Assessment of an Online RF Propagation Hybrid Architecture for Communication-Aware small Unmanned Aircraft Systems

by

Spencer G. Watza

B.S., Western Michigan University, Kalamazoo, 2015

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Master of Science

Ann & H.J. Smead Department of Aerospace Engineering Sciences

2018

This thesis entitled:
Assessment of an Online RF Propagation Hybrid Architecture for Communication-Aware small Unmanned Aircraft Systems
written by Spencer G. Watza
has been approved for the Ann & H.J. Smead Department of Aerospace Engineering Sciences

_____
Prof. Eric W. Frew

_____
Prof. Nisar Ahmed

_____
Prof. Tam Vu

Date _____

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Watza, Spencer G. (M.S.,Aerospace Engineering Sciences)

Assessment of an Online RF Propagation Hybrid Architecture for Communication-Aware small Unmanned Aircraft Systems

Thesis directed by Prof. Eric W. Frew

Small Unmanned Aircraft Systems (sUAS) are attracting significant attention for their use in a wide range applications. These applications can be categorized into two types based on their communication objective; communication-focused or communication-enabling. In both types it is beneficial to completing its mission if the sUAS is communicational-aware or it has information to assess the performance of a given communication channel. In the literature there is a lack of adaptable, robust, and online methods to provide the necessary information to be communication-aware. This thesis expands on a few authors' work to formally define and assess an online hybrid architecture that is also adaptable and robust. The architecture has a Bayesian approach combining an a-priori RF propagation model with a machine learning correction tool to provide an initial estimate and learn the deviation of that a-priori model to provide a combined prediction at any given point. The hybrid architecture is implemented and a series of assessments using simulation and flight data are performed on its capabilities and performance. The results from these assessments are that the online hybrid architecture provides a benefit over learning the field directly and when applied to a wireless airborne relaying application can perform as well as naive approaches.

Acknowledgements

I would like to acknowledge the graduate students in the System Integration Lab at CU Boulder that made my experience at CU enjoyable and fun. In addition, their help was instrumental in refining the concepts and ideas presented in this work. I would also like to thank the IRISS team for the flight and engineering support that was provided throughout this work. Without their help it would have been significantly more difficult to collect the data I needed. I would like to thank my advisor, Eric Frew, for being supportive and helping me to complete this work. Lastly, I would like to thank my family for the support they have given me that has helped me to this stage.

Contents

Tables

Table

Figures

Figure

Chapter 1

Introduction

1.1 Motivation

The unmanned aircraft system (UAS) industry has grown exponentially in popularity and size in the past 8 years. The financial impact is expected to triple by 2025[1]. The unmanned aircraft (UA) industry has shifted from the military sector to commercialization in part from the new potential use cases which have the potential to reduce risk and lower costs. Drone delivery, agricultural monitoring, atmospheric sensing, and mobile communication networks are several examples. At the University of Colorado Boulder (CU Boulder), drones are deployed in a variety of topics including archaeological mapping, wildlife monitoring, and atmospheric sensing. In all of these example missions, communication is critical for the safe operation of the vehicle and to provide real-time data to the users on the ground.

UAS operations can be categorized into two groups relating to communication; *communication-focused* and *communication-enabling*. The objective of communication-focused missions is to create a wireless communication link between one or more assets/users and the UAS. Two examples are creating a mobile ad-hoc network of UA to supplement cellular towers during a crisis where local infrastructure is damaged or using a UAS to provide data links between a firefighter team during a wildfire in remote regions. The second category, communication-enabling, is a mission where the performance could be improved if a wireless communication link is established between one or more assets/users and the UAS. In such operations improved could refer to increased efficiency, reduced time, or more information collected. An example of this type of mission is search and rescue where

the main objective is to find and recover missing individuals. However by using UAS to aid in the search by sharing its data to the other team members it is plausible that the mission could be successfully completed faster than without shared communication.

In order for these missions to be successful using autonomous vehicles, the vehicles need to be communication-aware:

> *Communication-awareness* is the ability to measure or derive spatial-temporal information about communication channel performance and use this information to make decisions in order to maintain or improve the channel throughout the mission.

Modeling wireless radio-frequency (RF) communication channels is complex. Wireless communication in the RF spectrum relies on propagating electromagnetic signals through a medium and receiving them at another location. Antennas act as the "mouth" and "ears" for this process as they convert electronic signals into electromagnetic waves and vice versa. The complexity is due to the way the electromagnetic signals propagate. The signal radiates in all directions, diffracting, scattering, and reflecting off particles in the environment causing attenuation. The waves that are part of the signal source may take a variety of paths to arrive at the receiver antenna each with various phase shifts and power attenuation based on the path taken. Each path that the wave travels is environmentally dependent, which includes the effects of external sources of radiation interfering with the signal's propagation. This radiation is generated from the natural state of many objects in the environment. This effect makes modeling every interaction impossible on the desired generated signal. While the number of interactions can be mitigated by assuming that a lot of the objects have negligible effects, it is still difficult to model the large effects caused by objects on Earth. Accounting for terrain may introduce new errors that come from the sensors, information that the model lacks such as vegetation and material composition, and may have physical errors due to the changing landscape. City models might make assumptions about the shapes and material compositions of the buildings as another example. These effects all introduce new errors into the RF propagation model.

1.2 Literature Review

Researchers have been investigating ways to make robotic systems communication-aware for over a decade [2, 3, 4].Recently communication centric applications have become a focal point for research in UAS [5, 6, 7, 8], however there is a lack in developing systems to provide the communication-awareness required for these applications. A lot of work has been focused on mobile ad-hoc networks (MANET), flying ad-hoc networks (FANETS), vehicle ad-hoc networks (VANETS) and sensor networks [9, 10, 11, 12, 13, 14, 15, 16]. One issue with this work is that they work on positioning algorithms, trajectories, routing, and energy efficiency among other things assuming that some approach will exist to provide communication performance information or that they rely on simple empirical or mathematical models that describe RF propagation. Work that is more closely linked with developing a strategy to become communication-aware rely on empirical models [17] or control optimization techniques around measurements [18, 19, 20]. A problem faced in these control optimization approaches is that they also use empirical models to demonstrate their robustness.

The problem with these approaches is that the empirical models used are not accurate or precise models for RF propagation [21] with inconsistent errors over 10s of dBs. These models significantly break down when faced with complex environments like flying within a city or in mountainous regions [22, 23, 24, 25, 26, 27]. The communication modeling literature has been working on developing new methods to predict RF path loss [26] but they still have limitations computationally or with error performance. The assumption that simplistic models are good representations of what the signal attenuation might be does not hold for a large portion of locations where UAS communication applications will occur [22, 7, 6, 13].

Some authors [28, 29, 30, 31, 32] have investigated other methods specifically to provide communication-aware information for a given application and have tested them [29, 28, 33, 34, 35]. In these works, the methods are application-specific such as using behavior of an aircraft to do gradient descent or creating a line of sight map for planning. There has been work to

generate more generic models but have relied on constructing RF propagation models for air to ground communication [36, 37, 38]. However in these works, they have not verified their models with experimental data. Looking at the work from the ground robotics literature there has been significant additions in mapping RF fields [39, 2, 40, 41]. Wagle [42] brought over the idea of learning the RF path loss using it to learn an empirical model offline. Carfang [43] furthered the work by instead learning a deviation from an empirical model.

In summary, the literature for creating communication-aware UAS is lacking a solution in terms of being robust and adaptable. Online solutions have been explored although they are application specific utilizing specific properties of the mission to learn and extract relevant information. In addition being able to quickly field the system is of importance for a large range of applications.

1.3 Problem Statement

The problem addressed in this thesis is the design and assessment of an architecture for learning the radio frequency environment online and is robust and adaptable. The main challenges are making the architecture robust for a variety of situations and is non application specific. Robustness requires that the algorithm is not heavily influenced by noise and can adjust to new measurements that may not fit the original prediction. Making the same robust system to be quick in deployment to ensure practicality is crucial but learning algorithms that would allow for robustness require sufficient data samples and time. Lastly, to make the architecture be versatile/adaptable for many applications a direct quantification of the communication channel is necessary.

1.4 Solution

The solution presented here is a hybrid online architecture that combines traditional (*a priori*) RF modeling methods with an active online learning tool to predict the expected RF path loss at given locations. The architecture (Figure 1.1) is designed to learn deviations in the traditional RF model with the correction tool. Adding this learned deviation to the the initial prediction from the traditional RF model creates the combined prediction at any given point.

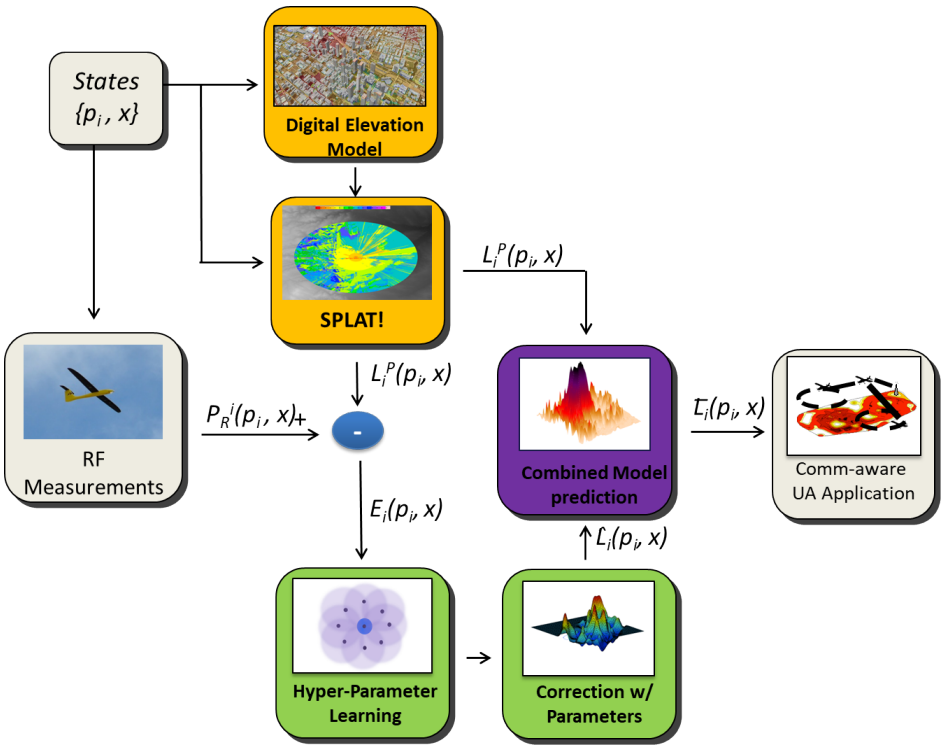

Figure 1.1: Diagram depicting the architecture form for online measurements which includes combining an *a-priori* model (e.g. SPLAT!) and a correction model (e.g. Gaussian process) to provide a prediction of the path loss

There are several advantages to this methodology. First, by abstracting the specific details of the a priori and correction model implementations, the architecture can be updated with newer methods and learning tools as those fields progress. Second, by utilizing a RF model to provide an initial estimate of path loss, planning systems/algorithms can immediately start running without needing to wait for data collection. This provides base level exploitation whereas it would normally be assumed that the UAS would first need to map the environment before flying the operation, requiring additional time. Third, by using a correction tool to learn the deviation of the RF model the system collects information that can be later used in simulation and for other work characterizing propagation models.

