.5).

$$Q_{k+1}(s,a) = Q_k(s,a) + \alpha \left[r(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) \max_{a'} Q_k(s',a') - Q_k(s,a) \right] \tag{2.5}$$

The ϵ-greedy policy favours a non-optimal action with some probability ϵ to promote exploration of the state space. It is defined as:

$$\pi\left(\mathbf{a}_t|\mathbf{s}_t\right) = \begin{cases} 1-\epsilon \text{ if } \mathbf{a}_t = \text{argmax}_{\mathbf{a}_t} Q\left(\mathbf{s}_t, \mathbf{a}_t\right) \\ \epsilon/(|\mathcal{A}|-1) \text{ otherwise} \end{cases} \tag{2.6}$$

This avoids the algorithm to get stuck in some local minima of the Q function. Once the Q function is learnt, the agent follows a greedy policy for maximizing its rewards.

2.3 Summary

Unlike traditional deep learning and machine learning techniques, reinforcement learning requires real-time decision making on account of an interactive agent that navigates an environment. Moreover, the primary objective of RL algorithms is to maximize a reward function instead of finding hidden structure in the data as in the case of unsupervised learning. The RL problem is formally defined in the MDP framework and various solvers have been introduced to solve it. MDP solvers are essentially categorized based on whether the exploration is motivated by developing a representation of the environment (model-based) or by improving generalization power of the optimal policy (model-free). For problems with large state-spaces, it it infeasible to apply dynamic programming methods and as a result, sampling based methods were developed. Sampling-based methods such as MCTS, can also function

as anytime algorithms that provide a solution to the problem. However, one of the drawbacks of MCTS is it can only be applied on a finite-horizon problem. Recently, function approximation based approaches such as DQN have been introduced that parameterize the Q-function or the policy as a deep neural network. DQN [33] is an extension of Q-learning that uses a deep convolutional neural network for approximating the optimal Q-function. In the applications that we are concerned with however, designing a suitable function approximator is non-trivial. Moreover, on account of the large state-space of our problems and requiring a solution within the prescribed time limit, we rely on MCTS for solving our environment. MCTS is suitable for our problem on account of its advantages such as being an anytime algorithm and being a sampling based algorithm.

Chapter 3

Science Autonomy

This chapter provides a background on the field of Science Autonomy. We go over the science objectives this field tackles in general and delve into the specific science autonomy objectives that are the focus of this thesis; namely, SSE and Active Spectral Reconstruction.

3.1 Related Work

There has been much work on developing efficient representations of science phenomena of interest. Thompson [45] focused on developing spatial models of the environment: a generative map that extrapolates from previous observations to predict measurements at future locations. Spatial models facilitate adaptive, online learning such that an agent can efficiently update the models in real-time. The work demonstrated an effective spatial model representation using Gaussian Processes that infused orbital data as latent input dimensions into the model. A key difference of this work from previous research on *topological mapping* [19], such as mapping an indoor environment is that unlike spatial modelling, topological mapping does not extrapolate beyond the sensing-horizon of the robot. One of the reasons why spatial modelling is able to achieve this is because of strong cross-sensor correlation such as between orbital sensing data and *in situ* sensory data. For instance, geological studies of the reflectance spectra measured by these sensors can characterize the minerals present at the surface. Comparing the spectral signatures of *in situ* measurement with the corresponding

orbital spectra at a particular location can allow us to make predictions of the minerals present at locations where similar orbital spectra are observed. Borrowing ideas from information-driven sampling for sensor networks [28], Thompson further proposed information gain based science objectives for spatial modelling.

Building upon Thompson's work, Foil [15] drew parallels between sequential modelling, a topic similar to spatial modelling (with a key distinction that sequential modelling allows non-stationary environment models) and active learning. Further Foil proposed a Dirichlet Process based Adaptive Gaussian Mixture Model (AGMM) model and a Gaussian Process based model for adaptive sampling: a form of sequential modelling with non-stationary objective functions. Numerous acquisition functions for the Gaussian Process model were explored and compared against traditional planning strategies such as Maximum Entropy Sampling [39], random sampling and grid search.

Furlong [17] proposed three algorithms for improving the performance of rovers conducting science autonomy. First, the foraging algorithm improved autonomy by deciding between sampling immediately or searching for better options. Second, the thresholding algorithm quantified the level of confidence the rover has in detecting whether a change has occurred. Lastly, the final algorithm involves rejecting unlikely hypotheses from a collection of hypotheses proposed by scientists for choosing sampling actions by using prior and *in situ* information. Recently, Candela *et al.* developed the science hypothesis map: a probabilistic structure in which the initial beliefs of scientists evolves as the robot takes measurements.

While significant work has been done in improving environment models for science autonomy, research on improving the planning capabilities of the rover that would use these models for determining optimal sampling locations is still in the nascent stage; with planners either relying on science-blind strategies or myopic strategies that only attempt to maximize the immediate information content. There is a dire need then on more sophisticated strategies that are computationally efficient as well as providing better convergence. We now look over some of the work from sister fields like Informative Path Planning (IPP) and discuss how techniques developed there can be translated to science autonomy.

The problem of of making a robot autonomously decide which path to take while collecting measurements is termed as IPP. On a cursory analysis, this problem may

seem indistinguishable from sequential modelling and indeed, some of the applications of sequential modelling can be brought under the umbrella of IPP problems. However, traditionally IPP has only focused on scalar sensory measurements such as the plankton density at Chesapeake Bay or the RF signal intensity field [11]. We introduce this additional nomenclature in this thesis merely to distinguish it from sequential modelling that also tackles high-dimensional sensory input such as the multi-spectral spectrometer data. Binney *et al.* [6] introduced the concept of IPP and an exhaustive search strategy that exploited the monotonicity of certain objective functions for faster convergence. Lim *et al.* [31] developed Recursive Adaptive Identification (RAId), an adaptive IPP algorithm with polylogarithmic approximation bound for an agent traversing in metric space. RAId was tested on a number of tasks such as robot grasping a 2-star graph search and was shown to achieve the best or nearly the best policies in all the tasks. Cao *et al.* [8] proposed two multi-agent IPP algorithms based on entropy and mutual information as metric functions and demonstrate that they perform better than state-of-the-art algorithms with increasing planning horizons. Employing entropy or mutual information as metric functions has been used in several IPP works with a fair degree of success [11], [32]. Consequently, we utilize entropy based objective functions as well which are formulated in the subsequent sections.

Of particular importance for this work are the tree-based algorithms developed for IPP problems. Morere *et al.* [34] developed a Bayesian Optimization based Partially Observable Markov Decision Process (POMDP) framework that utilized a variant of MCTS for finding approximate solutions to monitor a spatial phenomena using an Unmanned Aerial Vehicle (UAV). Clary *et al.* [12] similarly used MCTS for legged locomotion. Various other works have also focused on employing MCTS for trajectory planning [2], [24]. Ross *et al.* proposed a no-regret algorithm for learning non-stationary policies using Imitation Learning [37]. However, acquiring relevant demonstrations for science autonomy problems is non-trivial. Banking on the success of current reinforcement learning (RL) research, we also propose MCTS based algorithms for two science autonomy applications. Now, we elaborate on the science autonomy applications we are interested in and succinctly formulate them.

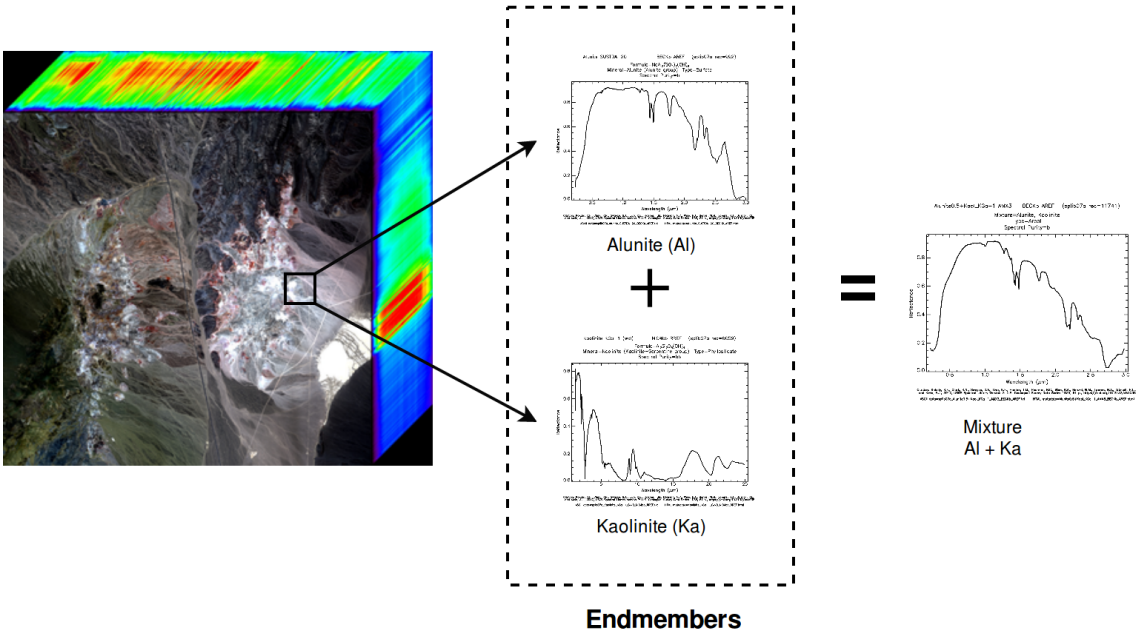


Figure 3.1: Linear Mixing Model: Exploded-view of a pixel of an orbital image from a location in Cuprite, Nevada. The pixel consists of two endmembers Alunite and Kaolinite which linearly combine to form the observed spectrum.

3.2 Spectral Mixture Models and Spatio-Spectral Exploration

One of the prominent requirements of geologic exploration is disambiguating pure minerals from orbital sensing measurements. This branch of geology is termed as *spectral unmixing* and also forms a part of a planetary exploration scenario known as spatio-spectral exploration. We detail both these topics in the forthcoming sections.

3.2.1 Spectral Mixture Models

Orbital hyperspectral sensors have poor spatial resolution on account of measuring from high altitudes. As a consequence, these sensors record scenes in which numerous distinct minerals contribute to the same spectrum that is measured from a single pixel of the hyperspectral image. This gives rise to mixed pixels, which is formed by a combination of pure minerals called *endmembers*. The fractions in which these

endmembers combine to form the mixed pixel are known as *abundances*. We consider here a linear mixing model (LMM) which assumes that the mixed pixel's spectrum is formed by a linear combination of the endmembers that constitute the pixel. Figure 3.1 displays the process. Formally, if a spectrum has d bands or channels, we can define a pixel of an orbital image by a vector $x \in \mathbb{R}^{\mathrm{d}}$. Further, if there are K endmembers observed in the entire orbital image, then each pixel x_j of an orbital image X can be expressed as:

$$x_j = \sum_{i=1}^{K} a_{ij} y_i \quad \forall x_j \in X \tag{3.1}$$

where $y_i \in \mathbb{R}^{\mathrm{d}}$ is the i_{th} endmember spectrum and a_{ij} is the abundance scalar for i_{th} endmember and j_{th} pixel. For physical realizability, endmembers must not have negative abundances, that is: $a_{ij} \geq 0,\ \forall i, j$ and $\sum_{i=1}^{K} a_{ij} = 1,\ \forall j$. Equation (3.1) can be written as $x_j = Y a_j,\ \forall x_j \in X$, where Y is a library of K endmember spectra: $Y = [y_1,\ y_2,\ \ldots\ , y_K]$.

An unmixing problem attempts to find the abundance vector $a_j = [a_{1j},\ a_{2j},\ \ldots\ a_{Kj}]^\top$ for each pixel x_j of X given Y. Typically, this is solved using non-negative least squares optimization [30] and involves solving the following objective for each pixel:

$$\begin{aligned} &\underset{a_j}{\operatorname{argmin}} \quad \|Y a_j - x_j\|_2 \\ &s.t.\ a_{ij} \geq 0 \quad \forall a_{ij} \in a_j \end{aligned} \tag{3.2}$$

3.2.2 Spatio-Spectral Exploration

SSE involves collecting a library of endmembers Y that can solve an unmixing problem. Consider a robot that collects N *in situ* spectral measurements along a path to form a library of endmembers. Let the set of sampling locations along the path be B. Further, let the *in situ* spectral measurements be defined as $y = f(b) + \epsilon,\ \forall b \in B$, where ϵ is a sensor noise sampled from a normal distribution: $\epsilon \sim \mathcal{N}(0, \sigma)$. The set of all *in situ* measurements on path B is then $Y_B = \{y_i : y_i \in \mathbb{R}^{\mathrm{d}},\ 1 \leq \mathrm{i} \leq \mathrm{N}\}$. Formally,

the robot must collect samples that minimize the objective:

$$O(B) = E\left[\sum_{x_j \in X} \min_{a_j} \|Y_B a_j - x_j\|_2\right] \tag{3.3}$$

$$s.t.\ a_{ij} \geq 0\ \forall a_{ij} \in a_j,\ C(B) \leq \beta$$

where $C(B)$ is the sampling cost for the rover.