1.5 Contributions

This work contributes to the field by assessing the hybrid online architectures' performance. This is achieved through implementation of the architecture in software, performing a simulation assessment of the architecture's learning capability, simulating a use case of the architecture for airborne relaying and evaluating the components of the architecture with flight data. The main contributions include:

1. Hybrid Architecture for predicting RF propagation losses.
    - Implemented the SPLAT! RF model in order to provide an initial estimate of path loss.
    - Formulated and implemented a Gaussian process to learn corrections of the SPLAT! RF model.
    - Performed an assessment study on the Gaussian process's ability to learn model deviations.
2. Airborne Wireless Relay Simulations.
    - Designed a simulation experiment to evaluate the architecture when applied to a relay application.
    - Compared the performance of the architecture against a naive approach.
    - Analyzed the influence of different types of a-priori models on the architecture's performance.
3. Experimental Field Testing and Flight Data.
    - Implemented and setup fixed wing aircraft to collect experimental RF data
    - Assessed the accuracy and precision of SPLAT! as a propagation model
    - Evaluated various Gaussian process models for their prediction behaviors

Chapter 2

Online Hybrid Architecture Implementation

The hybrid architecture provides a methodology to predict the path loss to a given location from a known source. Multiple models, using this hybrid architecture, are constructed to provide predictions for a set of signal source locations. The hybrid architecture is composed of two major parts, the a-priori model and the correction model. The a-priori model provides an initial estimate of the path loss expected at a point given the transmitters location and environment information. The correction model learns the deviation of the a-priori model from measured path loss in operation. By combining these two model outputs into the combined model, a prediction of the path loss can be provided at any given location from a transmitter. The architecture, shown again in Figure 2.1, highlights the corresponding sections with colors; yellow for the a-priori, green for the correction model, and purple for combined model.

2.1 A-Priori Model

The a-priori model chosen for the architecture is SPLAT! (Signal Propagation Loss and Terrain Analysis Tool), an open source implementation of the Longley-Rice model with modifications by the creators.[44]

SPLAT! has two methods of analysis: point to point and area coverage. For this work, the area coverage is of most interest because it allows for quick computation of the path loss at multiple locations. SPLAT! requires several different inputs. The first input is a digital elevation map (DEM) that provides terrain elevation. The authors of SPLAT! recommend to use data generated from

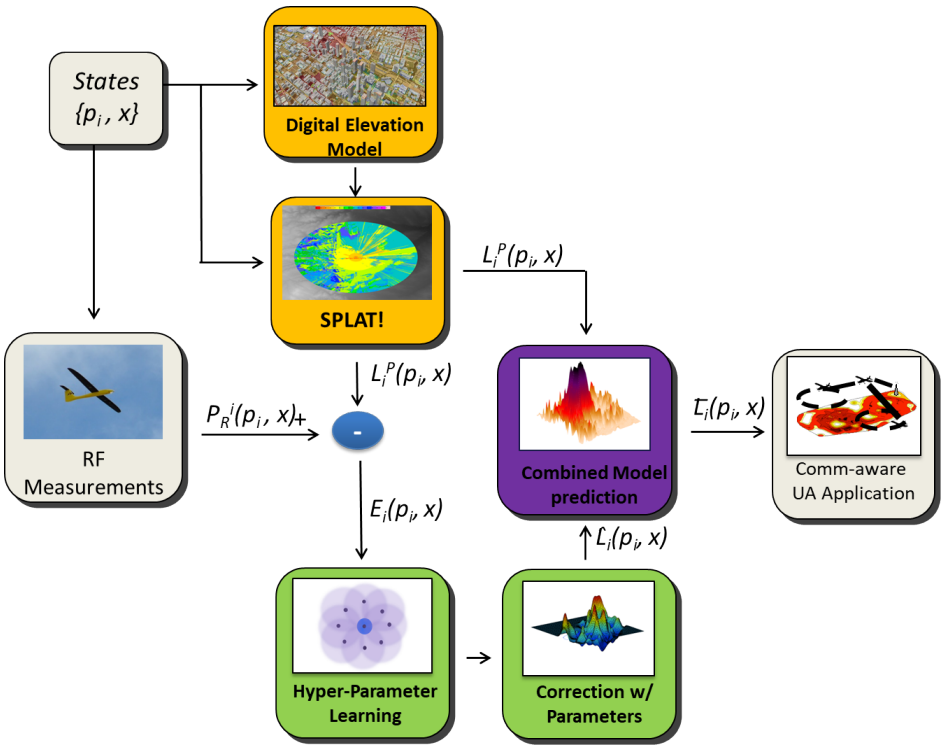

Figure 2.1: Diagram of Online Hybrid Architecture with color coding of components

the Shuttle Radar Topography Mission (SRTM), which is available through the USGS website and can provide up to 1 degree arc second resolution maps (approximately 30 meters). While higher resolution maps exist for the US from the USGS, SPLAT! has not been updated to handle that newer data. SPLAT! comes with tools that format the data from a variety of different elevation maps into the desired format for the algorithm. The next input is the transmitter location specified in a geographic coordinate system. In addition the name of the transmitter source and the antenna height above ground are required. This is all specified in a .qth file. Next, transmission propagation parameters should be specified (otherwise default values are assumed) to provide estimated values of certain coefficients in the algorithm. Lastly, a radius and receiver antenna height above ground needs to be specified when running.

The output of a single run of the SPLAT! software provides several variables over a discrete grid representing the receiver locations. The full set of output variables is an array that includes: the position of a receiver, the expected path loss at that position, the elevation angle, whether there

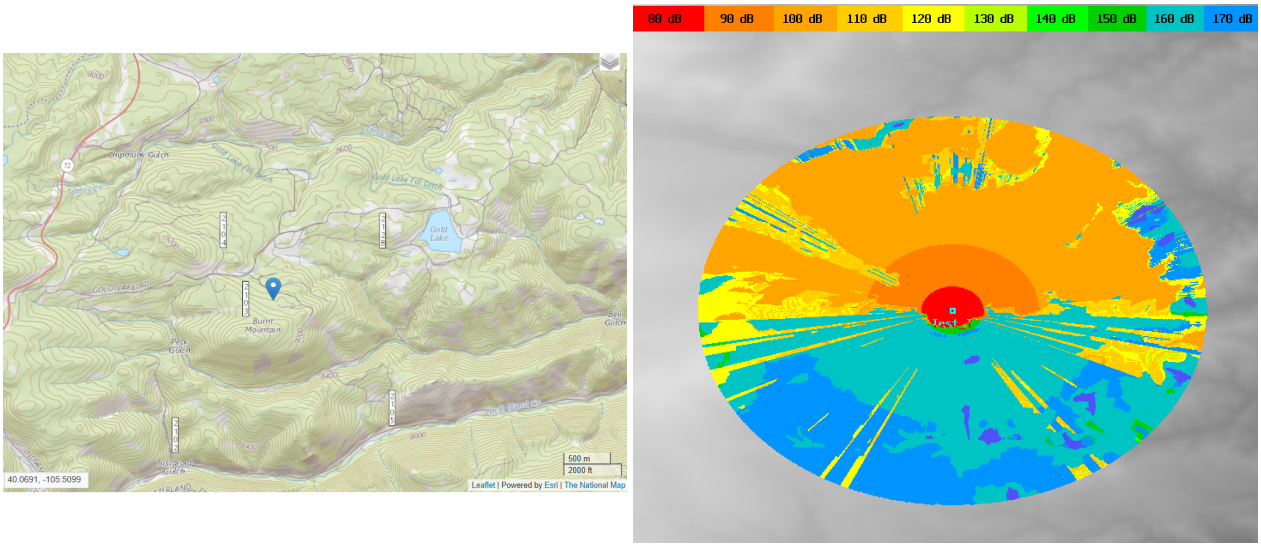

Figure 2.2: Two images for a SPLAT! output example. The left image is a topographical map from USGS depicting the transmitter location with a marker. The right image displays the path loss mapped to colors from SPLAT!

is line of sight between transmitter and receiver, and the azimuth angle. In addition the results can be plotted in a color map that depicts the path loss for all the receiver sites. Figure 2.2 depicts an example run with a transmitter placed just north of a mountain in Colorado (shown on the left topographical map) for a 5 kilometer radius with a receiver height 100 meters above the ground at 5.8 GHz.

The representation of height in SPLAT! poses a challenge for use for unmanned aircraft systems. Most aircraft fly at a constant altitude relative to mean sea level (MSL) whereas SPLAT! defines height above ground level (AGL). In situations where the terrain has only minor deviation we assume that the AGL prediction is sufficient in accuracy as the a-priori model, however in more varied terrain a more detailed approach must be used to generate an initial estimate field for MSL. One way to generate this is by creating a series of AGL fields from SPLAT! and using known terrain elevation to interpolate and rebuild a MSL field. While the accuracy of this generated MSL field is important for early predictions, the errors will be minimized regardless by the correction model.

2.2 Correction Model

The correction model is based on a machine-learning method called a Gaussian process (GP). A Gaussian process is a data driven, non-parametric approach for regression and classification. In this work, we use Gaussian process regression to learn a model of the a-priori deviation from measured signal strength. The Gaussian process regression is designed to run online and could potentially run onboard the aircraft as well. For simplicity in the regression, the inputs are defined as the two dimensional geographic position coordinates derived from flat Earth approximations. The output is the deviation in the a-priori path loss prediction. While the aircraft is flying, it is collecting received signal strength measurements and converting them into perceived path loss measurements and recording the position of these measurements. This set of measurements will be the training data to learn the GP model. Though an advantage of the GP is that it can make predictions over a continuous domain, the test points that we want to predict will be the set of discrete SPLAT! output locations.

2.2.0.1 Gaussian Process Regression

Gaussian process regression consists of two main steps: learning and prediction. The goal of the learning step to determine a set of hyperparameters that define a covariance function (also known as the kernel function) by optimizing the log likelihood of the training data. After this optimization, using the learned hyperparameters and trained data points the GP can specify the mean and variance of a Gaussian probability distribution function of the output at a given input location. Thus, the GP can provide estimates of the deviated path loss value for various communication-aware applications. The following part provides a mathematical description of the relevant parts.

The Gaussian process is defined as

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x'})) \tag{2.1}$$

where $m(\mathbf{x})$ is the mean function and $k(\mathbf{x}, \mathbf{x'})$ is the covariance function of a real process $f(\mathbf{x})$. In

this work the covariance function is the *squared exponential*,

$$k = \sigma^2 \exp -\frac{(\mathbf{x}_p - \mathbf{x}_q)^2}{2l^2} \tag{2.2}$$

where $\mathbf{x}_p$ and $\mathbf{x}_q$ are two random variables (inputs) and l is the length scale. In the squared exponential kernel when two inputs are very close the covariance is almost unity, decreasing when the inputs become further apart in their space. The hyperparameters for the square exponential are l and σ where σ is the scale factor. The learning portion of the Gaussian process attempts to optimize the hyperparameters by maximizing the log marginal likelihood of

$$\log p(\mathbf{y}|\mathbf{X}, \theta) = -\frac{1}{2}\mathbf{y}^T(K + \sigma_n^2 I)^{-1} - \frac{1}{2}\log|K + \sigma_n^2 I| - \frac{n}{2}\log 2\pi \tag{2.3}$$

where $\mathbf{X}$ is the set of all inputs, $\mathbf{y} = f(\mathbf{x})$, θ is the hyperparameters, K is the covariance function, and σ_n^2 is the variance of white noise. The exact method for performing this calculation and the optimization is dependent on implementation. The Gaussian process implementation used in simulation was the *fitrgp* function from MATLAB (with default parameters). Once the hyper parameters are selected, $p(\mathbf{y}|\mathbf{X}, \theta)$ is calculated for any desired test points with

$$p(\mathbf{y}|\mathbf{X}, \theta) = \int p(\mathbf{y}|\mathbf{f}, \mathbf{X})p(\mathbf{f}|\mathbf{X})d\mathbf{f} \tag{2.4}$$

where $p(\mathbf{f}|\mathbf{X})$ and $p(\mathbf{y}|\mathbf{f}, \mathbf{X})$ come from the covariance matrix and training data used to create it. This is a summary of Gaussian processes from Rasmussen[45].

Chapter 3

Hybrid Architecture Simulation

This chapter provides insight into the performance of the hybrid architecture, specifically investigating the GP correction tool. Section 3.1 provides a discussion on the way truth data is generated for the simulations. This is followed by section 3.2 presenting setup and results of the simulation. For implementation details in MATLAB refer to A.

3.1 Nature Run

In order to have truth data for assessing the architecture, a *nature run* is derived by combing three components. The methodology for generating a *nature* needs to provide similar structure to the hybrid architecture in order to assess its capability. A *nature run* field consists of combining two fields; a baseline field and a disturbance field are added together to create a mean field. Noise is added to the mean field to simulate received signal strength measurements being taken becoming

$$L_{nature}(\mathbf{p}) = L_{base}(\mathbf{p}) + L_{dist}(\mathbf{p}) + v \tag{3.1}$$

where v is the random noise. This mimics the process of the hybrid architecture allowing for comparison of the disturbance model to the learned GP model, both representing the deviation from the a-priori model. L_{nature} is a 2-dimensional model and is expanded into 3-dimensions (when needed) by stacking multiple mean fields vertically. For simplicity, these mean fields are discretized using the same structure in the hybrid architecture creating a 3D grid. As the simulated aircraft

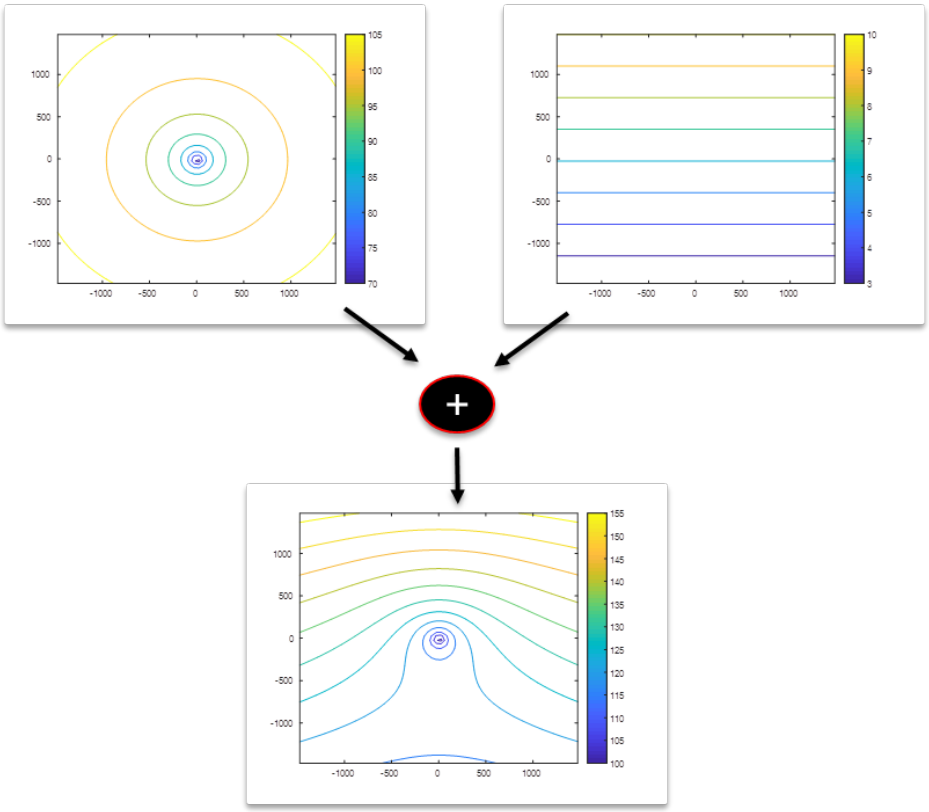

Figure 3.1: Diagram depicting the formation of the mean field in a nature run by combining a baseline model (upper Left) with a disturbance model (upper right).

flies in continuous states, a bounding algorithm defines which grid to sample from the *nature run*. A diagram illustrating the generation of the mean field is provided in Figure 3.1, where free space path loss is used as the baseline model (left) and is added with a linear varying disturbance model (right).

3.1.1 Baseline Model

The purpose of the baseline model is to provide shape to the mean path loss field based on mathematical or empirical models. This work features two different baseline models: free space path loss (FSPL) and SPLAT!. The FSPL model takes into account the distance R between the transmitter and the receiver and the wavelength λ in calculating the path loss at a point:

$$L_{freespace} = (\frac{\lambda}{4\pi R})^2. \tag{3.2}$$

The FSPL model is a simplistic model in literature and is often used as a basis for what can be considered naive approaches. The FSPL model is also a core component of most other models [26].

SPLAT! is used as a second baseline model which was discussed previously in Chapter 2. Unlike FSPL, SPLAT! accounts for terrain properties. It should be noted here that when the baseline model is the same as the a-priori model there is an inherent bias in the overall performance however it allows for analysis on the Gaussian process's ability to learn the deviation represented by the disturbance model.

3.1.2 Disturbance Model

Three main types of disturbance models are used. The goal of the different disturbance models is to provide different spatial correlation structure in the deviation from the a-priori model. The three disturbance models are normal, linear, and constant models. When creating data for the 3D map of the mean field, a disturbance model was generated for each altitude, with the same disturbance model was used for all altitudes.

3.1.2.1 Normal Model

The normal model is a cell independent geospatial process that assigns a path loss value from a normal distribution.

$$L_{normaldist}(\mathbf{p}) = \mathcal{N}(\mu, \sigma^2) \quad \forall \quad \mathbf{p} \tag{3.3}$$

where μ and σ^2 are the mean and variance of the distribution. This model assumes a worst case scenario as there are no correlations between neighboring cells except that they are drawn from the same distribution. The advantage of this model in simulation will be that it weakens the dependence of the overall performance on the a-priori model testing the correction model.

3.1.2.2 Linear Model

The linear model is a simple geospatial process that assigns a series of values across the cells in a linear fashion shown

$$L_{lineardist}(\mathbf{p}) = m\mathbf{p} + b \tag{3.4}$$

where m is the rate of change and b is the minimum offset. The linear model is often represented by a shift from some minimum value to a maximum value along an axis direction in the field (along X or along Y axis) using the number of grids to determine the rate.

3.1.2.3 Constant Model

The constant model is a non-geospatial process where we assign a single value to all possible grids for all heights shown

$$L_{constdist} = \mathcal{N}(\mu, \sigma^2) \tag{3.5}$$

here μ and σ are the mean and variance of the distribution.This single value is also drawn from a normal distribution. The primary use of this model is to evaluate the effects of the noise in sensor measurements in learning the baseline or disturbance model.

3.1.3 Multipath Noise

An important part in generating the simulated measurements is adding noise. So far if the aircraft was to sample the path loss at any given point it would be the mean field value created from the methodology just presented. The value would only change if the aircraft moved into a new grid cell which is not representative of nature. Instead it is expected that measurements taken at the exact same location will have some variation due to multipath and sensor error effects over time. In literature there are two common multipath noise models [46, 47, 48]; the Rician distribution and Rayleigh distribution. The Rician distribution is used when a strong line of sight component between the source and destination exists while the Rayleigh is used when there is no line of sight.

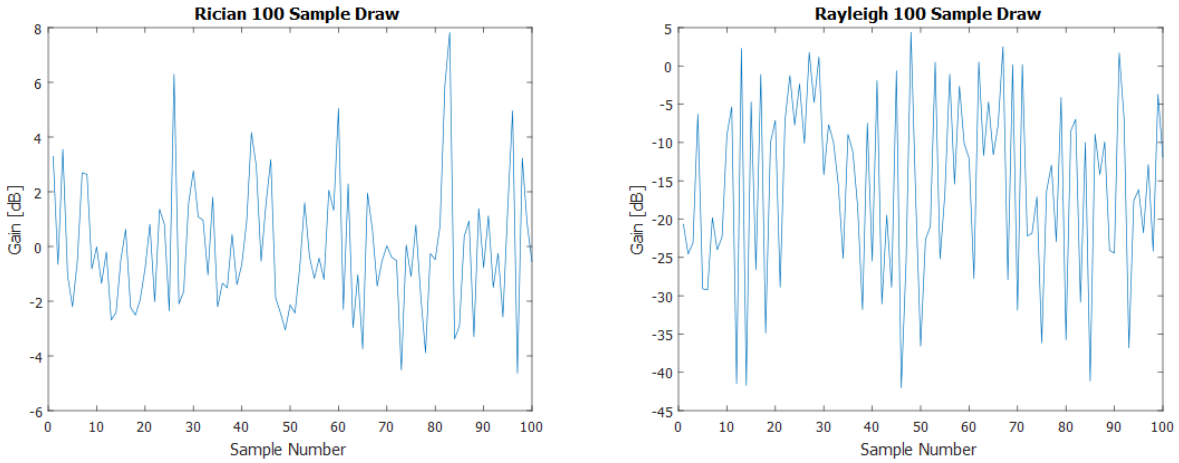

Figure 3.2: The two charts represent 100 drawn samples from the respective multipath distributions, Rician and Rayleigh

Although there have been a variety of methods proposed to generate realistic multipath fading that has the form of the mentioned distributions, this work will draw samples directly from the distributions with specified parameters.

Table 3.1 defines the parameters for these distributions and Figure 3.2 shows 100 samples drawn from their respective distributions. The parameters were tuned in order to match several measured data sets [48, 47]. For the Rician distribution the mean is close to 0 with almost all variations between -5 and 5 dB while the Rayleigh distribution has a larger variation in fading up to -40 dB with occasional +5 dB attenuation.

3.1.4 Sensor Measurements

The previous sections described the methodology used to generate the mean path loss for a given position and the time varying multipath noise that is added. The last step in the model generation process determines the path loss that the aircraft's receiver will experience. Most sensors would provide an indication of the received signal strength rather then the path loss directly. In

Parameter	Rayleigh	Rician
σ	15	4
ν	NA	0

Table 3.1: Table of the Simulated Multipath noise parameters

generating the signal strength value, the pointing behavior of antenna is taken into account where the roll angle affects the aircraft's antenna gain.

$$P_r = P_t + G_t + G_r \cos(roll) - L_{nature}(\mathbf{p}) \tag{3.6}$$

Equation 3.6 calculates the received signal strength in dB where G represents the gains of the channel (transmitter and receiver respectively), P is the power received or transmitted, and L_{nature} is the mean path loss drawn from the nature run mean field. When the aircraft backs out the path loss by rearranging the equation (looking for L_{nature}), estimates of the gains of the channel are used creating a minor sensor bias.

$$z = L_{measured} = P_t + G_t + G_r - P_r \tag{3.7}$$

where is z to denote a measurement.

3.2 Simulation Assessment

A series of simulation experiments were conducted in order to assess the hybrid architecture in providing an accurate prediction of the mean path loss, with emphasis on assessing the performance of the Gaussian process regression. The simulations involved a small UAS flying a lawn mower pattern over a region of interest with the architecture performed post flight. Three main interests are investigated in these simulations: i.) the ability for a Gaussian process to learn the nature run's baseline model with noise, ii.) the ability for the GP to predict the disturbance model, and iii.) the architectures ability to predict the nature run's mean field.