In its present form, (3.3) cannot be solved during the traverse as all the elements of Y_B are not available. However, the solution can be approximated by replacing the elements in Y_B of unvisited locations by the corresponding remote sensing measurements. Let $V_t \subseteq B$ be the set of locations the rover has sampled until time t and Y_{V_t} be the corresponding *in situ* spectra. Further, let $D_t \subseteq B$ be the set of unvisited locations and X_{D_t} be the set of remote sensing measurements at those locations. Then as demonstrated in [46], solving (3.3) is analogous to solving the following objective function:

$$O(B|V_t) = \sum_{x_j \in X} \min_{a_j} \|[Y_{V_t}\ X_{D_t}]a_j - x\|_2 \tag{3.4}$$

$$s.t.\ a_{ij} \geq 0\ \forall a_{ij} \in a_j,\ C(V_t) + C(D_t) \leq \beta$$

3.3 MDP for Spatio-Spectral Exploration

We now formulate the MDP required for solving the problems described in the previous sections. While there are some differences between the MDPs for SSE and Active Spectral Reconstruction, the foundational structure and the reasoning behind their formulation is the same. Therefore, we first explain the MDP for SSE and then only describe the differences in formulating the MDP for Active Spectral Reconstruction.

We first address the inherent uncertainty present in the environment on account of *in situ* sensing. As the future *in situ* spectra will not be available at present time, this makes our environment partially observable. While it is certainly possible to apply POMDP based techniques to this problem, we display how using the inherent structure of SSE allows us to reduce this problem to solving multiple MDPs with

complete observability.

3.3.1 State Definition

Formally, we define the state s as a tuple (Y, V, X_D, D, X) where V, D are sets of visited and unvisited locations respectively, Y is the set of *in situ* spectra collected from the visited locations, X_D is the set of remote observations corresponding to the locations in D and X is the orbital image. This definition is necessary to hold the Markov assumption. However, a consequence of including previous measurements into the state definition is that the size of the state-space increases significantly. In this regard, utilizing MCTS for finding the optimal policy is beneficial as the branching factor for the tree is only dependent on the size of the action-space. We limit our action-space to 8 actions to maintain tractability.

3.3.2 Action Definition

Let the set of all actions be A. As we do not have access to *in situ* measurements, we consider an action as adding a remote observation measurement to the spectral library. Thus, an action in this MDP corresponds to adding a new location to D and the corresponding remote sensing observation to X_D. An action is said to be valid if the new location is reachable from the current rover's position. We assume the rover can only traverse on an eight-connected grid and the reachable locations are the immediate neighboring locations on the grid. For verbosity, let the set of valid actions for a state s be $\Omega(s)$. The transition function T, is deterministic as every action from a particular state has a unique successor.

Notice that in this definition of MDP, the number of *in situ* measurements remains fixed for a particular MDP. Thus, at each time step t, all the states for $MDP(t)$ possess only those *in situ* measurements collected till time t, where $MDP(t)$ refers to the MDP generated at time t. This ensures full observability. While we cannot utilize Equation (3.3) for solving this MDP, the simplification described in Equation (3.4) can be fully defined using the state and actions as described by our MDP. This then allows us to develop planning strategies while maintaining tractability.

3.3.3 Reward Definition

Computing NNLS error and LS error for a given library of spectra is computationally expensive on account of the matrix inversions required for computing its solution. Consequently, it is infeasible to define our reward function as either of the two metrics. We chose to use the differential entropy as a reward for solving this MDP. Differential entropy has a closed form solution and is independent of the size of the orbital image. Moreover, differential entropy has been previously applied for science-aware exploration and shown to provide competent results [20, 45]. A brief description of it is given below.

Let X be a random variable with a probability density function f whose support is χ. Then the differential entropy is defined as [13]:

$$h(X) = -\int_{\chi} f(x) log\left(f(x)\right) dx. \tag{3.5}$$

For a multivariate Gaussian random variable with covariance Σ, (3.5) has the following closed form solution [1]:

$$h(X) = \frac{1}{2} \ln |2\pi \mathrm{e} \Sigma|. \tag{3.6}$$

For a state s in our MDP, we define our reward function $R(s)$ as:

$$R(s) = h(S) = \frac{1}{2} \ln |2\pi \mathrm{e} \Sigma_{S,S}| - \tau U(S). \tag{3.7}$$

where $S = [Y,\ X_D]$, $U(S)$ is a function that penalizes paths with multiple visits to the same locations and τ is a scalar weight.

3.3.4 Non-myopic Planner for Spatio-Spectral Exploration

Here, we describe our planner named Non-Myopic Planner for Spatio-Spectral Exploration (NMPSE). Our approach comprises of creating and solving multiple trees using MCTS in a sequential manner for each new MDP generated. The algorithm is shown in Algorithm 1. First, the current state is passed to the MCTS solver. The algorithm constructs a tree with the root as the current state and returns the optimal action for that state. The optimal action for this MDP consists of a sampling location

on the map. Next, an *in situ* spectral measurement is taken at the location alluded by the optimal action. A new state is created by appending the new location to the set of visited locations and the sampled *in situ* spectral measurement to the set of *in situ* measurements. This is represented by the $Traverse$ function in the algorithm For this new state, we again construct a tree employing MCTS with the new state as the root node and find the next optimal action. In this way, additional *in situ* samples are collected until the sampling budget is exhausted. We now explain MCTS as applied to SSE.

MCTS is a partial tree search algorithm that efficiently balances the exploration versus exploitation trade-off. Each node in the tree corresponds to a particular state s of the MDP. Applying an action a from s results in a successor state s'. This is depicted in the $Step$ function. The function returns s' and a reward r' which is computed according to Equation (3.7). The algorithm only expands nodes that result from applying an action $b \in \Omega(s)$.

If an action of the current node has not been tried before, that action is given precedence. The algorithm then rolls out from the resulting successor state until a fixed depth and returns the cumulative discounted reward. On the other hand, if all feasible actions of a state have been tried, the next action to take is based on the Upper Confidence Bound metric that efficiently trades-off exploration against exploitation. Line 28 in Algorithm 1 shows this specific metric. This process is repeated for the successor state until a fixed depth is reached.

The first new successor state resulting from either of the two cases is appended to the tree and the reward is backed up to the root of the tree. Finally, the action leading to the successor with the maximum cumulative reward from the root is returned. Note that all the nodes of the tree constructed from a particular MCTS call will have the same set of visited locations as no action in the MDP causes the rover to physically move to another location.

A key distinction between our method and traditional MCTS approaches is we use a particular MCTS tree to find just one optimal action. Once an optimal action has been found, the agent executes the action and constructs a new MCTS tree from the successor state. This is necessary as the previous tree does not possess information about the additional *in situ* measurement.

Algorithm 1: NMPSE

10 **function** `NMPSE`(v_0, y_0)
  $V \leftarrow v_0$, $Y \leftarrow y_0$
  $X_D, D \leftarrow \emptyset$
  $s \leftarrow \{Y, V, X_D, D, X\}$
  while $|Y| < \beta$ **do**
    $T(root) \leftarrow s$
    $a^* \leftarrow \text{MCTS}(T)$
    $v_{new}, y_{new} \leftarrow \text{Traverse}(a^*)$
    $V \leftarrow \{V \cup v_{new}\}$
    $Y \leftarrow \{Y \cup y_{new}\}$
    $X_D, D \leftarrow \emptyset$
    $s \leftarrow \{Y, V, X_D, D, X\}$
  end
function `MCTS`(T)
  for $i = 0$ **to** *max_iterations* **do**
    Simulate($T(root)$, 0)
  end
  return $\underset{a}{\text{argmax}}\, Q(s_0, a)$
function `Simulate`(s, *depth*)
  if $\gamma^{depth} < \epsilon$ **then**
    return 0
  end
  if s *has untried actions* **then**
    Sample a from untried actions
    $(s', r) \leftarrow \text{Step}(s,a)$
    return $r + \gamma \cdot \text{Rollout}(s', depth + 1)$
  end
  $a \leftarrow \underset{b \in \Omega(s)}{\text{argmax}}\, Q(s, b) + \kappa\sqrt{\frac{log(N)}{N_a}}$
  $(s', r') \leftarrow Step(s, a)$
  $G \leftarrow r' + \gamma \cdot \text{Simulate}(s', depth + 1)$
  $N(s) \leftarrow N(s) + 1$
  $N(sa) \leftarrow N(sa) + 1$
  $Q(s, a) \leftarrow Q(s, a) + \frac{G - Q(s,a)}{N(sa)}$
  return G
function `Rollout`(s, *depth*)
  if $\gamma^{depth} < \epsilon$ **then**
    return 0
  end
  $a \sim \Omega(s)$
  $(s', r') \leftarrow \text{Step}(s, a)$
  return $r' + \gamma \cdot \text{Rollout}(s', depth + 1)$
function `Step`(s, a)
  $\{Y, V, X_D, D, X\} \leftarrow$ Unwrap(s)
  $d_{new}, x_{new} \leftarrow$ Unwrap(a)
  $s' \leftarrow \{Y, V, \{X_D \cup x_{new}\}, \{D \cup d_{new}\}, X\}$
  $r' \leftarrow Reward(s')$
  return $(s',\ r')$

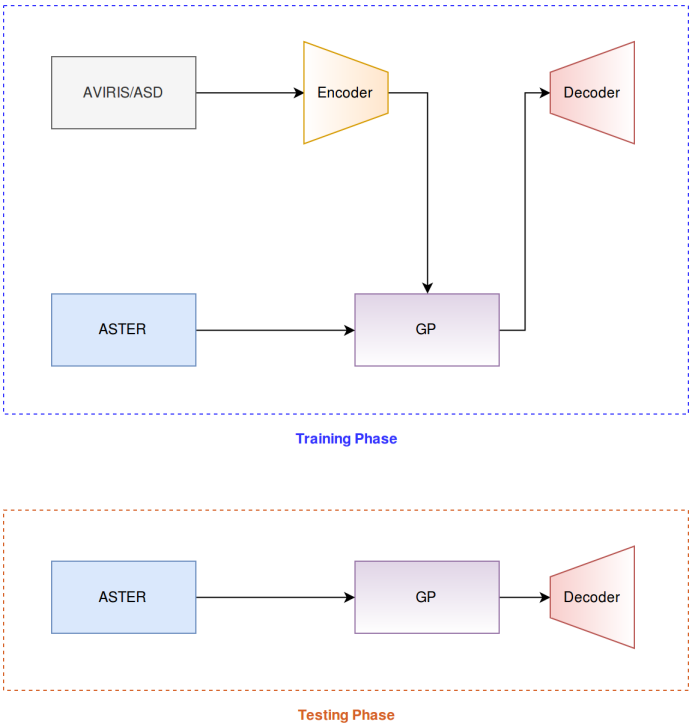


Figure 3.2: Architecture for Spectral Prediction

3.4 Active Spectral Reconstruction

Instead of using *spectral unmixing* to determine the pure spectra, some research has also focused on developing a generative model that predicts the pure spectra using orbital measurements. This model will be updated online as new training samples are obtained by the rover. We now focus on one such method and explain how MCTS planning process can be applied to this problem.

3.4.1 Architecture Description

Figure 3.2 displays the training and testing process for this method. During training, *in situ* spectra acquired from the rover are mapped to lower dimensions using an autoencoder. The compressed spectra are used as training labels for the Gaussian Process (GP) [35] module which takes as input the corresponding orbital spectrum at a particular location. The GP model consists of six independent GPs, each of which predict one of the dimensions of the compressed *in situ* spectra. The predictions from

the GP module are passed into the decoder network of the autoencoder that maps the predictions to the higher dimensional space of the *in situ* spectra.

During the testing phase, The orbital spectral measurements are directly passed into the GP module to obtain the compressed *in situ* spectra. These are further passed through the decoder to get predictions of *in situ* spectra at unvisited locations.

3.5 MDP for Superresolution

As mentioned previously, we only focus on the key differences between this MDP and the one used for SSE. In order to understand this formulation, it is first imperative to understand how the reward is computed. Thus, after defining the state-space, we focus on defining the reward.

3.5.1 State Definition

Unlike the state defined in Section 3.3.1, we do not require the orbital spectra of unvisited locations. This is because the reward of this MDP is dependent on the GP model. Thus, instead of the orbital spectra, the state now consists of the GP model. The complete state tuple then is (Y, V, G, D, X) where V, D are the sets of visited and unvisited locations, Y is the set of *in situ* spectra collected from visited locations, GP is the GP model and X is the orbital image.

3.5.2 Reward Definition

The reward is defined on the confidence of the GP model or, conversely, on the uncertainty of its predictions. We define uncertainty as the net variance of the GP's predictions over the entire map. This variance can be directly computed as follows:

$$\sigma^2(S^*) = K(S^*, S^*) - K(S^*, X) \cdot K_X^{-1} \cdot K(X, S^*) \tag{3.8}$$

where S^* is the set of unvisited locations, $K(S^*, S^*)$ is the covariance matrix of the GP for unvisited locations with respect to itself, $K(S^*, X)$ is the covariance matrix of unvisited and visited locations and K_X^{-1} is the covariance matrix of the visited

locations. The reward for our MDP then, is simply negative variance as defined in (3.8).

3.5.3 Action Definition

The action is simply defined as querying the GP model to get the updated reward for an unvisited location. Once an action is executed, we develop a state in the MCTS tree and cache the reward for future use.

3.5.4 MCTS planner for Active Spectral Reconstruction

Here, we describe our planner simply defined as MCTS. Our approach is similar to NMPSE, with the sole difference being the reward function used for this problem is defined in (3.8). Similarly, the *Traverse* function is now also modified to instead include the set of pseudo-visited locations and caching the reward for future use. Apart from this difference, the MCTS planner is analogous to NMPSE and continues in the same way. In the next chapter, we focus on describing the simulation environment as well as discussing the performance of our planners.

3.6 Summary

This chapter introduced the concept of science autonomy and expounded the progress that has been made in this field. The work in this field can be classified into two categories: research on developing mathematical models for representing the science phenomena and research on navigation strategies developed to obtain information pertaining to these models. While significant work has been done on the first part, research in the latter field has not observed as much progress. For two problems of SSE and Active Spectral Reconstruction, we proposed an MDP formulation employing orbital sensing measurements and *in situ* spectral measurements. Further we proposed an MCTS based planning method for solving these MDPs.

Chapter 4

Simulation Experiments

4.1 Environment Description

We simulate an exploration scenario using a high resolution data acquisition system. Specifically, we use the data obtained by Airborne Visible Near Infrared Spectrometer - New Generation (AVIRIS-NG) [21, 23] as proxy for *in situ* spectra. For remote sensing measurements, we use data obtained by the Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) [16].

4.1.1 AVIRIS-NG

AVIRIS-NG observes spectra in the range of $[0.38\mu m,\ 2.5\mu m]$ at a spectral resolution of $5nm$ and an exceptionally high spatial resolution in the range of $[0.3m,\ 4m]$. Spectra are measured as images with 600 cross-track elements, providing it with $> 95\%$ cross-track spectral uniformity and $>= 95\%$ spectral IFOV uniformity. The system also consists of an INS/GPS and an on-board calibrator that performs automated calibration of raw spectra. The system's navigation data is also used to access surface elevation information from a global topographic dataset [21]. Studies have shown that AVIRIS-NG measurements are a good analog to *in situ* infrared spectra [48], with inferences drawn on AVIRIS-NG transferring well onto data collected by *in situ* sensing instruments. AVIRIS-NG has been used in several science applications such as atmosphere correction [18], ecology and vegetation [36], geology and soil [43]

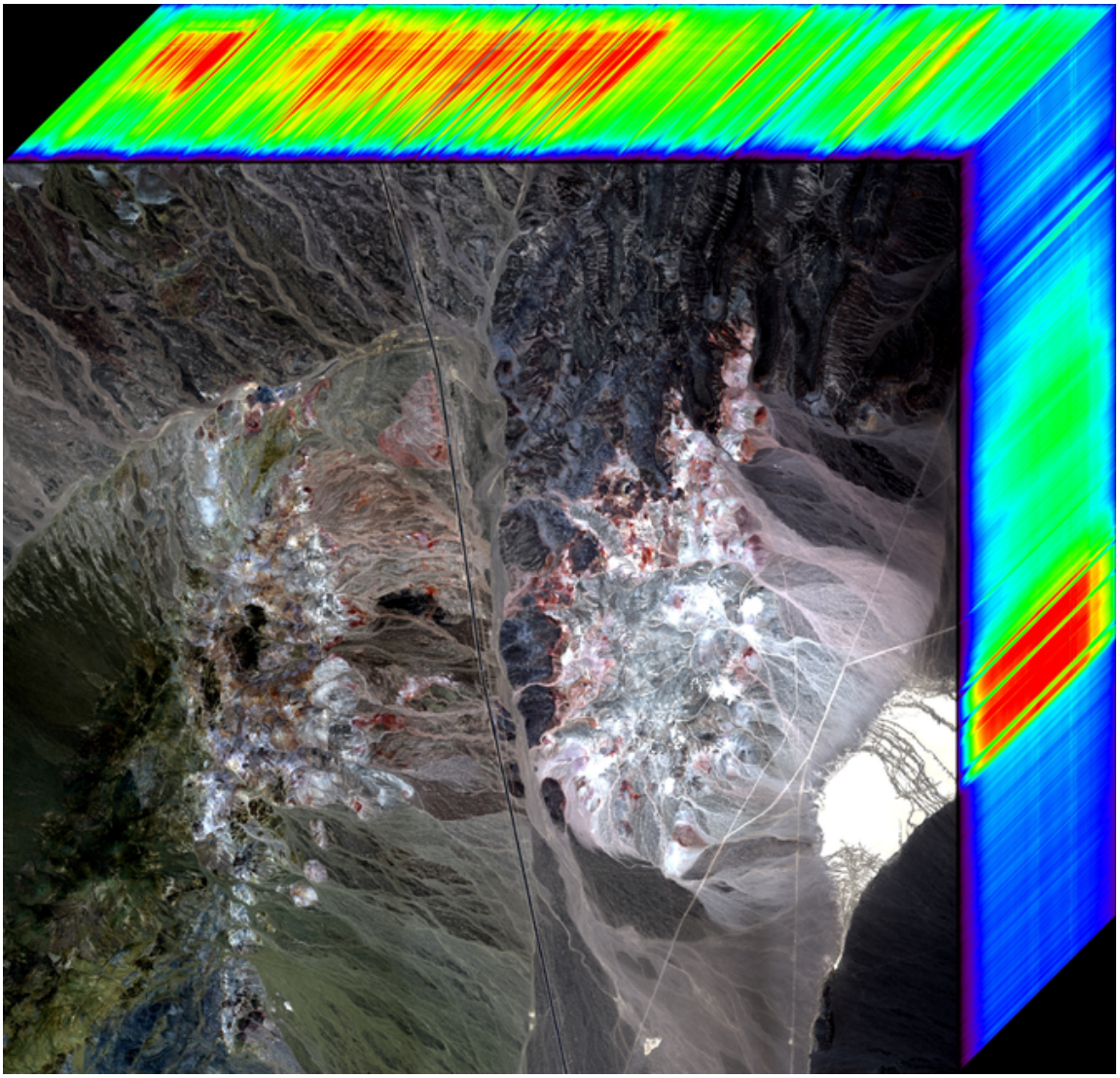


Figure 4.1: Hyperspectral Image of the Cuprite mining site in Nevada.

Wavelength	380 nm to 2510 nm
Spectral Resolution (FWHM, minimum)	5nm ± 0.5 nm
Field of View	36 ± 2 degrees with 600 resolved elements
Instantaneous Field of View	1.0 1.4 mrad ± mrad
Spatial Sampling	1.0 mrad ± 0.1 mrad
Spectral Distortion (smile)	Uniformity >97%
Spectral Distortion (keystone)	Unformity >97%
FPA	480 (spectral direction) X 640 (cross track)
Frame Rate	10 - 100 frames per second
Pixel Size	27 microns x 27 microns
Calibration	On-board calibrator
Data Resolution	14 bits
Data Rate	Up to 74 MB/s of throughput
Data Volume	Up to 1.0 TB of raw data
Physical Volume	83 cm (H) x 57 cm (Dia.)
Mass	465kg
Vacuum Requirement	10-4 torr
Ambient Operating Temperature	-40 to +50C
Maximum Altitude	18 km

Table 4.1: AVIRIS-NG Specifications

as well as coastal and inland waters [38]. A detailed specification of AVIRIS-NG instrument is provided in Table 4.1.

4.1.2 ASTER

ASTER is an imaging spectrometer onboard Terra, the sattellite of of NASA's Earth Observation System. The instrument consists of multiple cameras possessing three visible short-wave infrared (VSWIR) and six short-wave infrared (SWIR) bands. Compared to AVIRIS-NG, ASTER has a lower spatial resolution in the range of $[15m, 90m]$. This necessitates using the finer AVIRIS-NG measurements to disambiguate the minerals present on the surface.

Several studies have used AVIRIS-NG and ASTER measurements as proxies for low and high resolution spectral measurements, respectively [29, 46, 47]. We evaluate our approach on measurements taken at a mining district in Cuprite, Nevada; a well-studied site with high mineralogical diversity [44]. Figure 4.1 shows an example

of the AVIRIS-NG spectroscopic map of Cuprite. We were able to associate the two instruments' observations by aligning them with respect to both their spatial and spectral dimensions. We first registered both images with a planar homography approach. We then used the empirical line method [40] to find the correspondence between the ASTER and AVIRIS-NG reflectance values.

4.2 Spectral Mixture Models Experiments

We now evaluate the simulation results for the Spectral Unmixing problem. The evaluation is conducted in three stages. First, NMPSE is evaluated with itself to determine the optimal MCTS parameters. Next, it is compared against two uniformed planners which are planners that do not use any science information for navigation. Finally, it is compared against the current best planner for Spectral Unmixing: Greedy Spectrum Selection (GSS) which is an informed planner.

4.2.1 Planners

The two uninformed planners: Random and Fixed Step and the informed planer GSS are now described in detail.

- Fixed Step: This is an exhaustive strategy sequentially samples waypoints uniformly along one direction until the sampling budget is exhausted. If the rover reaches the end of the map before the budget is exhausted, it moves one step to the right and continues sampling in the opposite direction.
- Random: In this case, the robot randomly samples a location from the reachable locations of the robot. The next location is sampled from a uniform distribution over the neighboring locations. In this strategy, there is a possibility of the rover randomly choosing to sample a previously visited location.
- Greedy Spectrum Selection: GSS is an informed planner that greedily selects the next best waypoint. The algorithm is displayed in Algorithm 2. This planner is also known as Maximum Entropy Sampling in the planetary robotics community. Intuitively, this algorithm always samples the most informative location provided it respects the traversal budget. If the sampling location requires more traversal budget than required, it iteratively finds the next most

Algorithm 2: Greedy Spectral Selection

```
function GSS(V_start, V_end, X, β, W)
    Q ← {V_start, V_end}
    while C(Q) < β do
        forall v ∈ W \ Q do
            X_V ← {X_Q, X_v}
            R(Q ∪ v) = ½ ln |2πeΣ_{X_V,X_V}|
            if R(Q ∪ v) > R* then
                R* ← R(Q ∪ v)
                v* ← v
            end
        end
        Q ← {Q, v*}
    end
```

informative point that satisfies the traversal budget. The objective function that quantifies the "informativeness" of a point can be varied. For instance Thompson et al. [46] used the NNLS metric defined in 3.4. In our work, the metric used is the MES loss instead of NNLS loss. It greedily selects the next sample from a set of waypoints that maximizes (3.6) under some traversal constraint. It does not adaptively replan the path after each *in situ* spectral measurement is collected. In order to conduct a fair comparison between GSS and NMPSE, we provide the same start locations for both the algorithms. Further, we define the waypoints as all the points on the eight-connected grid which our rover can traverse in the MDP. In addition, GSS algorithm requires a goal location and a path traversal budget. We pass the same goal location and the traversal cost observed during NMPSE evaluation for a particular initial rover configuration.

We used mean reconstruction error (MRE) of the collected samples as the performance metric in all of our evaluations. The metric is simply (3.3) averaged over 50 trial runs. Apart from comparing NMPSE with the planners mentioned above, we also evaluate the effects of varying the hyperparameters of the MCTS in NMPSE on its computational efficiency.

Table 4.2: Comparison of Mean Reconstruction Error (RE) and average computation time for MCTS with different maximum tree depths.