For the simulations, two transmitters were put into the environment as signal sources for the aircraft to map. The positioning of the two nodes allowed for both line of sight and blocked line of sight sensor measurements. The aircraft flew the trajectory shown in 3.3 which measured 19,215 samples at a rate of 5 Hz. 15 trials were conducted with 2 duplicated aircraft each creating the

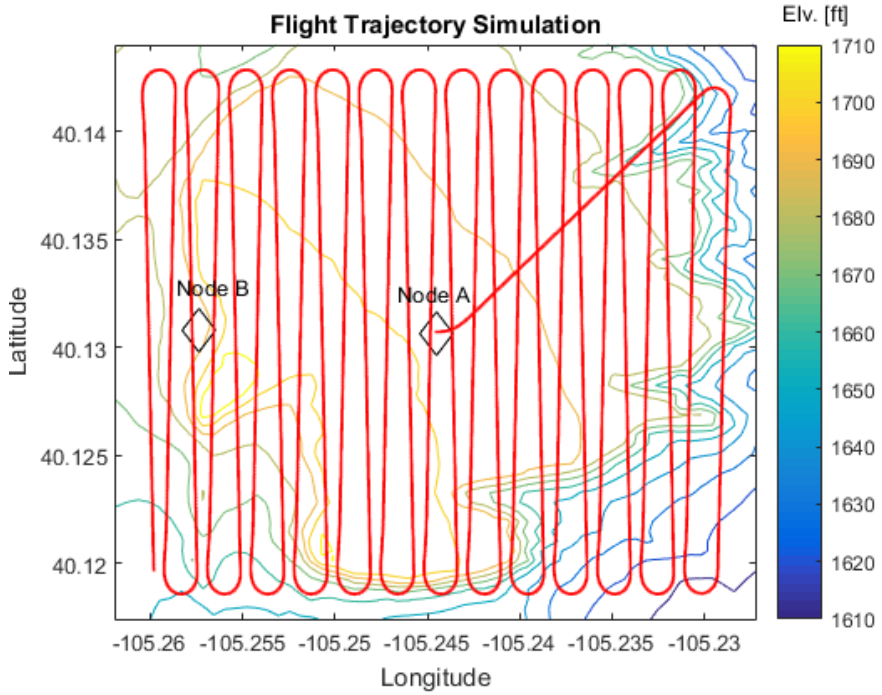

Figure 3.3: Trajectory shown of the aircraft's lawn mower pattern overlayed on terrain elevation contours with markers indicating the location of transmitter sources.

data sets that were analyzed. Further analysis was performed by sub-sampling the 19,215 samples for each aircraft to a 1 Hz rate maintaining the shape but reducing the density.

3.2.1 Baseline Model Learning

In the first 5 simulation trials the disturbance model was set to be a constant of zero and data was collected for both baseline models. The aircraft collected and learned path loss measurements directly in an attempt to predict the baseline model using the Gaussian process. This prediction provided by the Gaussian process was then compared to the associated nature run's mean field to assess performance.

Figure 3.4 displays results for a single trial when the baseline model was SPLAT! while figure 3.5 displays results for a single trial with FSPL as the baseline model. In both figures, the output is after 19,215 path loss samples were used in the correction model's Gaussian process. For both figures, a (left) is the predicted results from the GP and b (right) is the nature run's mean field

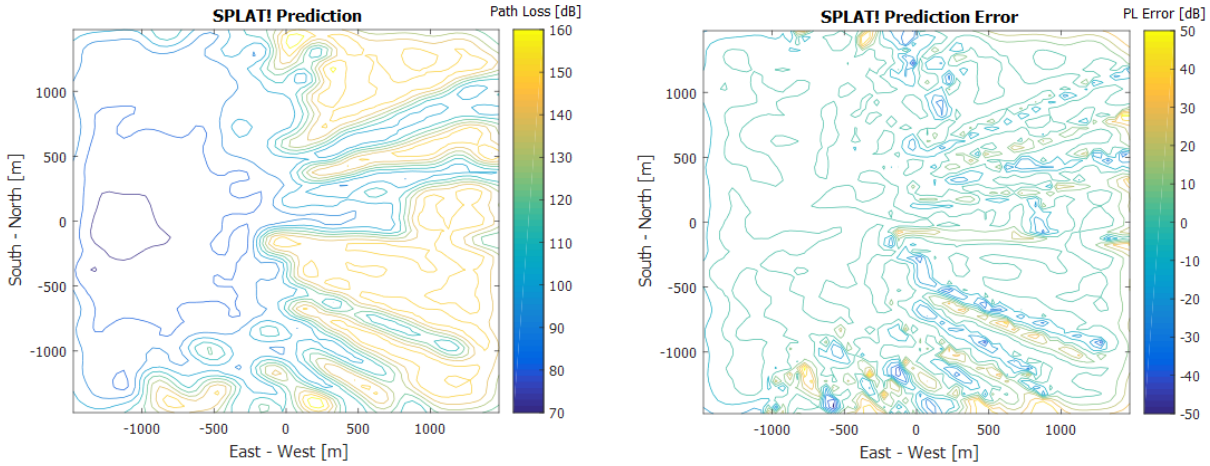

Figure 3.4: Contour plots representing the prediction of the GP trying to learn the SPLAT! baseline model used in the nature run. Left plot shows the field for the learned path loss and the right plot shows the error between the nature and learned path loss.

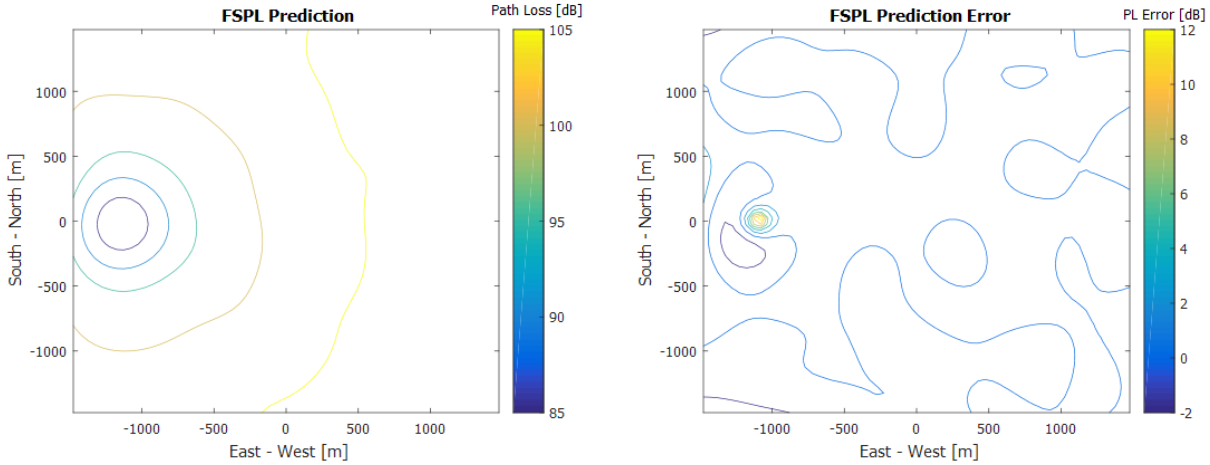

Figure 3.5: Contour plots representing the prediction of the GP trying to learn the FSPL baseline model used in the nature run. Left plot shows the field for the learned path loss and the right plot shows the error between the nature and learned path loss.

minus predicted. In these examples the mapping is associated with transmitter node B's position.

In both of the shown examples the GP properly captures the shape and form of the baseline model. In the SPLAT! case large errors come from small regions that have significantly different path loss than the surrounding areas due to terrain effects. For the FSPL case it does not quite capture the significant drop off right next to the source. An important result to note is that the sensor noise does not appear to be hampering the learning as can be seen by the majority of the

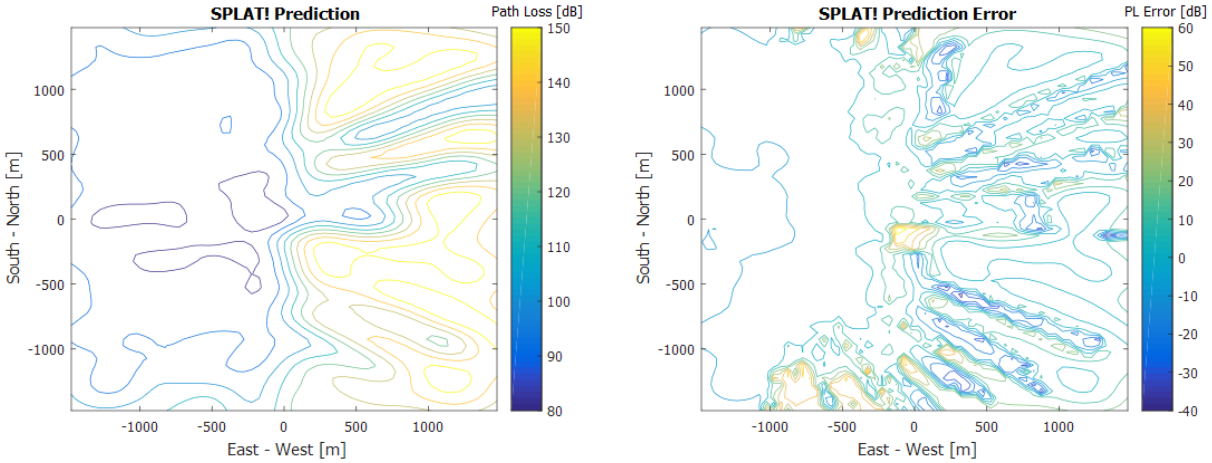

Figure 3.6: Contour plots representing the prediction of the GP trying to learn the SPLAT! baseline model used in the nature run. Left plot shows the field for the learned path loss and the right plot shows the error between the nature and learned path loss.

FSPL being within +/- 2 dBs. Rather the difficulty in learning comes from the way terrain changes the correlation between points

Figure(s) 3.6 represent the SPLAT! case previously (Figure 3.5) but when sub sampled to a 1 Hz measurement rate resulting in only 3,843 samples. The overall shape is still learned however it is not as refined as the previous example. This result is expected since 80% of the samples were removed but does indicate even with low resolution samples the GPR can infer a lot of information.

Table 3.2 lists the average performance over 10 data sets for each baseline and node transmitter. The table reports that the mean errors are all within 1.5 dBs for the map. This matches with the two example trials shown previously. In addition we see that the variance is large, signaling that even though the mean is close to 0 there are still areas that have large prediction errors. There is also a substantial difference between within the variation between the two nodes where node B is twice that of node A. This can be explained by the terrain effects and loss of line of sight in the far reaches, which results in larger multipath noise. In addition, the SPLAT! model would make learning the effects of the terrain more difficult.

The GP performs well with the sub-sampled data sets as well shown in table 3.3. We see that for node B the performance is very similar despite having significantly less samples, however for

Model/Node:	SPLAT! A	SPLAT! B	FSPL A	FSPL B	Units
Mean Error:	0.9829	0.2300	-1.1032	-0.4507	dB
RMSE:	9.3297	13.8929	9.5237	13.7294	dB
Variance of Error:	94.9904	212.3122	98.9653	207.2580	dB

Table 3.2: Table displaying the statistics for the baseline learning trials

node A the mean error has grown but the variance is lower when compared to the larger sampled results. This likely indicates that high density samples in a Gaussian process may cause over fitting of the data treating the larger variations as noise rather than truth.

3.2.2 Disturbance Model Learning

The next set of simulations investigated the ability of the GP to learn the disturbance model injected into the nature run assuming that the baseline model and a-priori model are exactly the same. This allows for the model errors to be only those of the disturbance injected and sensor noise. Two different disturbance models were tested, linear with parameters $m = 0.1$ and $b = 2$ and normal with $\mu = 7.1$ and $\sigma = 13.4$. (The normal parameters come from [25])

Figure 3.7 depicts the error between the disturbance model and the correction model's prediction assuming that the baseline was perfectly known for a single trial. 3.7a depicts the results for the linear model and 3.7b depicts the results for the normal model. The linear model was almost completely learned while the normal model had mostly zero error but includes a lot of points where the models mean guess was significantly off.