Depth	Mean RE NMPSE	Average Action Time (sec)
5	783.64	**0.7451**
7	776.1989	1.0570
10	**758.3488**	1.5304

4.2.2 Evaluation of MCTS Parameters

We first evaluate the performance of NMPSE by varying the depth of the MCTS. Increasing the depth of the tree would cause the planner to consider longer horizons in its planning strategy. This in turn, would lead to an increase in the computation cost. The discount factor γ was kept at 0.9 and the number of Monte-Carlo simulations during each MCTS call were kept at 500. MRE and the average action time which is the mean time taken by MCTS to provide one optimal action is provided in Table 4.2. The MRE decreased with increasing depth. However, the average action time increased at a faster rate. A depth of 5 seemed ideal for this problem as it allowed us to keep the computation time at less that one second. All the experiments explained below are implemented with 500 Monte-Carlo simulations, a max depth of 5 and a discount factor of 0.9.

4.2.3 Evaluation of NMPSE

We divided the evaluation of NMPSE into two parts. The first evaluation was done by constraining the number of samples the rover can collect and the second evaluation was conducted by constraining the path length. A detailed explanation of each of the evaluations is given below.

4.2.4 Constraining Path Length

As GSS is constrained by traversal cost instead of sampling cost, it is more equitable to compare the performance of the planners against path length. The path cost is defined as 10 units for traversing to one of the neighbors of a point on the eight-connected grid. In a real-world scenario, this is equivalent to traversing approximately 300m at

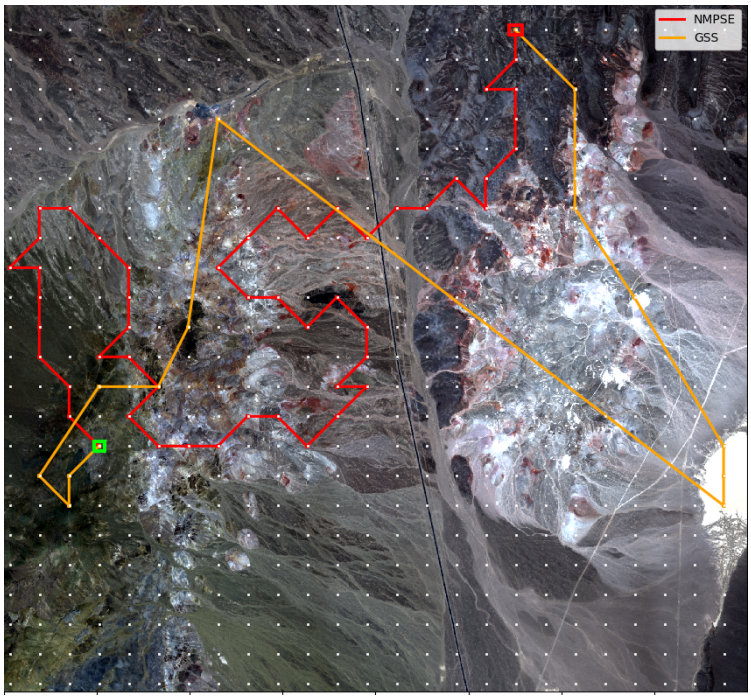


Figure 4.2: Example paths from NMPSE (red) and GSS (orange) when constrained by traversal budget overlaid on the map of Cuprite mining site. The white dots on image refer to discretized grid of sampling points. The green and red squares correspond to start and end locations of the traverse.

Table 4.3: Comparison of Mean Reconstruction Error (MRE) and t-test for NMPSE and GSS

Path Length	MRE NMPSE	MRE GSS	$p < 0.05$
250	**1200.79**	2533.62	Y
500	**783.64**	2033.13	Y
750	**592.19**	1066.78	Y
1000	**469.61**	563.38	Y
1250	396.32	**372.53**	**N**

the Cuprite Site. NMPSE outperforms GSS in all but one case. Surprisingly, GSS is able to achieve comparable performance only with a traversal cost of 1000 and above. It was expected that GSS would more efficiently choose waypoints with a smaller path budget. One explanation for this discrepancy is as GSS is myopic in nature, it finds it difficult to choose additional waypoints that do not exceed the traversal cost with a smaller path budget. Figure 4.2 demonstrates this behavior. GSS greedily samples the most informative location, despite it requiring a significant portion of the traversal budget. This leaves it with a small portion of the budget in the latter parts of the traverse.

Figure 4.3 displays the corresponding boxplot. MRE for both the approaches and whether statistically significant difference was observed is displayed in Table 4.3. We evaluated statistical significance with a one-tailed t-test: comparing NMPSE with GSS. While GSS achieved better performance than NMPSE for path length of 1250, it did not achieve statistical significance. Moreover, from a practical point of view, rover traverses rarely exceed more than 1-2 km in a day [50] which roughly translates to around 750 units of path length in our simulated experiment. NMPSE then would provide better results compared to GSS in field experiments.

4.2.5 Constraining Sampling Budget

We evaluated NMPSE against two naive planning approaches; namely, random search and fixed step sampling. Figure 4.4 displays the boxplots for each of the planners while varying the sampling budget. NMPSE significantly outperformed both random sampling and fixed sampling planners. With increasing sampling budget, NMPSE reduced the reconstruction error at a faster rate than the other two planners. MRE and whether statistical significance ($p < 0.05$) was achieved is displayed in Table 4.4. We evaluated statistical significance with two one-tailed t-tests: comparing NMPSE with random sampling and fixed sampling. As shown, NMPSE achieved a lower mean with statistical significance in all but one case. We believe NMPSE was not able to achieve $p < 0.05$ in that case due to the sample size being too small. Figure 4.5 displays a path obtained from each of the planning strategies from one of the 50 experiments. Observe that NMPSE spends a significant sampling budget in areas with high mineral diversity that lie on the right portion of the image.

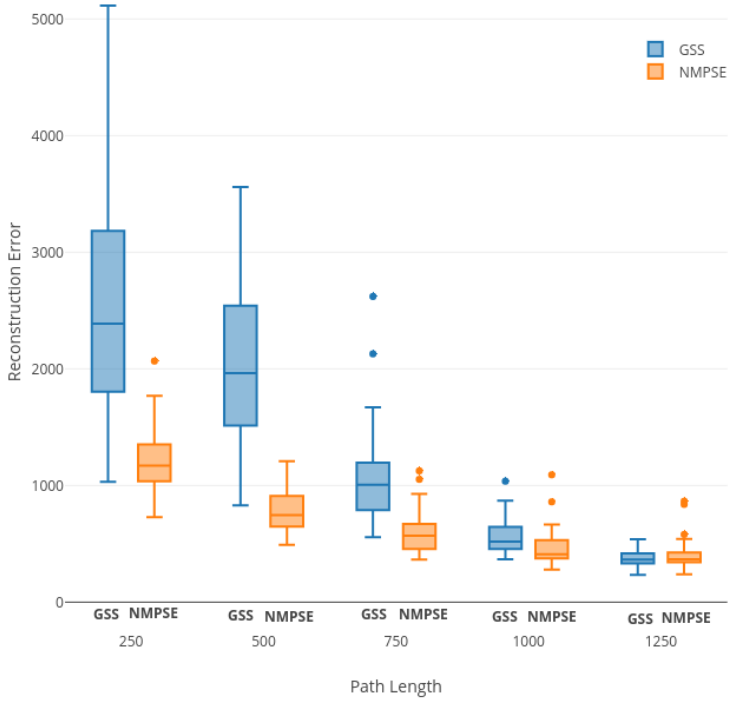


Figure 4.3: Comparison of the performance of NMPSE (orange) against GSS (blue) with varying path lengths. Both planners were initialized with the same rover configurations.

Table 4.4: Comparison of Mean Reconstruction Error (MRE) and t-test for NMPSE, Fixed Sampling and Random Sampling planners with different sampling budgets.

Samples	MRE Fixed	MRE NMPSE	MRE Random	Fixed $p < 0.05$	Random $p < 0.05$
25	1269.95	**1200.79**	1779.89	**N**	Y
50	999.96	**783.64**	1333.05	Y	Y
75	949.32	**592.19**	1163.42	Y	Y
100	976.8	**469.61**	1090.02	Y	Y
125	879.55	**396.32**	974.09	Y	Y

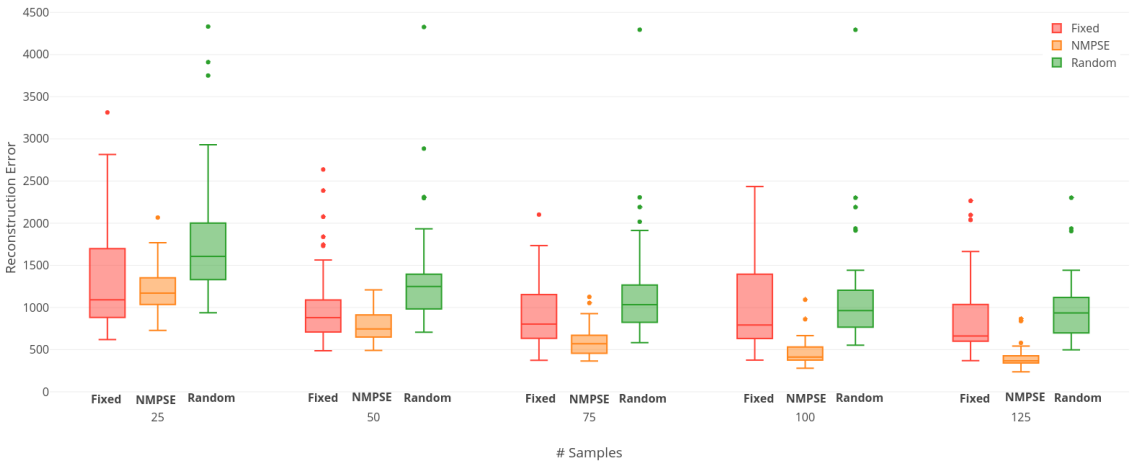


Figure 4.4: Comparison of the performance of NMPSE (orange) with Fixed Sampling (red) and Random Sampling (green) with varying sampling budgets. All planners were initialized with the same rover configuration.

4.3 Active Spectral Reconstruction Experiments

We simulate an exploration scenario with an environment similar to one described in Section 4.1. A key difference is the rover is now forced to follow slope constraints. As the Cuprite region is relatively hilly, there are several regions which are not traversable for a rover. Consequently, we define certain no-go zones wherever the slope of the terrain is greater that 18°. The no-go zones also include regions that the rover is unable to travel to from its current location as depicted in Figure 4.6.

The baseline planners used are also different to match the field requirements. Instead of using GSS, we now use a greedy planner that is constrained to navigate on the eight-connected grid, similar to NMPSE. This is because the traversal region is significantly smaller in size. As GSS is seen to perform better only with a large traversal budget, it will not construe as a relevant baseline for MCTS. The other planner we use is the Random planner, which remains the same as previous experiments. The reward for these planners are also changed to take into consideration the GP model as described in section 3.5.2. We simulated traverses at three test sites we were likely to visit for the field experiment [7]. The test sites are displayed in Figure 4.7. All of these test sites demonstrate an abundance of spectral diversity that is critical for our analysis. We simulated 100 traverses with different start locations and evaluated

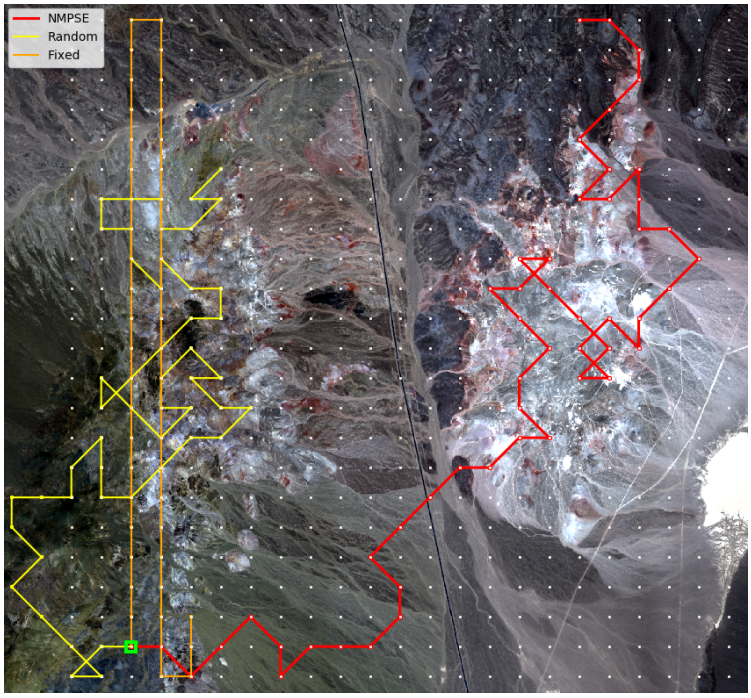


Figure 4.5: Example paths from NMPSE (red), Random (yellow) and Fixed Step (orange) when constrained by sampling budget overlaid on the map of Cuprite mining site. The white dots on the image refer to discretized grid of sampling points. The green square corresponds to the start location.