Another metric for determining the performance of learning the disturbance model is to evaluate whether the covariance predicted from the GP at any given point contained the correct value of the disturbance model. Figure 3.8 maps the correct value to the 95% confidence interval

Model/Node:	SPLAT! Node A	SPLAT! Node B	Units
Mean Error:	-2.4381	0.2028	dB
RMSE:	7.1952	13.934	dB
Variance of Error:	49.2837	200.2598	dB

Table 3.3: Table displaying the statistics for learning with the sub sampled data points of a SPLAT! baseline

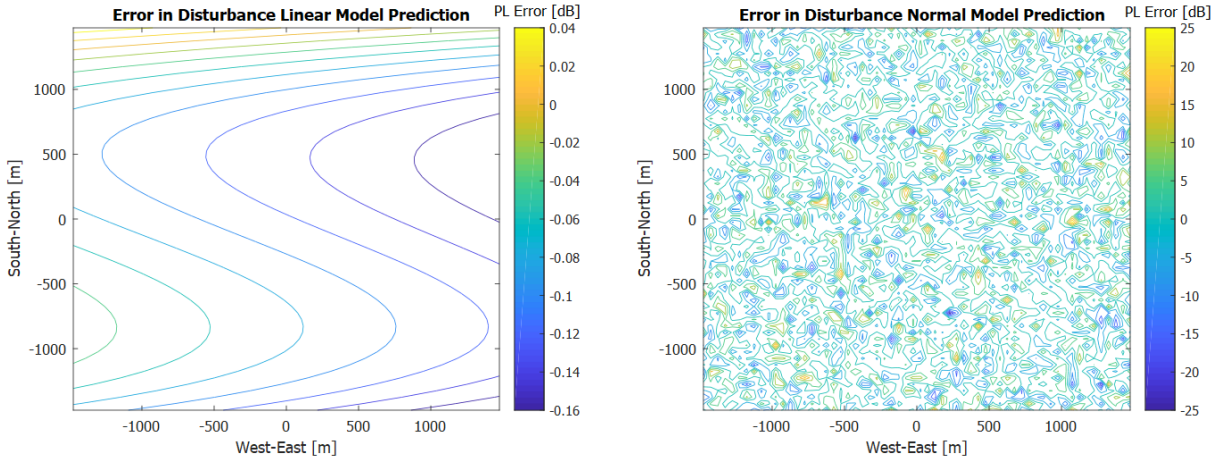

Figure 3.7: Contour plots depicting the error in learning the disturbance model with the left being the linear model and the right being the normal model. Left plot shows the field for the learned path loss and the right plot shows the error between the nature and learned path loss.

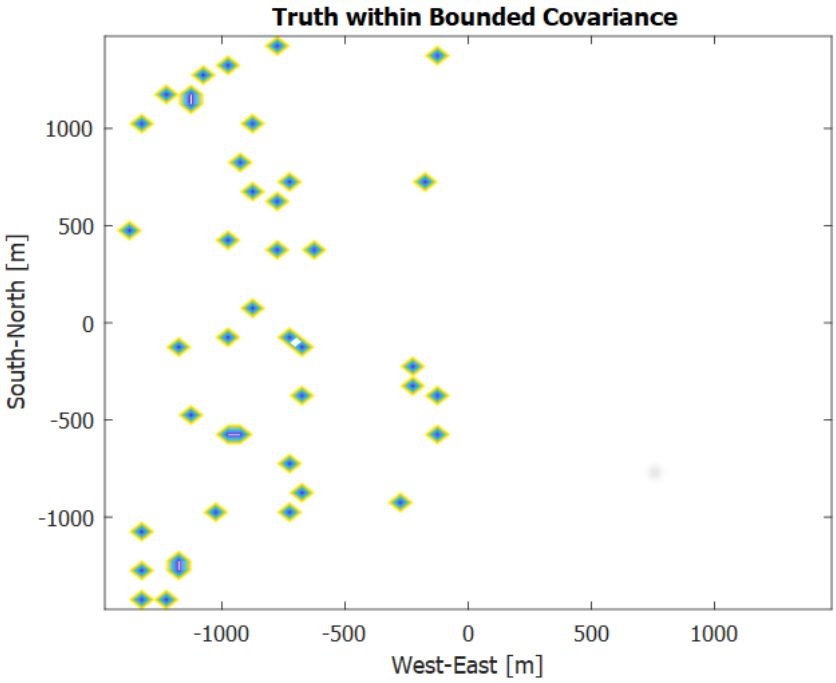

Figure 3.8: Depicts a contour plot where the marked values indicate where the covariance did not properly capture the nature run's mean value.

(CI) and gives a binary indication of whether the correct value lies in that region. The marks that are blue on the left side indicate a few points were the covariance did not capture the disturbance

model's values but otherwise all other values were.

Disturbance Model/Node:	Normal Node A	Normal Node B	Linear Node A	Linear Node B	Units
Mean:	-0.0109	-0.0495	-0.0541	-0.0537	dB
RMS:	6.9325	6.9576	0.0685	0.0685	dB
Var:	48.0387	48.4316	0.0017	0.0018	dB

Table 3.4: Table displaying the statistics of the disturbance model learning

There were 20 trials of each model and node combination. There are no differences in how the disturbance models were generated between nodes so they should be similar. This results is indicated in table 3.4. The mean errors in learning the disturbance injected (for both models) were with a tenth of a dB. The variance of the linear model is practically 0 while the normal model had lower variance than that of learning the entire nature run's mean field. In these trials it can be said that the hybrid architecture was able to confidently learn a linear model with negligible error.

3.2.3 Full Model Learning

The last set of simulations involved using different a-priori models with the GP correction tool to learn the mean field of the nature run. Half of the trials were run with a baseline model of FSPL and an a-priori of SPLAT! and others had the opposite configuration. Two different disturbance models were included; linear with parameters $m = 0.1$ and $b = 2$ and normal with $\mu = 7.1$ and $\sigma = 13.4$. (The normal parameters come from [25])

In figure 3.9 the error is again centered around 0 for most of the field. The hybrid architecture that is applied to a nature run with a linear disturbance model is smoother and has decreased the overall errors when compared to the case when the architecture is applied to a nature run with a normal disturbance model. This is intuitive based on previous results. However both examples show large errors in similar locations with similar errors indicating a dependence on the a-priori model. This is especially true near the edges of the map that were not measured and so the prediction becomes heavily influenced by the a-priori model used. This behavior was apparent when the opposite configuration was investigated.

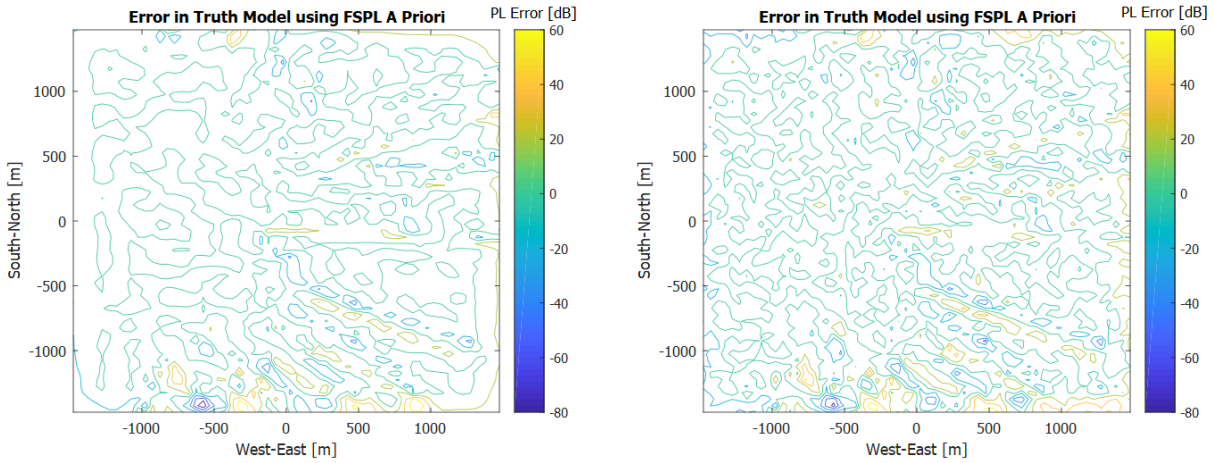

Figure 3.9: Contour plots depicting the error in learning the true model with the left being with a linear disturbance model and the right being with the normal disturbance model.

Disturbance Model/Node:	Normal Node A	Normal Node B	Linear Node A	Linear Node B	Units
Mean Error:	-1.1032	-0.4507	-1.1463	0.4505	dB
RMSE:	9.5237	13.7294	7.2937	13.6793	dB
Variance of Error:	98.9653	207.2580	50.2077	197.4031	dB

Table 3.5: Table showing the average statistical results of learning the nature runs' true field

In table 3.5 each column represents the average over 10 samples when SPLAT! is used as the baseline model and FSPL as the a-priori model. The means are close to 0 with all cases being within $+/-1.2$ dB. Node A has a slightly larger magnitude of the cases and Node B has a mean 0.00 between the two disturbance models. This mean data suggests that the node location (and by association terrain effects) have a bigger effect on the resulting mean error of a trial. This comes about from the baseline model and there isn't a major improvement between the two means of a node even with differing nature run disturbances injected. The mean variance results in the table indicate a different behavior. For node B in both disturbance model cases the variance was about 200 dB. There was a minor decrease for the linear disturbance model case but it was not substantial. This fits with the conclusion drawn from the mean data. However, the mean variance between the two disturbance models for node A are substantially different. In the normal disturbance injection, the variance is almost twice that for the linear case. While this makes sense, it would be expected to be on a similar scale to that for node B.

The important result to draw from this table is that on average the hybrid architecture was able to have the same performance in mapping the nature run's mean field when compared to using the GP method directly to map the entire field (done in the baseline test). Furthermore, a scenario was discovered where the hybrid architecture was able to on average substantially out perform the direct GP method.

3.2.4 Results Summary

An assessment was performed with simulations evaluating the hybrid architecture in three areas: i.) the ability for a Gaussian process to learn the nature run's baseline model with noise, ii.) the ability for the GP to predict the disturbance model, and iii.) the architectures ability to predict the nature run's mean field.

The simulated results indicate that a GP with a squared exponential kernel can learn the path loss mean field with a mean error within 1.5 dBs and a large variance. When learning the baseline models the complexity of the shape and number of sharp jumps is a large factor int he performance of the GP's learning. As expected with machine learning tools, higher density of samples allow for better mapping of the mean field but had a side effect of increased variance in the error due to over fitting when compared to lower sampled datasets. The GP was tested with learning disturbance models that had strong and weak correlation. It is found and was expected that weakly correlated fields were more difficult to learn resulting in higher variance errors however the mean was similar to that of strong correlated field. A simple linear field was able to be completely learned. In the analysis of the GP learning the disturbance fields the GP's predicted covariance was able to properly capture almost all of the variation in the field. The results when the full hybrid architecture was applied demonstrated performance on par with the GP directly learning the nature run's mean field but had a case where the hybrid architecture out performed using the GP directly.

Other important notes from this assessment are that the GP used does not provide accurate mean predictions far from where samples were taken leaving the a-priori model to dominate mean prediction behavior. This could be relevant for certain missions and proper management is needed.

An important advantage of the hybrid's architecture is that it does not need a previous flight to provide information and was shown in the case of mapping to do as well as mapping directly through a GP on data.

Chapter 4

Wireless Relaying Simulation

This chapter will discuss the simulation and results for an application test with the online hybrid architecture. The setup of the simulation will be provided with a discussion of what is wireless relaying. A utility function is developed for optimal positioning in the relaying application and then the results of the simulation are shown.

4.1 Relay Application Setup

The principle behind an airborne relay application is to put the aircraft into a position that would enable or increase the performance of a communication link between two or more ground nodes. In relaying scenarios the goal is to act as a repeater, retransmitting the signal with minimal processing minimizing any delay that could occur. Thus the goal is to increase the data rate between the two ground nodes.