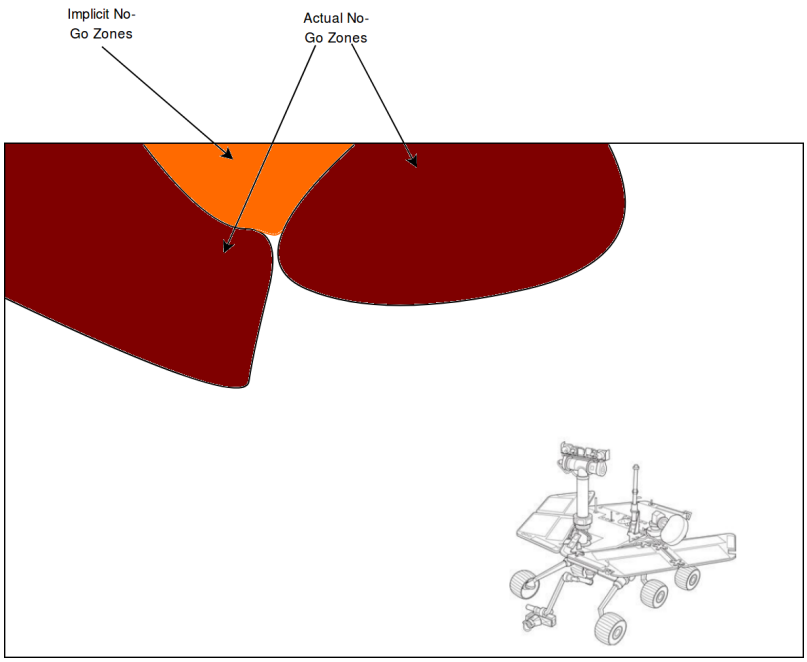


Figure 4.6: Demonstration of how no-go zones are defined for the rover. The orange region also becomes untraversable despite having acceptable slope on account of the rover being unable to reach there.

the planners based on Root Mean Squared Error (RMSE) of predicted samples and ground truth and the Shannon entropy of the GP model.

We now analyze the performance of the planners for test site A. As expected, RMSE is directly correlated with entropy. This is expected as a reduction in entropy indicates a higher degree of confidence in GP predictions. Figure 4.8a displays this relation. While this is a trivial result to arrive at, it has an important implication: the planners can be evaluated using just the entropy of the final GP model instead of the RMSE. This conclusion will become useful in field experiments where gathering groundtruth data of the large Cuprite region is infeasible, whereas computing the entropy is simple. However, we do provide RMSE plots in field where we approximate the spectra from AVIRIS-NG as *in situ* spectra.

We analyse the performance of the planner using results from Site A. We found the results to be consistent across all three test sites and thus, any conclusions drawn from Site A results are also true for Sites B and C. Figure 4.8 displays the results for Site A. Analyzing the RMSE of the planners from Figure 4.8e, we notice that it is monotonic with respect to the samples collected. This is expected as RMSE is a sub-modular function of samples collected. In this case, MCTS fares better than the rest of the planners although it achieves statistical significance only after collecting 7 samples. Statistical significance was determined using one-tailed paired t-test. Figure 4.8g and 4.8h display the t-test results. The dotted red line indicates $p = 0.05$.

4.4 Summary

This chapter simulated exploration scenarios with AVIRIS-NG measurements as a proxy for *in situ* measurements and ASTER measurements as orbital measurements for both SSE and Active Spectral Reconstruction. In the case of SSE, comparison against Random, Fixed Step and GSS planners demonstrated that NMPSE provided lower reconstruction error with statistical significance as measured by a one-tailed t-test. NMPSE was observed to navigate to regions with higher spectral diversity leading to its lower reconstruction error. Moreover, it was observed that GSS required a significant traversal budget to obtain comparable results. Performing analysis of GSS also revealed a deficiency in its approach: provided that the next sampling location satisfies the traversal budget, GSS will choose that waypoint even though

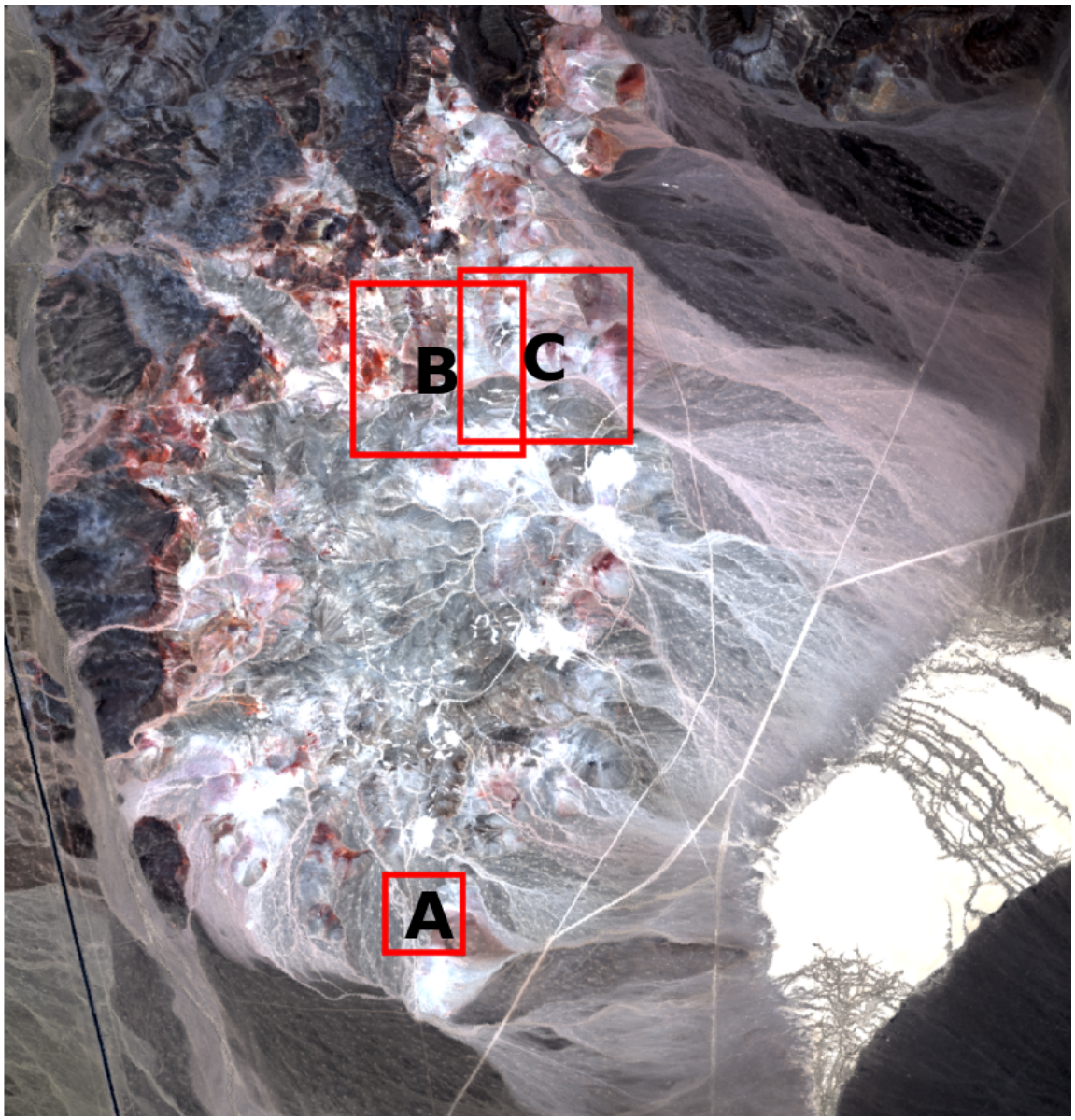
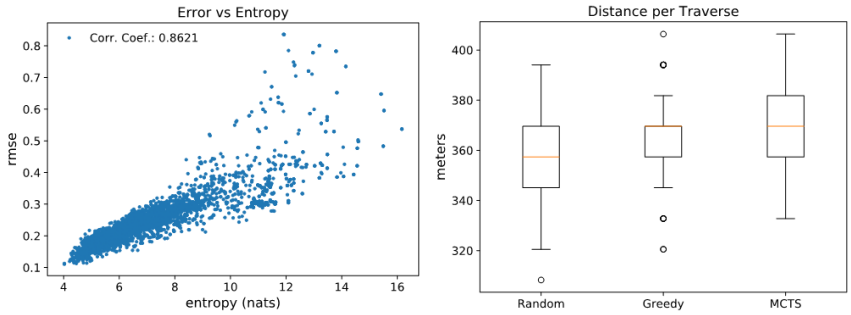
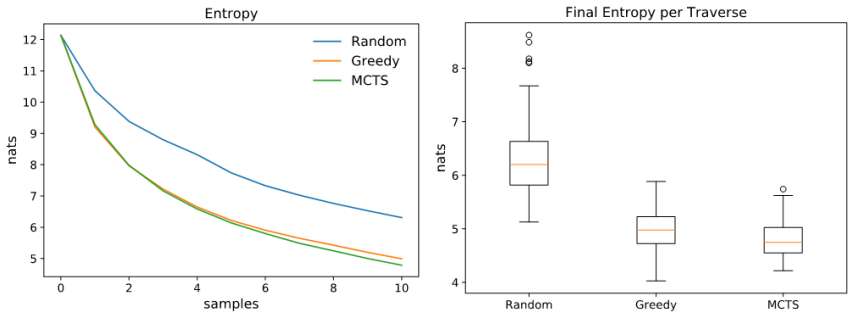


Figure 4.7: Test Site locations for validating MCTS planners at Cuprite, NV.



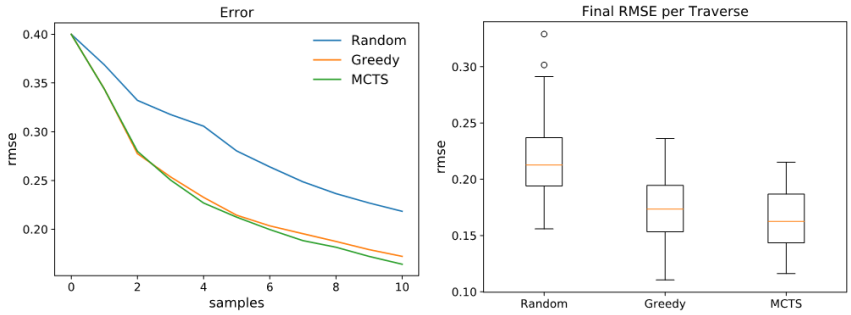
(a) Correlation of Entropy against RMSE

(b) Box plots comparing distance travelled for each planner



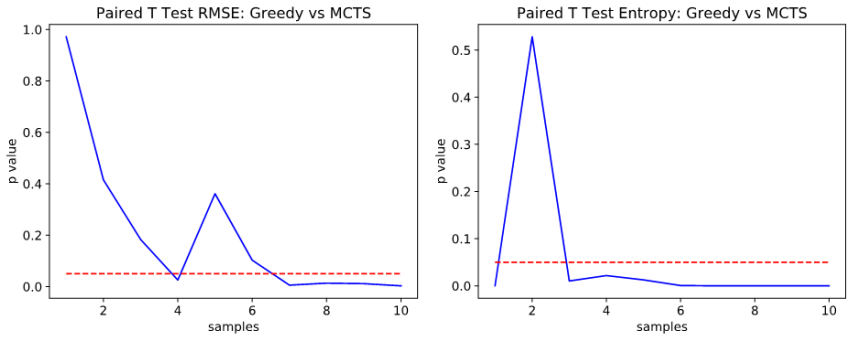
(c) Entropy against samples collected for each planner

(d) Box plots comparing Entropy for each planner



(e) RMSE against samples collected for each planner

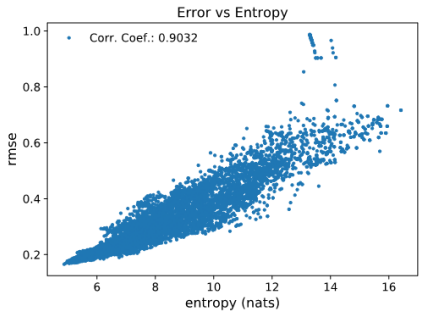
(f) Box plots comparing RMSE for each planner



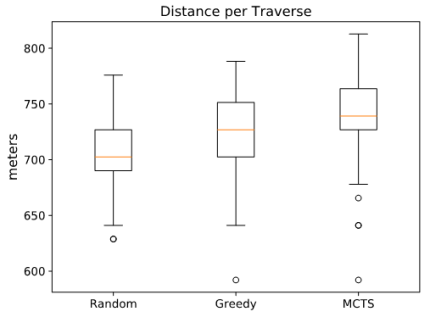
(g) P-Values for RMSE against samples collected

(h) P-Values for entropy against samples collected

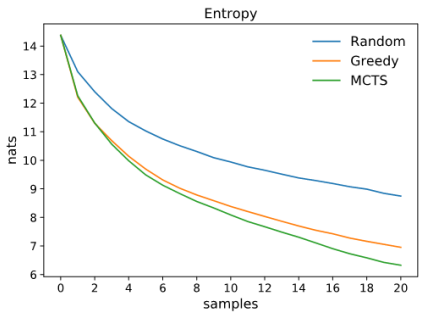
Figure 4.8: Simulation results for Active Spectral Reconstruction at Test Site A.