In the test scenario considered here, an unmanned aircraft is tasked to find the global optimal relaying point between two ground nodes when restricted to a region around the nodes. To simplify the problem and to evaluate the architecture, this region is discretized into a grid structure 4km by 4km with cells having 50 meter resolution in the X-Y plane. The grid will also have a vertical component from 50-250 meters AGL with 10 meter spacing. The test scenario was simulated in an area that could be flown in the near future for validation and comparison. The chosen area was the Pawnee National Grasslands located in north-eastern Colorado. A picture of the area that was used for terrain data is shown in Figure 4.1. The terrain is comprised of small rolling hills

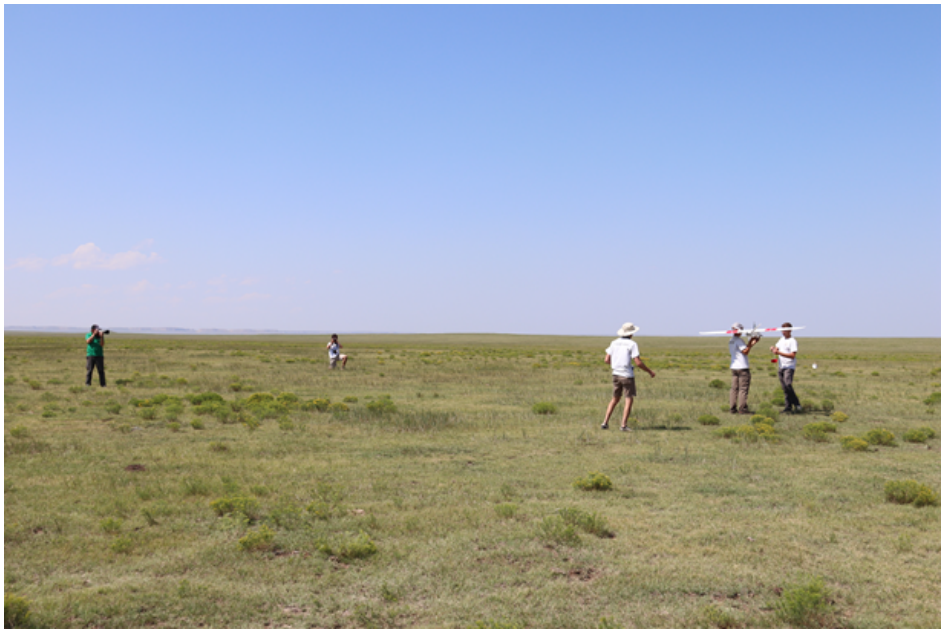

Figure 4.1: Picture of a sUAS launch located in the Pawnee National Grasslands of northern Colorado. The terrain consists of gentle rolling hills and light vegetation

and light vegetation. A box with 4 kilometer edges was defined to be the region of interest for the simulation and four transmitter node locations were selected within that box. Figure 4.2 depicts the the topographical terrain and transmitter locations selected for the simulation.

A dataset consists of 20 trial simulations where each trial will have a new nature run created prior to running the simulation starting. Four transmitter pairs were used with five trials per pair. The SPLAT! baseline model and normal disturbance model were used for every trial. The parameters for the normal disturbance model were derived from flight data analyses discussed in chapter 5. Two datasets were simulated and analyzed to assess several aspects of the architecture. The aspects that were of interest included the starting location, the initial waypoint , and the a-priori model used in the architecture. In each trial, three aircraft were simulated given the same starting conditions. The start position was randomized to be within the region but far enough away from the edges to prevent geo-fence violations. The starting altitude is set to 100 meters AGL from the start position. Each aircraft were given a different a-priori model and are listed in table 4.2. In

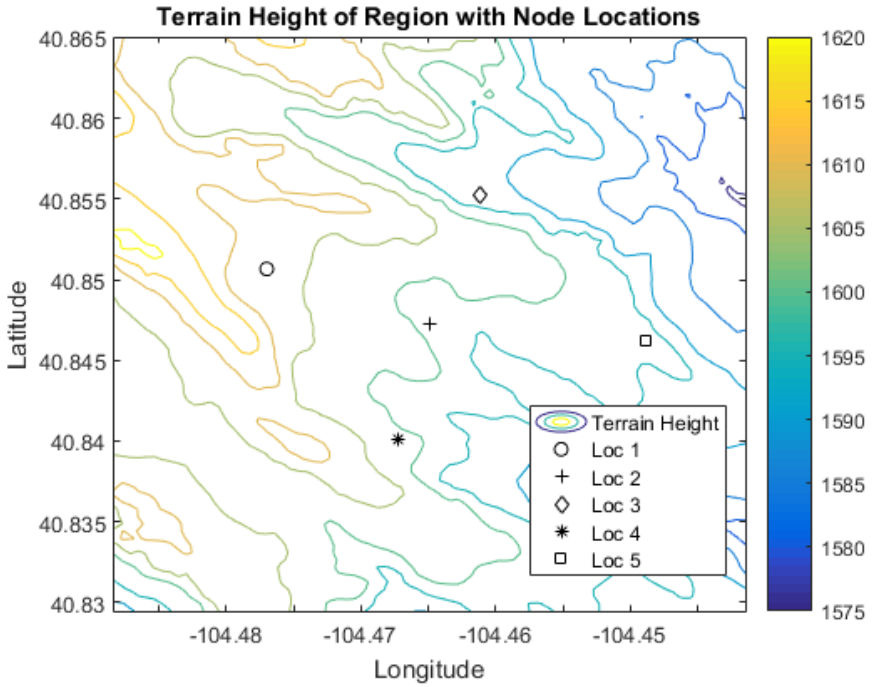

Figure 4.2: Image of terrain elevation contours with the marked locations for the 4 transmitter nodes that will be used.

Parameter	Value
Simulation Time	900 seconds
Time Step	0.01 seconds
RSS Data Rate	5 Hz
Correction Step Processing Time	60 seconds
Normal Disturbance Model μ	0.2301 dB
Normal Disturbance Model σ	4.2175 dB

Table 4.1: Table of the Simulation Parameters

the first dataset all the aircraft were given the same initial waypoint at halfway between the two nodes, which is representative of a naive approach. In the second dataset the a-priori model is used to provide an initial waypoint.

As the aircraft are flying in the simulation they are collecting RF measurements from both ground nodes, which are assumed to be cooperative and static. Each aircraft runs two hybrid models, one for each node, that provide the path loss predictions to be used to find the optimal relaying location. Every 120 seconds (with the first starting 30 seconds) the hybrid models run the GP correction tool to update the path loss predictions. The updated values are then used in the

Aircraft Number	A-Priori Model
Aircraft 1	SPLAT!
Aircraft 2	FSPL
Aircraft 3	previous mapped flight data

Table 4.2: Table of the a-priori model for each aircraft

Location No.	Latitude	Longitude
Location 1	40.850658	-104.477009
Location 2	40.847252	-104.464897
Location 3	40.855262	-104.461202
Location 4	40.840063	-104.467257
Location 5	40.846242	-104.448894

Table 4.3: Table of the transmitter node locations.

utility function to find the optimal location to loiter the sUAS. For each trial, the aircraft are given 15 minutes and the final waypoints used represent their best guess.

4.2 Utility Function

The utility function is needed that provides a way to characterize the performance of a communication link that can then be used for optimizing. There is a link between the channel capacity and the attenuation of a signal.[49] Channel capacity represents the theoretical maximum bit rate that a channel could attain given a prescribed signal to noise ratio. In practice, this is never achieved in wireless communication due to encoding methodologies and error correction techniques that help ensure successful data transfer through a medium. However, for a given encoding method and error correction technique, the channel that has a higher channel capacity will have a higher goodput, a term to describe the useable data transfered over a channel.

The Shannon-Hartley theorem states 4.1 that the capacity C in bits per second (bps) achievable by a channel is proportional to the bandwidth B of the channel in Hertz and the signal to noise ratio (SNR) $\frac{P_r}{N}$

$$C[\text{bps}] = B\log_2(1+\frac{P_r}{N}) \approx .332B * \text{SNR[dB]}. \tag{4.1}$$

An approximation of this in decibels is also shown in Equation 4.1 for SNR values much greater than 1. In order to increase the maximum capacity of a channel the user has two options, increase the bandwidth or increase the signal to noise ratio. Bandwidth sizes are often limited by a regulatory

commission for a given frequency band so the only action is to increase the signal to noise ratio. If we assume that the noise a receiver experiences is constant for a given area over a short period of time, then by maximizing the received power the SNR is increased. The received signal strength was shown in equation 3.6.

The primary methods for increasing the received signal strength is by increasing the transmission power, the gains of the system on both ends of the channel, and reducing the losses the signal experiences. Transmission power is also limited by regulatory commissions and system gains are hardware dependent. The only possible method is to reduce the losses from signal propagation which results in the maximum power received for a given channel / hardware setup. Thus maximizing the channel capacity of the channel.

The goal in the relaying application is maximizing the throughput for both channels combined. The weakest channel capacity of the relay limits the overall throughput. The utility function is defined to balance the two individual channels based on the total path loss for both channels. The derived utility function is shown in 4.2

$$J(\mathbf{p}) = |\hat{L}_i(\mathbf{p}) - \hat{L}_j(\mathbf{p})| + \hat{L}_i(\mathbf{p}) + \hat{L}_j(\mathbf{p}) \tag{4.2}$$

where $\hat{L}_i(\mathbf{p})$ is the predicted path loss from transmitter i at position $\mathbf{p}$. The absolute value of the difference between the two transmitters' path loss provides the balancing term. The predicted path loss values come from updates in the hybrid architecture. Using the same discrete approach the hybrid model takes, an exhaustive search can be run over all the positions to find the one with the minimum cost. This position is then fed into the guidance loiter algorithm (Appendix A).

4.3 Simulation Results

The following section discusses the results of the 40 trials simulated in the two datasets. First will be a discussion on the behavior of the aircraft during the simulations followed by the individual aircraft's performance in achieving the best possible location to loiter at. This performance is compared across aircraft and to a naive approach of just loitering halfway with no information.

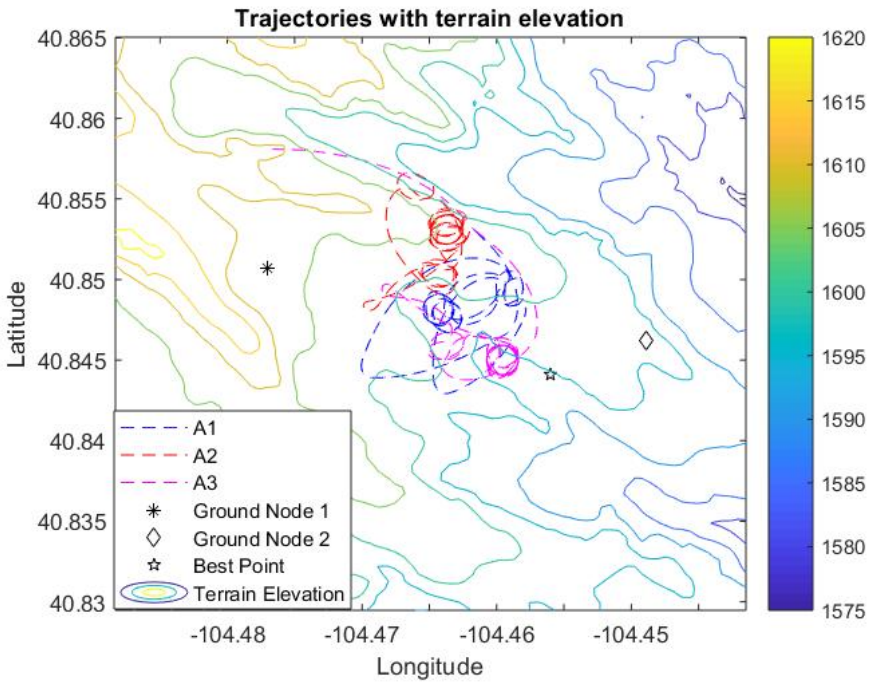

Figure 4.3: Figure depicts the trajectories of the three aircraft over the 15 minute mission simulation with the nodes and best location marked. This information is overlayed on terrain elevation contours

4.3.1 Aircraft Behavior

Figure 4.3 shows the trajectories of the three aircraft for the duration of the mission. The aircraft all start north of Ground Node 1 and begin flying to the center point between the two nodes. As the aircraft approach a new update occurs from the communication-aware architecture providing a new guess on the best location to loiter. This shifts the aircraft trajectories and allows them to collect new data. Eventually the aircraft settle into a spot they believe is optimal and remain there. In this example the best location is south east of the naive approach and aircraft 3 comes closest to the point.