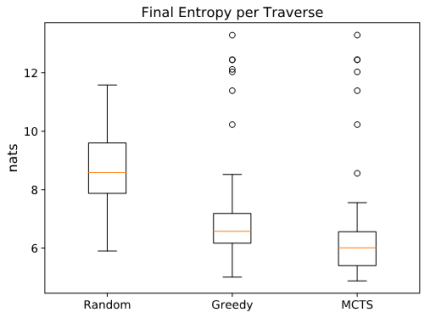
(a) Correlation of Entropy against RMSE



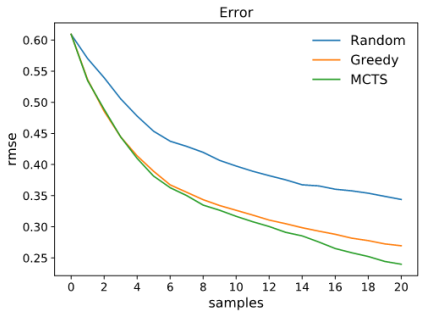
(b) Box plots comparing distance travelled for each planner



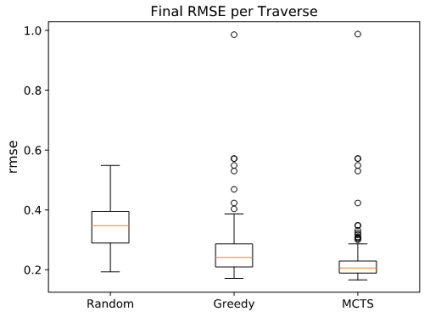
(c) Entropy against samples collected for each planner



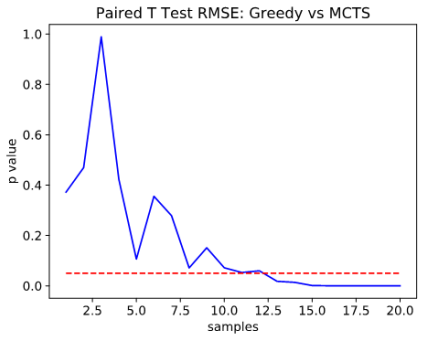
(d) Box plots comparing Entropy for each planner



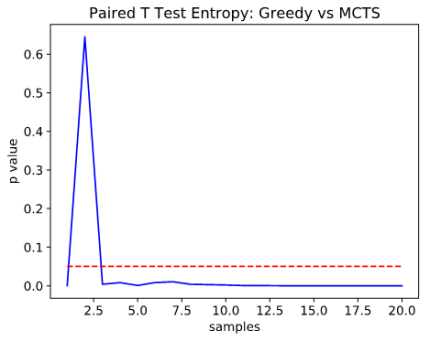
(e) RMSE against samples collected for each planner



(f) Box plots comparing RMSE for each planner

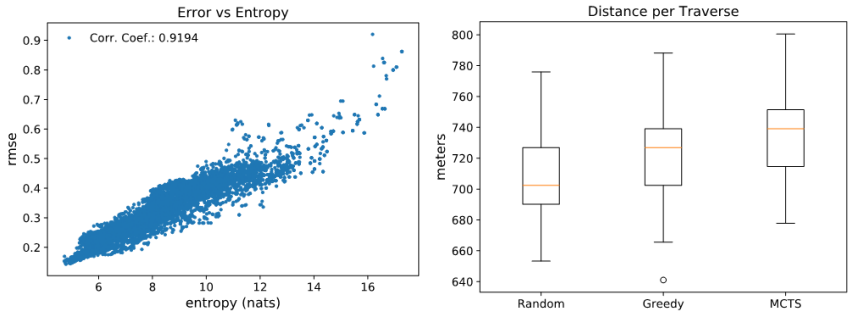


(g) P-Values for RMSE against samples collected



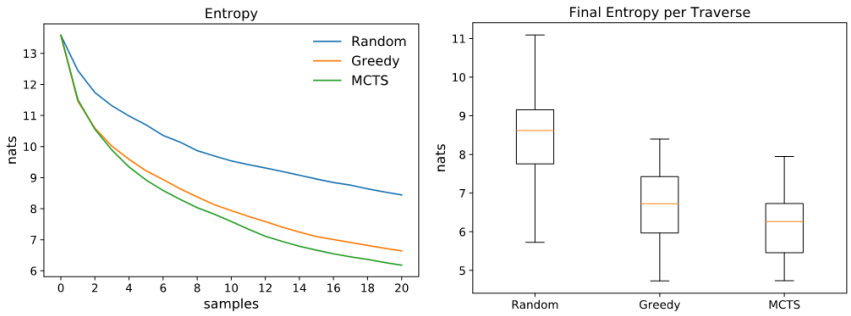
(h) P-Values for entropy against samples collected

Figure 4.9: Simulation results for Active Spectral Reconstruction at Test Site B.



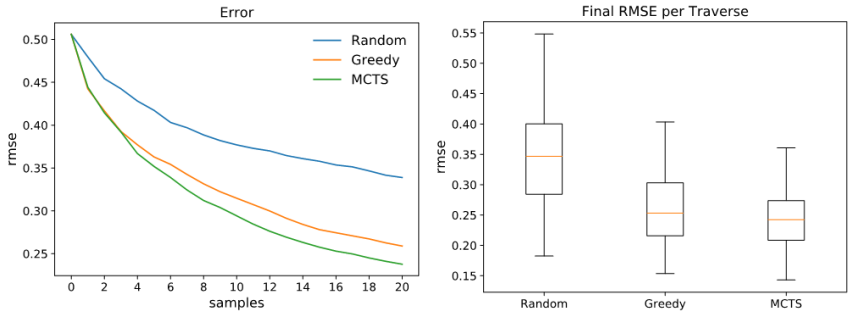
(a) Correlation of Entropy against RMSE

(b) Box plots comparing distance travelled for each planner



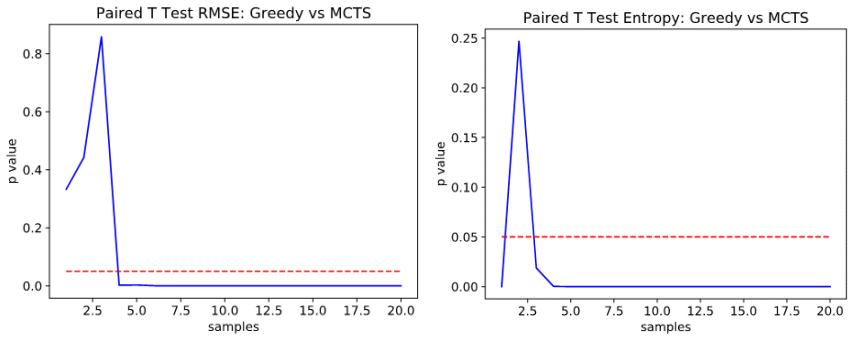
(c) Entropy against samples collected for each planner

(d) Box plots comparing Entropy for each planner



(e) RMSE against samples collected for each planner

(f) Box plots comparing RMSE for each planner



(g) P-Values for RMSE against samples collected

(h) P-Values for entropy against samples collected

Figure 4.10: Simulation results for Active Spectral Reconstruction at Test Site C.

the traversal budget remaining for future waypoints is insufficient. In fact, it may happen that GSS may utilize all of its budget for the first sampling location itself if the map is large enough. Thus, for traversing with small budget and a large map approaches such as NMPSE with fixed sampling distance is favourable.

For Active Spectral Reconstruction, we further constrained the sampling area by incorporating slope constraints. Regions that had slope greater than 18°were considered non-traversable for the rover. Comparing our planner with Random and Greedy planners demonstrated the superior performance of the MCTS approach for this experiment.

Chapter 5

Field Experiments

5.1 Experimental Setup

We begin with an overview of the prototype rover used for the experiments. Zoë is a solar powered rover with passive steering and passive suspension. It consists of an Advanced Spectral Devices (ASD) Fieldspec Pro VIS-NIR spectrometer with 18°resolution mounted on a pan-til unit. The rover also has two stereo camera for depth perception that aids the local navigator on-board the rover. The rover consists of a central processing unit named System Executive that handles all instrument control and drive-arc generation. A picture of Zoë is shown in Figure 5.1. The navigation is divided into two portions: a local navigator that performs obstacle avoidance and go-to-goal functions and a global navigator that provides the next goal location to the local navigator in the form of GPS coordinates. The global navigator uses the informed and uninformed planners discussed in this thesis.

The entire planning process follows a set of specific sequences. First, the system executive queries the global navigator for the next waypoint. Once it receives the GPS coordinates of the waypoint, the local navigator is commanded to reach that location. Upon reaching the desired location, the VIS-NIR spectrometer is commanded to sample a predefined number of spectra in a grid pattern. This is conducted to represent the AVIRIS-NG measurements as closely as possible. Once the spectra are sampled, they are updated in the model maintained by the global navigator and the process repeats.

Figure 5.1: Prototype Rover Zoë sampling spectrum at a location in Cuprite, NV

5.2 Results

The experiments are conducted at the same sites as in the simulation experiments described in Section 4.3. A key difference however, is the planner now also consists of traversable and untraversable regions on account of the physical constraints. As Zoë is only capable of climbing slopes less than 18°, actions that query samples from locations with a higher slope are deemed invalid. It is worthwhile to further analyze this additional constraint. Compared to other planners, MCTS will be able to better accommodate the constraints in its planning strategy as the multi-step lookahead will prevent it from navigating into local minimas: locations with no route ahead.

The results presented here will also be published in a manuscript at a future date [7]. For analysis purposes, we compared the spectra acquired by Zoë's spectrometer with AVIRIS-NG spectra and the predictions of the GP model. Figure 5.2 displays the spectra taken from Sites A,B,C. As is visible, the spectra from all three modalities closely match, provided ASD spectrometer spectra do not contain significant noise. Noise in the measurement was heavily reliant on the environmental conditions such as the intensity of incident radiation and cloud cover during the capture of the individual

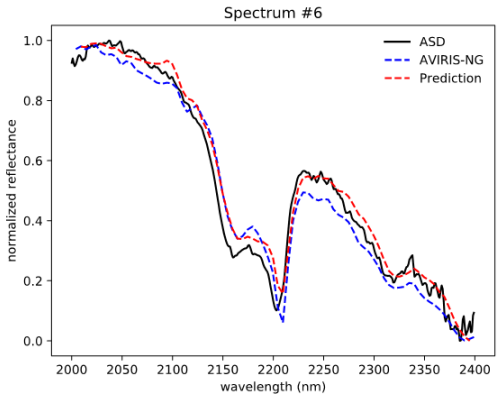
spectra. However, the noise deviation did not significantly affect the GP model. On account of the similarity between ASD spectrometer spectra and AVIRIS-NG spectra, it a viable to use AVIRIS-NG spectra wherever the ASD spectra are unavailable such as for computing the RMSE of the GP predictions at unvisited locations.

The entropy and RMSE of traverse taken at test site A is displayed in Figures 5.3a, 5.3b respectively. MCTS performs better than Greedy and Random planner with both entropy and RMSE being lower than the latter planners. The path of the traverse is depicted in Figure 5.3c. MCTS and greedy closely follow each other but the MCTS path has fewer divergences, conserving its samples for exploring the Northern region.

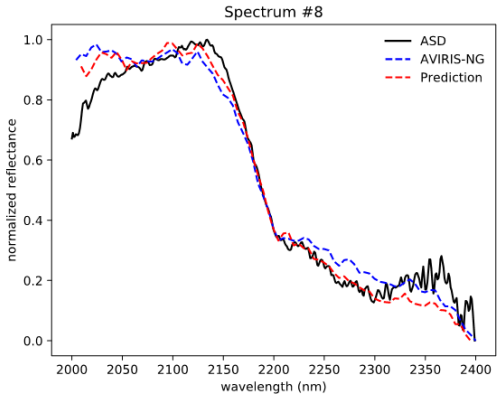
Similarly, the entropy, RMSE and path traversed for test site B is displayed in Figures 5.5a, 5.5b, 5.4c respectively. Once again, MCTS turned out to be the best planner amongst all three. Finally the results are shown for test site C in 5.5. The path of MCTS deviates significantly from the Greedy planner, sampling regions with significantly diverse spectra as is visible in Figure 5.5c.

5.3 Summary

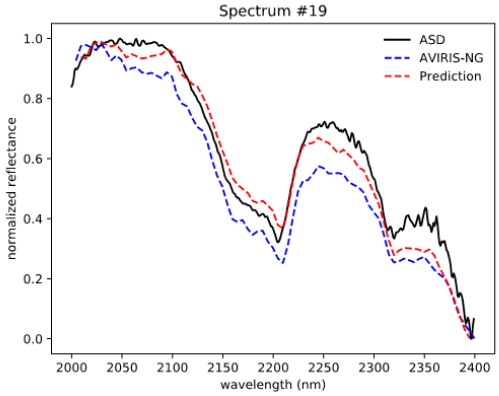
The field results corroborated the results observed in simulation. Our planner achieved lower reconstruction error compared to the baseline planners Random and Greedy. The entropy of the models was seen to be directly correlated with the reconstruction error, signifying that using entropy as a reward is a viable option. Moreover, the spectral measurements from AVIRIS-NG matched well with measurements from FieldSpec ASD spectrometer on-board Zoë. This allowed us to compute reconstruction error for the GP model using AVIRIS-NG measurements as a proxy.



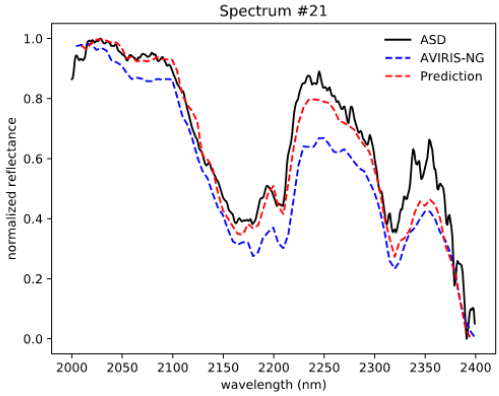
(a) Spectrum collected at Site A.



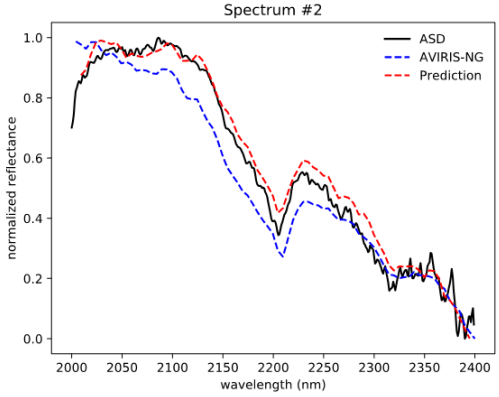
(b) Spectrum collected at Site A



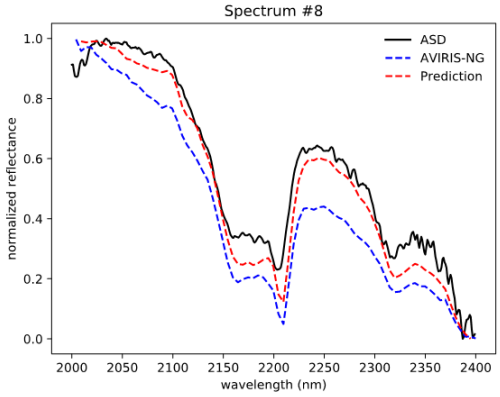
(c) Spectrum collected at Site B



(d) Spectrum collected at Site B

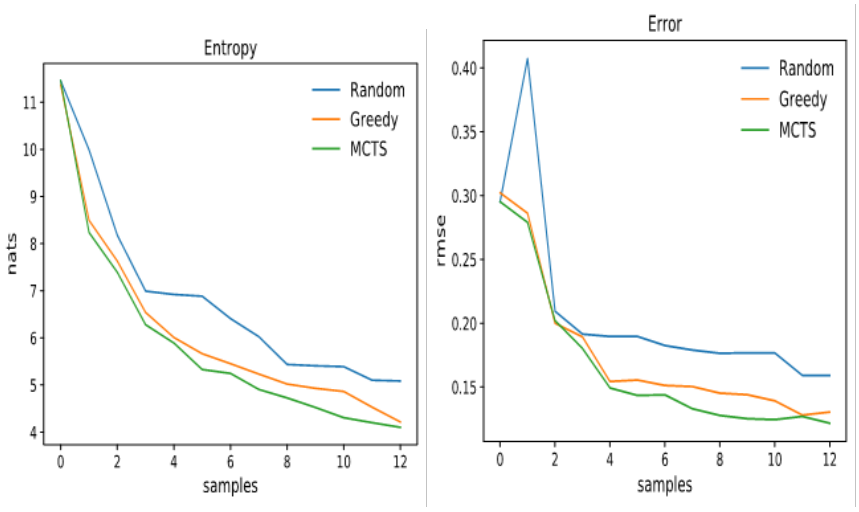


(e) Spectrum collected at Site C

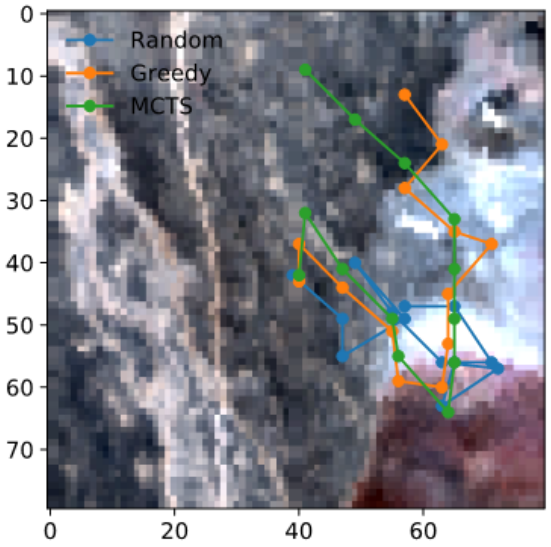


(f) Spectrum collected at Site C

Figure 5.2: Spectral measurements from ASD spectrometer, AVIRIS-NG and GP prediction from three test sites.

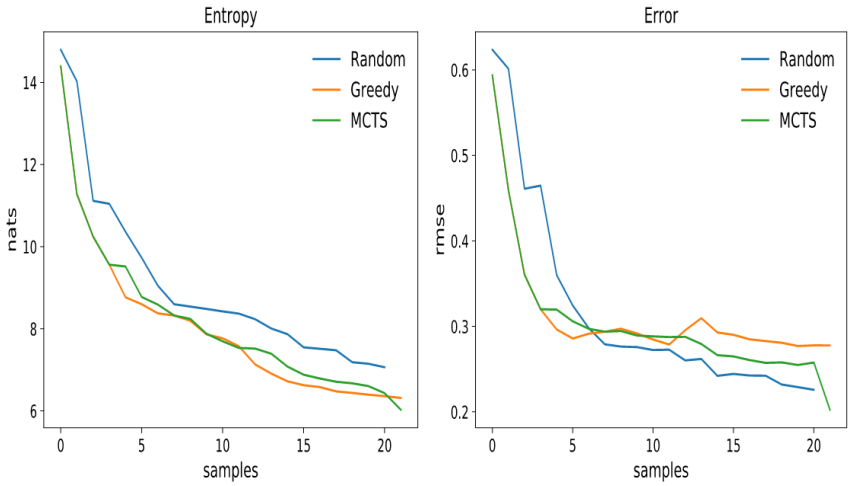


(a) Entropy against samples collected(b) RMSE against samples collected.

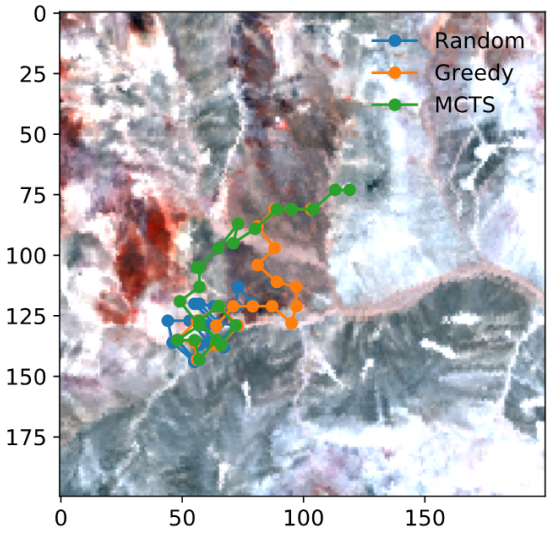


(c) Path traversed by planners.

Figure 5.3: Results for Test Site A.

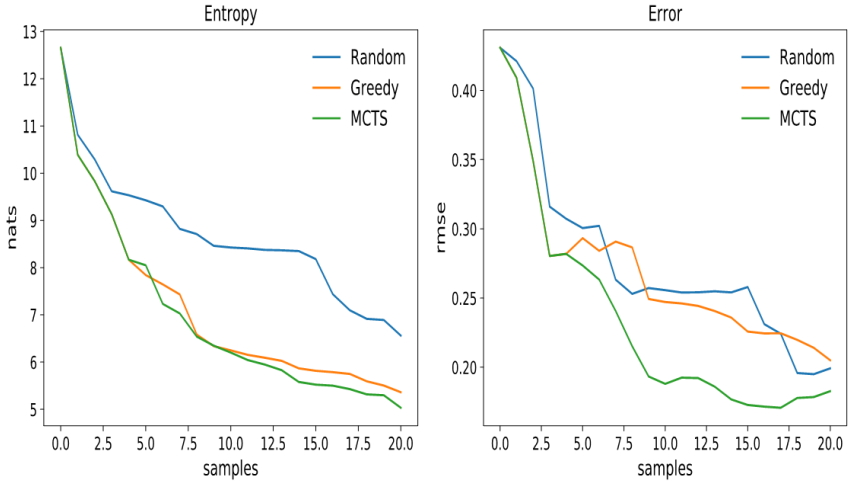


(a) Entropy against samples collected(b) RMSE against samples collected.

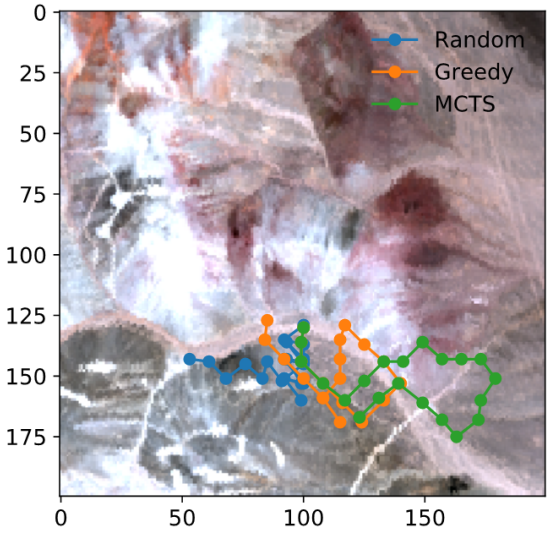


(c) Path traversed by planners.

Figure 5.4: Results for Test Site A.



(a) Entropy against samples collected(b) RMSE against samples collected.



(c) Path traversed by planners.

Figure 5.5: Results for Test Site C.

Chapter 6

Conclusion

6.1 Summary of Work

Path planning has been studied to a limited extent for science autonomy applications. Traditional planners either relied on naive strategies such as randomly sampling the next waypoint or did not fully utilize the sensing information for efficient planning.

This thesis presented MCTS as a viable planning method for SSE and Active Spectral Reconstruction. For SSE, over 50 tests were conducted in simulation verifying the efficacy of NMPSE: the MCTS variant for the *spectral unmixing* problem. The tests show that NMPSE provided significant improvement over the uninformed planners Random and Fixed-Step. Moreover, NMPSE empirically proved to possess lower RMSE compared to the state-of-art planner GSS.

In the case of Active Spectral Reconstruction, over 100 simulations were conducted at three different test sites using AVIRIS-NG as a proxy for *in situ* spectra. The tests proved that MCTS gave statistically significant improvement over Greedy and Random planners once sufficient samples were collected.

The results of Active Spectral Reconstruction were validated in-field at the Cuprite mining district in Nevada. In all three traverses, MCTS proved better than Greedy and Random search strategies.