An important part of this architecture is that it allows the aircraft to immediately begin performing the mission and use those trajectory segments to inform future locations to loiter. Even though no trajectories are generated, the simple guidance loiter algorithm creates a trajectory over time based on how that loiter point changes. This is a nice feature of the architecture and hints at

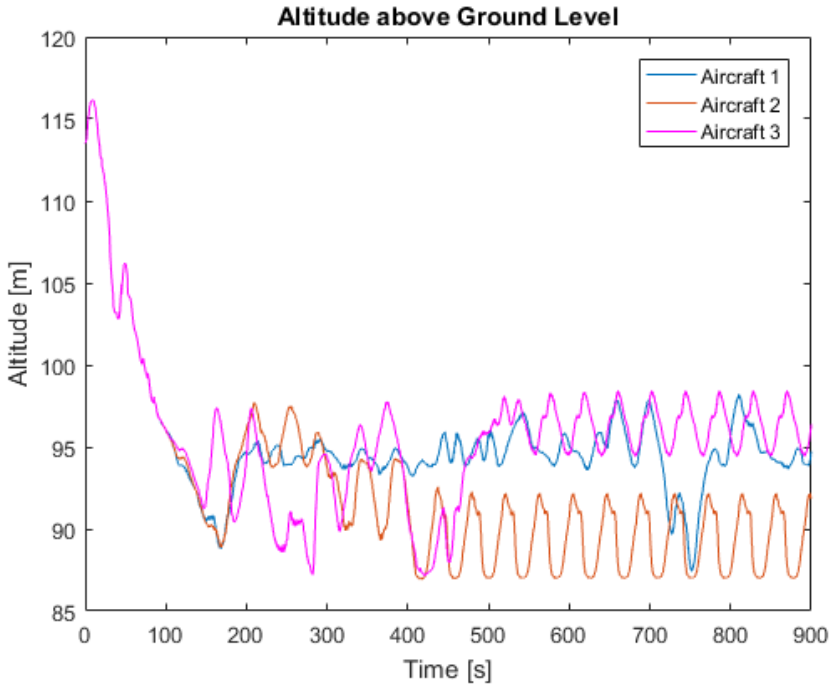

Figure 4.4: Altitude above ground (AGL) for all aircraft in an example trial

the potential of the architecture for use in other applications.

Figure 4.4 depicts the altitude over an example trial for all three aircraft in AGL. Almost all of the variations of altitude are within 20 meters of the started 100 meter due to the fact that the aircraft is flying constant MSL. Since SPLAT!'s prediction used in this case assumes the aircraft will always be at 100 meters AGL modeling errors are introduced. However, these errors will be corrected through the Gaussian process.

4.3.2 Performance

Figure 4.5 compares the final waypoint cost relative to the best possible cost for the first dataset. It can be seen in most of the trials the three aircraft all perform better than the naive approach of loitering at halfway. Aircraft 3 appears to do the best of the three, which is not surprising in that it had previously flown and collected data over the entire area. When compared to the naive approach there appears to be an average 10 percent increase in reducing the communication cost

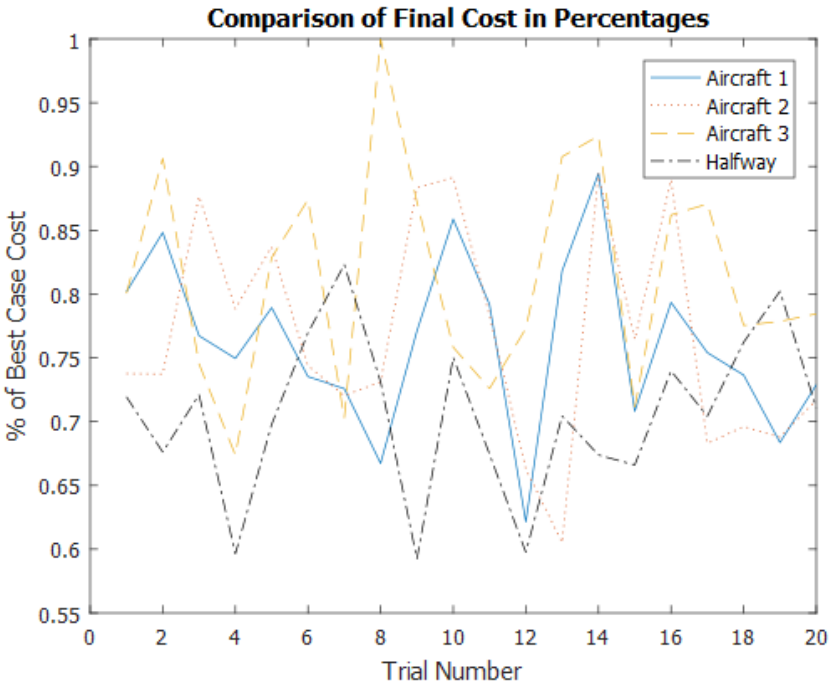

Figure 4.5: This figure displays the final cost achieved by the three different aircraft when compared with the best possible case (100 percent) and the halfway point when the initial waypoint is set to be halfway

with the hybrid architecture. Also in this data we can see that the a-priori model does not appear to have a strong influence on the final performance for this simulation scenario. This behavior can likely be explained by two factors; the Gaussian process is learning the model error and will account for any deviations and by the end of the flights the hybrid architecture's predictions are dominated by the Gaussian process.

Figure 4.6 shows the same comparison as the previous figure, but when using the a-priori model to drive the initial waypoint in dataset two. The performance of aircraft one is significantly worse then in dataset one. Also, on average over all three aircraft there was less success in minimizing the cost when compared to the naive approach. However aircraft one had more consistent performance aside from one outlier then compared to the others. Aircraft one's cost sat at approximately 75 percent of the best cost. Aircraft three had a large variation, achieving as high as 90% best cost for four trials and then achieving as low as 60% for three trials. Aircraft

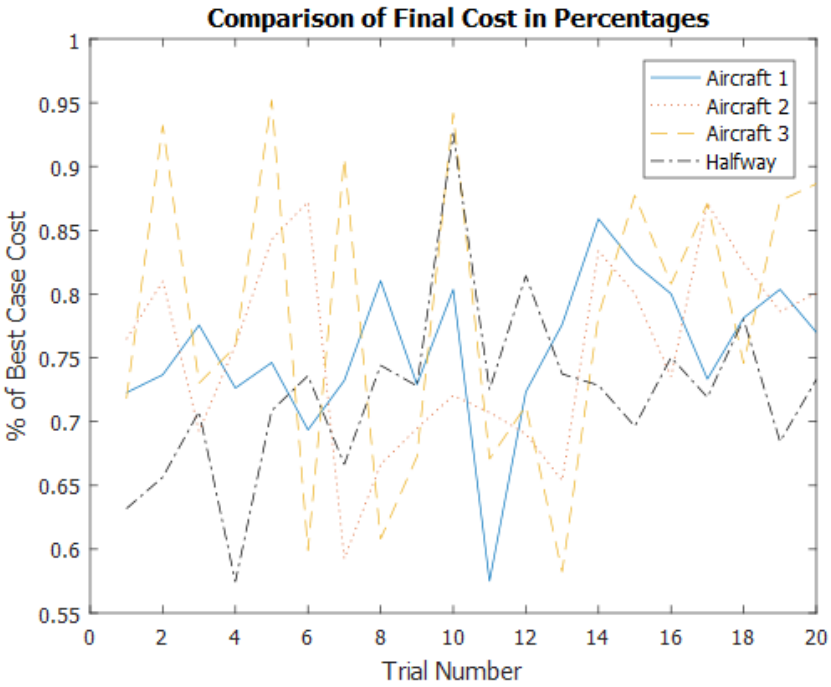

Figure 4.6: This figure displays the final cost achieved by the three different aircraft when compared with the best possible case (100 percent) and the halfway point when the initial waypoint is set from A-Priori Model

two's performance sat between the other two aircraft in consistency. A surprising result is that it was expected that the performance of aircraft one and three would increase while the performance aircraft two would remain the same since FSPL is equivalent to the naive approach. However this behavior did not appear to happen. A possibility is that the trials used in dataset two ended up having statistically worse nature run generations than the first dataset. This is indicated by the halfway point performing five percent better in dataset two than dataset one.

In the figure 4.7 both images display the relationship between the final cost for aircraft's one and two and the relative start distance away (4.7a) or the relative distance between the naive position and best position (4.7b). Starting far away appeared to have the same level of performance as starting close and there was no indication that the distance between the best position and naive position had any impact on the performance.

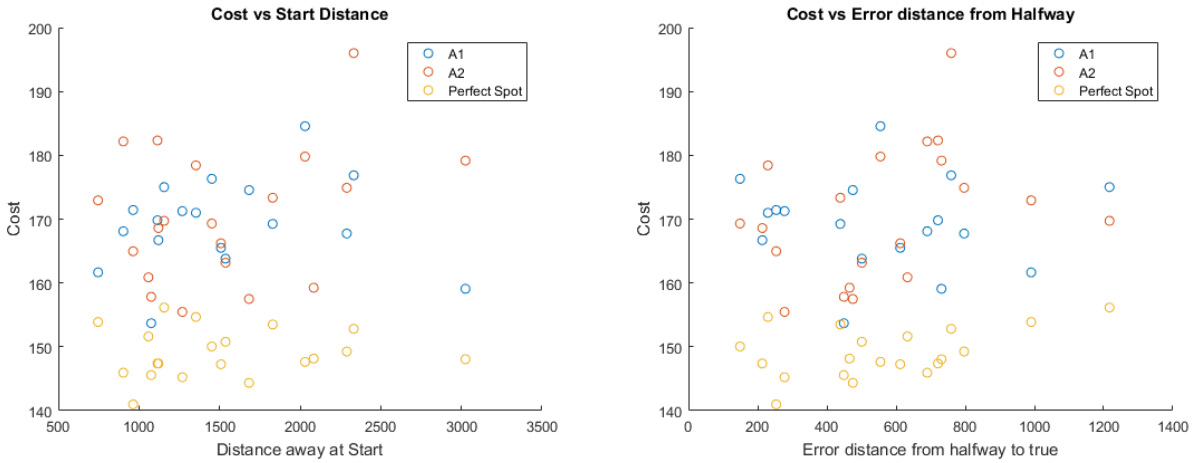

Figure 4.7: The images depict the final waypoint cost relative to two different distances between aircraft 1 and 2 and the best possible case. The chart on the left uses the start distance away from the best possible relay location as the x-axis. The chart on the right uses the distance between the initial waypoint from the best possible relay location as the x-axis.

4.3.3 Summary

A wireless communication relay example using sUAS between two ground nodes was simulated in order to evaluate the architecture in a Communication-Focused operation. A set of transmitters was placed at the Pawnee National Grasslands and three aircraft were flown with varying A-Priori models in the hybrid architecture. The results indicate that on average there was a small improvement over the naive approach in decreasing the utility function's cost. Two data sets were simulated, the first using the naive approach to drive the first waypoint and the second using the a-priori model. The aircraft that performed the best utilized previous flight data as its a-priori model, however it was not consistent. Aircraft one had the most consistent behavior utilizing SPLAT! as its a-priori model. Overall there was not a significant difference between a-priori models impact on performance, a result that can be explained by the Gaussian process correction tool fixing any a-priori model discrepancies. The architecture also demonstrated basic exploration-exploitation behavior using only the mean predicted value, a loiter guidance algorithm, and constant update rate of the Gaussian process. It is expected that if these simulations were conducted again in more varied terrain that there might be more distinguishing between the A-Priori models in the

performance.

Chapter 5

Flight Data Analysis

Two flight test data sets were analyzed to help explore the performance of the architecture. Additional data on RF have been collected and may be analyzed in future work to further verify the results reported here.

The first data set was collected while the aircrafts were performing a target tracking mission from the KAIST-CU[50] project in August of 2017. This data set will be referred to as Set 1. In these flights a stationary 2.4 GHz rWiFi source was measured with multiple flights conducted over several days.