6.2 Contributions

This thesis advances the field of planetary exploration in the following ways:

- We developed novel MDP formulations for the problem of SSE and Active Spectral Reconstruction that utilized orbital measurements to overcome the uncertainty in the stat-space.
- Designed a simulation environment for testing planners for these problems.
- Developed MCTS based planners that provide improvement over current planners with statistical significance.
- Designed software for rover Zoë using ROS framework for integrating the planners with its instruments.
- Field tested the planners at a mining district in Cuprite, NV proving practical feasibility of the planners.

6.3 Future Work

This research set up foundational structures for employing RL based methods for science autonomy.

Moreover, the field experiments provided a key question which needs to be addressed: how to combine local navigation with science-aware planning for complete autonomy. As mentioned in Section 4.3, slope constraints were determined using DEM acquired by ASTER. However, much like its spectral measurements, these DEM suffer from poor spatial resolution. As a consequence, sharp pits and small hills tend to get overlooked in the planning process. One way to address this is maintain a generative model of the slope which is updated as the rover traverses over the region, similar to the works of Berkenkamp et al. [41], [14].

Another avenue of future work is proposing improved tree search in MCTS. For instance, using the submodular property of the objective functions, it is possible to incorporate Branch and Bound [5] methodology for pruning sub-optimal nodes.

Bibliography

[1] N. A. Ahmed and D. V. Gokhale. Entropy expressions and their estimators for multivariate distributions. *IEEE Transactions on Information Theory*, 35(3): 688–692, May 1989. ISSN 0018-9448. doi: 10.1109/18.30996. 3.3.3

[2] A. Arora, R. Fitch, and S. Sukkarieh. An approach to autonomous science by modeling geological knowledge in a bayesian framework. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3803–3810, Sep. 2017. doi: 10.1109/IROS.2017.8206230. 3.1

[3] Richard Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966. 2.2.1

[4] Dimitri P. Bertsekas. *Dynamic Programming: Deterministic and Stochastic Models*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1987. ISBN 0132215810. 2.2.1

[5] J. Binney and G. S. Sukhatme. Branch and bound for informative path planning. In *2012 IEEE International Conference on Robotics and Automation*, pages 2147–2154, May 2012. doi: 10.1109/ICRA.2012.6224902. 6.3

[6] Jonathan Binney and Gaurav S Sukhatme. Branch and bound for informative path planning. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 2147–2154. IEEE, 2012. 3.1

[7] Alberto Candela, Suhit Kodgule, and Wettergreen David. Adaptive sampling and spectral reconstruction for planetary rover exploration. Manuscript under preparation, 20xx. 4.3, 5.2

[8] Nannan Cao, Kian Hsiang Low, and John M. Dolan. Multi-robot informative path planning for active sensing of environmental phenomena: A tale of two algorithms. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems*, AAMAS '13, pages 7–14, Richland, SC, 2013. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-1-4503-1993-5. URL `http://dl.acm.org/citation.cfm?id=2484920.2484926`. 3.1

[9] W CARRIER, III. Soviet rover systems. In *Space Programs and Technologies Conference*, page 1487. 1

[10] Guillaume Chaslot, Sander Bakkes, Istvan Szita, and Pieter Spronck. Monte-carlo tree search: A new framework for game ai. 2008. 2.2.3

[11] Doo-Hyun Cho, Jung-Su Ha, Sujin Lee, Sunghyun Moon, and Han-Lim Choi. Informative path planning and mapping with multiple uavs in wind fields. In *Distributed Autonomous Robotic Systems*, pages 269–283. Springer, 2018. 3.1

[12] Patrick Clary, Pedro Morais, Alan Fern, and Jonathan Hurst. Monte-carlo planning for agile legged locomotion. In *International Conference on Automated Planning and Scheduling*, 2018. 3.1

[13] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. A Wiley-Interscience publication. Wiley, 2006. ISBN 9780471748816. URL https://books.google.com/books?id=EuhBluW31hsC. 3.3.3

[14] Rikky RPR Duivenvoorden, Felix Berkenkamp, Nicolas Carion, Andreas Krause, and Angela P Schoellig. Constrained bayesian optimization with particle swarms for safe adaptive controller tuning. *IFAC-PapersOnLine*, 50(1):11800–11807, 2017. 6.3

[15] Greydon Foil. *Efficiently Sampling from Underlying Physical Models*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, October 2016. 3.1

[16] Hiroyuki Fujisada. Design and performance of aster instrument. In *Proc. SPIE*, volume 2583, pages 2583 – 2583 – 10, 1995. doi: 10.1117/12.228565. URL https://doi.org/10.1117/12.228565. 4.1

[17] P. Michael Furlong. *Foraging, Prospecting, and Falsification – Improving Three Aspects of Autonomous Science*. PhD thesis, Pittsburgh, PA, May 2018. 3.1

[18] Bo-Cai Gao, Alexander FH Goetz, and JA Zamudio. Removing atmospheric effects from aviris data for surface reflectance retrievals. 1991. 4.1.1

[19] Emilio Garcia-Fidalgo and Alberto Ortiz. Vision-based topological mapping and localization methods: A survey. *Robotics and Autonomous Systems*, 64:1–20, 2015. 3.1

[20] S. Gautam, B. S. Roy, A. Candela, and D. Wettergreen. Science-aware exploration using entropy-based planning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3819–3825, Sept 2017. doi: 10.1109/IROS.2017.8206232. 3.3.3

[21] Robert O Green, Michael L Eastwood, Charles M Sarture, Thomas G Chrien, Mikael Aronsson, Bruce J Chippendale, Jessica A Faust, Betina E Pavri, Christopher J Chovit, Manuel Solis, Martin R Olah, and Orlesa Williams. Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (aviris). *Remote Sensing of Environment*, 65(3):227 – 248, 1998. ISSN 0034-4257. doi: https://doi.org/10.1016/S0034-4257(98)00064-9. URL http:

//www.sciencedirect.com/science/article/pii/S0034425798000649. 4.1, 4.1.1

[22] John P. Grotzinger, Joy Crisp, Ashwin R. Vasavada, Robert C. Anderson, Charles J. Baker, Robert Barry, David F. Blake, Pamela Conrad, Kenneth S. Edgett, Bobak Ferdowski, Ralf Gellert, John B. Gilbert, Matt Golombek, Javier Gómez-Elvira, Donald M. Hassler, Louise Jandura, Maxim Litvak, Paul Mahaffy, Justin Maki, Michael Meyer, Michael C. Malin, Igor Mitrofanov, John J. Simmonds, David Vaniman, Richard V. Welch, and Roger C. Wiens. Mars science laboratory mission and science investigation. *Space Science Reviews*, 170 (1):5–56, Sep 2012. ISSN 1572-9672. doi: 10.1007/s11214-012-9892-2. URL https://doi.org/10.1007/s11214-012-9892-2. 1

[23] L. Hamlin, R. O. Green, P. Mouroulis, M. Eastwood, D. Wilson, M. Dudik, and C. Paine. Imaging spectrometer science measurements for terrestrial ecology: Aviris and new developments. In *2011 Aerospace Conference*, pages 1–7, March 2011. doi: 10.1109/AERO.2011.5747395. 4.1

[24] Daniel Hennes and Dario Izzo. Interplanetary trajectory planning with monte carlo tree search. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015. 3.1

[25] Geoffrey A. Hollinger and Gaurav S. Sukhatme. Sampling-based robotic information gathering algorithms. *The International Journal of Robotics Research*, 33(9):1271–1287, 2014. doi: 10.1177/0278364914533443. URL https://doi.org/10.1177/0278364914533443. 1

[26] Ronald A Howard. Dynamic programming and markov processes. 1960. 2.2.2

[27] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996. 1

[28] Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9(Feb):235–284, 2008. 3.1

[29] Fred A. Kruse and Sandra L. Perry. Regional mineral mapping by extending hyperspectral signatures using multispectral data. In *IEEE Aerospace Conference Proceedings*, pages 1–14, 2007. ISBN 1424405254. doi: 10.1109/AERO.2007.353059. 4.1.2

[30] Charles L Lawson, 1938 Hanson, Richard J., Society for Industrial, and Applied Mathematics. *Solving least squares problems*. Philadelphia : SIAM, [rev. ed.] edition, 1995. ISBN 0898713560 (pbk.). URL http://www.zentralblattmath.org/zmath/en/search/?an=0860.65029. "This SIAM edition is an unabridged, revised republication of the work first published by Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1974"–T.p. verso. 3.2.1

[31] Zhan Wei Lim, David Hsu, and Wee Sun Lee. Adaptive informative path planning in metric spaces. *The International Journal of Robotics Research*, 35 (5):585–598, 2016. doi: 10.1177/0278364915596378. URL https://doi.org/10.1177/0278364915596378. 3.1

[32] Roman Marchant and Fabio Ramos. Bayesian optimisation for informative continuous path planning. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6136–6143. IEEE, 2014. 3.1

[33] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015. 2.3

[34] Philippe Morere, Roman Marchant, and Fabio Ramos. Sequential bayesian optimization as a pomdp for environment monitoring with uavs. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 6381–6388. IEEE, 2017. 3.1

[35] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003. 3.4.1

[36] DARA Roberts, ROBERTO Green, DONALDE Sabol, and JOHNB Adams. Temporal changes in endmember abundances, liquid water and water vapor over vegetation at jasper ridge. 1993. 4.1.1

[37] Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS*, 2011. 3.1

[38] Juanita C Sandidge and Ronald J Holyer. Coastal bathymetry from hyperspectral observations of water radiance. *Remote Sensing of Environment*, 65(3):341–352, 1998. 4.1.1

[39] Michael C Shewry and Henry P Wynn. Maximum entropy sampling. *Journal of applied statistics*, 14(2):165–170, 1987. 3.1

[40] Geoffrey M. Smith and Edward J. Milton. The use of the empirical line method to calibrate remotely sensed data to reflectance. *International Journal of Remote Sensing*, 20(13):2653–2662, 1999. doi: 10.1080/014311699211994. URL https://doi.org/10.1080/014311699211994. 4.1.2

[41] Yanan Sui, Alkis Gotovos, Joel Burdick, and Andreas Krause. Safe exploration for optimization with gaussian processes. In *International Conference on Machine Learning*, pages 997–1005, 2015. 6.3

[42] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018. 1, 2

[43] Gregg Swayze, Roger N Clark, Fred Kruse, Steve Sutley, and Andrea Gallagher. Ground-truthing aviris mineral mapping at cuprite, nevada. 1992. 4.1.1

[44] Gregg A. Swayze, Roger N. Clark, Alexander F.H. Goetz, K. Eric Livo, George N. Breit, Fred A. Kruse, Stephen J. Sutley, Lawrence W. Snee, Heather A. Lowers, James L. Post, Roger E. Stoffregen, and Roger P. Ashley. Mapping advanced argillic alteration at cuprite, nevada, using imaging spectroscopy. *Economic Geology*, 109(5):1179, 2014. doi: 10.2113/econgeo.109.5.1179. URL http://dx.doi.org/10.2113/econgeo.109.5.1179. 4.1.2

[45] David R. Thompson. *Intelligent Mapping for Autonomous Robotic Survey*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, August 2008. 3.1, 3.3.3

[46] David R. Thompson, David Wettergreen, Greydon Foil, Padraig Michael Furlong, and Anatha Ravi Kiran. Spatio-spectral exploration combining in situ and remote measurements. In *Association for the Advancement of Artificial Intelligence*, January 2015. 3.2.2, 4.1.2, 4.2.1

[47] David R. Thompson, Alberto Candela, David S. Wettergreen, Eldar Noe Dobrea, Gregg A Swayze, Roger N. Clark, and Rebecca Greenberger. Spatial Spectroscopic Models for Remote Exploration. *Astrobiology*, 18(7):934–954, 2018. ISSN 0261-2194. doi: 10.1089/ast.2017.1782. 4.1.2

[48] David R. Thompson, Vijay Natraj, Robert O. Green, Mark C. Helmlinger, Bo-Cai Gao, and Michael L. Eastwood. Optimal estimation for imaging spectrometer atmospheric correction. *Remote Sensing of Environment*, 216(May):355–373, 2018. ISSN 00344257. doi: 10.1016/j.rse.2018.07.003. URL https://doi.org/10.1016/j.rse.2018.07.003. 4.1.1

[49] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8 (3-4):279–292, 1992. 2.2.4

[50] David Wettergreen, Nathalie Cabrol, Vijayakumar Baskaran, Francisco Calderón, Stuart Heys, Dominic Jonak, R Allan Luders, David Pane, Trey Smith, James Teza, et al. Second experiments in the robotic investigation of life in the atacama desert of chile. In *Proc. 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2005. 4.2.4