The second data set is from a single flight in July, 2018 doing lawn mower patterns to measure the path loss experienced from several 5.8GHz transmitters. In this flight, three transmitter beacons were positioned that would provide interesting terrain effects on the signals being measured by the aircraft. This data set will be referred to Set 2.

With these two datasets, analyses were performed on the two main components of the hybrid architecture, evaluating the a-priori model SPLAT! and the Gaussian process correction model.

5.1 Aircraft Hardware Configuration

Talon Unmanned Aircraft were used in collecting the two data sets (Figure 5.2). The Talon was configured with a Pixhawk autopilot running arduplane software. Onboard the aircraft was a BeagleBone Black companion computer that ran python scripts to collect data and do autonomous control. The sensor configuration for the first data set involved several different radio devices, a

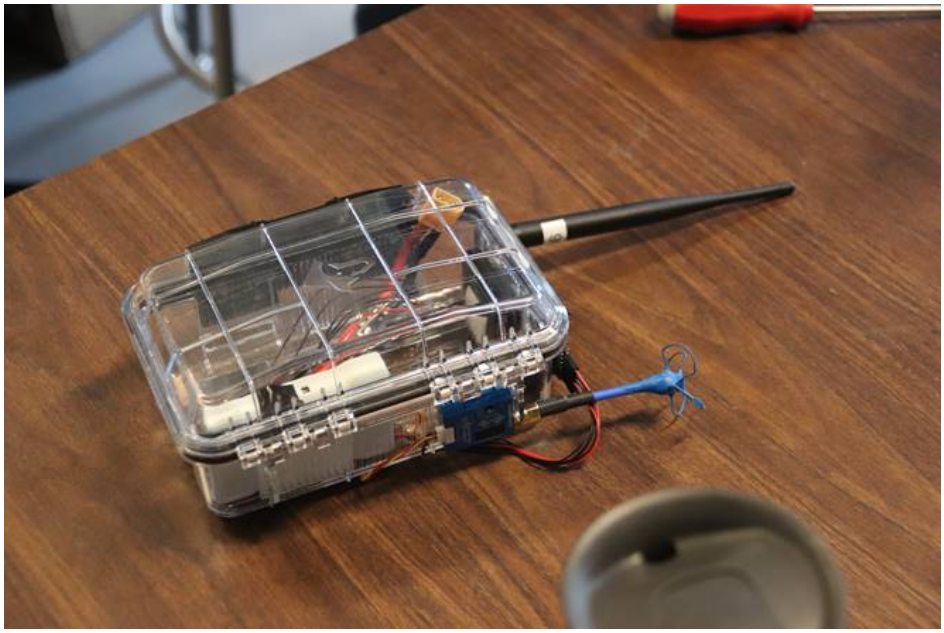

Figure 5.1: Photo of the 5.8GHz transmitter beacon

Figure 5.2: A picture of the Talon UA setup for an RF localization mission with multiple antennas including a 900MHz telemetry radio, 2.4GHz C2 link, and a 5.8GHz binary direction finder setup

900 MHz telemetry radio, a 2.4GHz Wifi module for a command and control (C2) link, and two 5.8GHz antennas with a diversity receiver.

For the July 2018 flights the 5.8GHz system was modified. The original setup used the two 5.8GHz sensors as a binary sensor including a directional and omni-directional antenna. The directional antenna was replaced with a second omni-directional antenna closer to the rear of the fuselage.

Figure 5.3: Image of the ground station antenna used for the 2.4GHz C2 link during summer 2017 flights for KAIST.

5.2 Summer 2017 Data

The 2.4GHz C2 link proved best for evaluating the performance of SPLAT! for two reasons. Interference was low as there were minimal networks in the area and the radio hardware provided a better indicator of the received signal strength, likely directly measuring the power received between the two C2 transceivers. One issue with RF hardware and especially low cost devices is that they typically only provide an indication of the signal strength using percentages or some other metric.[51] This makes it difficult to determine the actual received power the device measured which is necessary in determining the signal's experienced path loss. The Microhard radios used provided received signal strength data between the master-slave channels and allows for evaluating

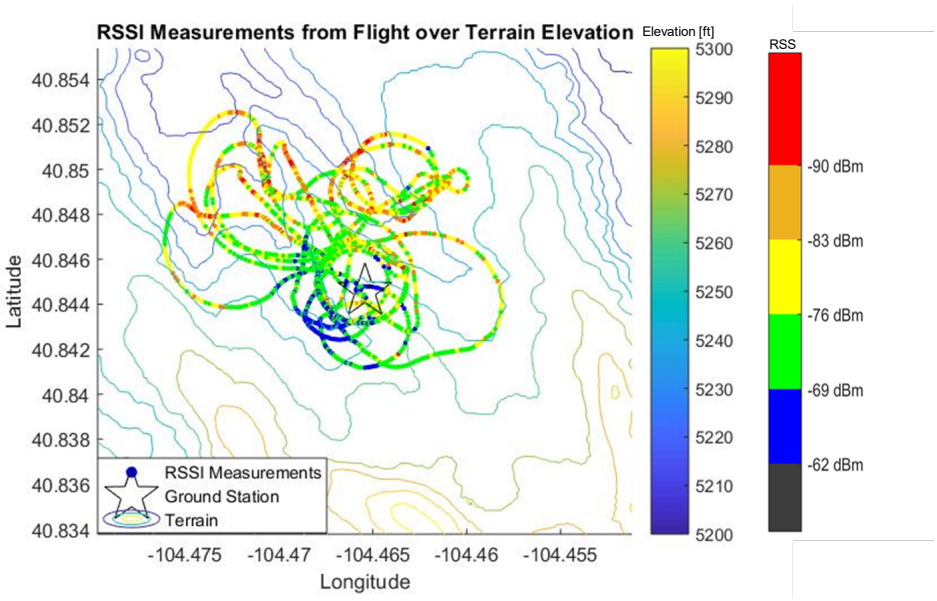

Figure 5.4: Chart displaying the received signal strength measurements along the trajectory of the aircraft superimposed over terrain contours.

SPLAT!'s accuracy and precision in predicting path loss.

The RF data collected in Set 1 included data from multiple aircraft with varying heights depending on the mission flown. One flight might last over an hour with multiple missions, each with two to three different altitudes flown. The data collected was separated into individual trials that would continuous data points for a given altitude. For example a mission that had three different altitudes would be broken into three trials, one for each altitude. In total the data was separated into 36 trials. Half of the trials showed the aircraft loitering and about three quarters of those were around the ground station within a 300 meter radius. The remaining trials involved the aircraft flying all over the area covering a 1.5 km edged box centered around the ground station.

An example trial from Set 1 is shown in figure 5.4. This displays the aircraft RF measurement samples color coded on the RSS value and where the star is the ground station location. This information is overlayed on top of terrain contours to highlight correlations between terrain elevations and RSS values. An interesting set of measurements is that on one side of the ground station there was stronger signal while the other side had weaker signal. There are a variety of factors that

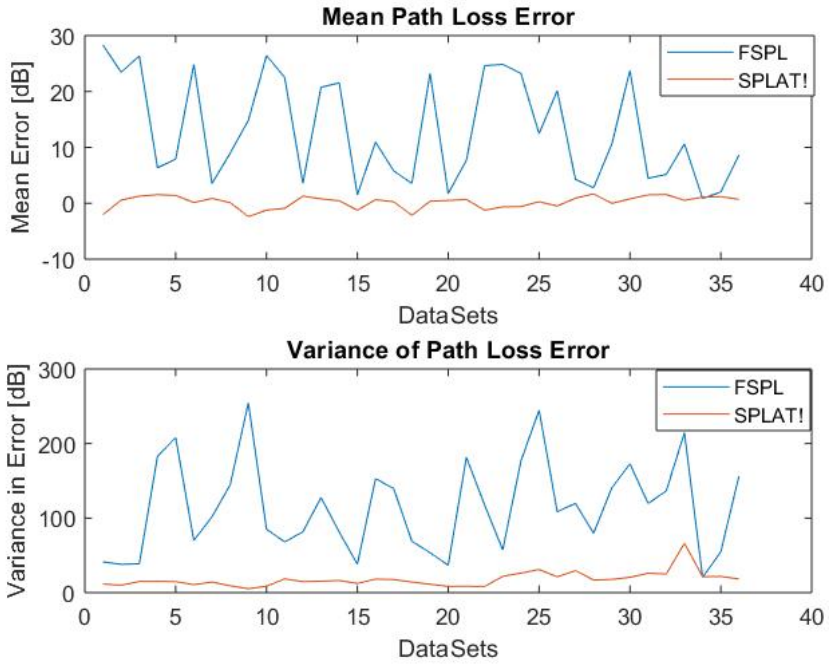

Figure 5.5: This figure shows the mean error and mean variance of the error between a propagation model's prediction and measured values.

could have influenced: antenna alignment, ground station position error, measurement delays, and terrain. Antenna alignment and ground station position error was ruled out based on the setups and data collected. This leaves measurement delays and terrain as likely influences on this result. The positions to the left (west) of the ground station have the same flat terrain beneath while the positions to the right (east) were over a small valley. Ground planes cause significant effects on how antennas radiate energy [52]. The measurement delays come from three different computer systems interacting (Pixhawk, Radio, BeagleBone) and the required processing and sending time for the measurement logs.

For all of the trials the measured path loss was compared to predictions from SPLAT! (in a area coverage approach) and to the free space path loss (FSPL) prediction. Each trial varied in the number of measurement samples ranging from 300 to 4,000. The mean and variance of the errors between the measured and SPLAT! or FSPL are shown in figure 5.5. The blue lines indicate the results from the FSPL comparison and the red lines indicate the results from SPLAT! comparison.

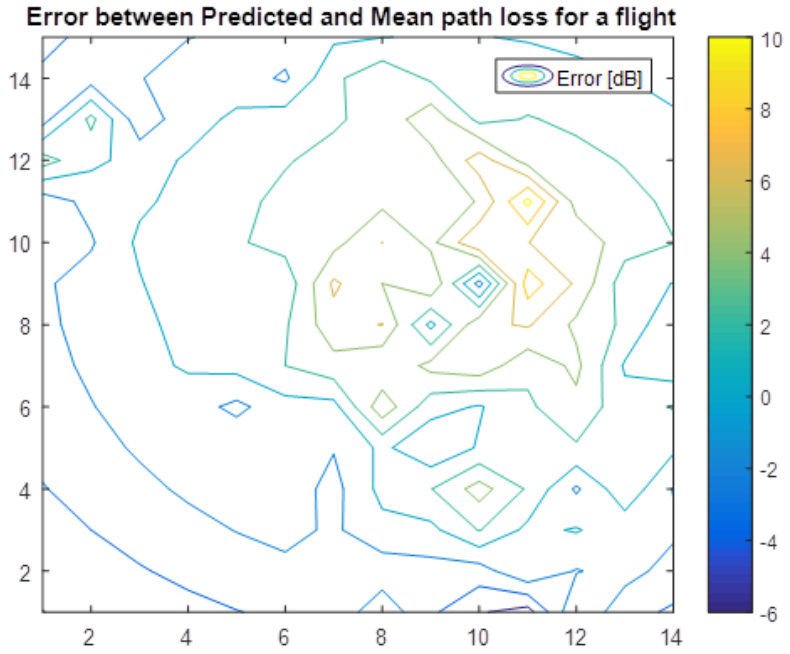

Figure 5.6: Contour map highlighting the errors of SPLAT!'s prediction versus measured values

The comparison in performance between SPLAT and FSPL prediction errors in figure 5.5 clearly show that SPLAT! is an order of magnitude better in terms of both the mean and mean variance error in predicting path loss. This suggests that SPLAT! is far more accurate and precise as a prediction model at this location. It should be noted, however, that all of the data points were within two kilometers of the ground station and there is debate about the accuracy of these RF models within these ranges [26]. That being stated, many current sUAS applications involve flying and communicating over these short distances. Looking at this data, it also shows that the mean error for SPLAT! is around zero, which lines up well with assumption that the deviation the correction model is trying to learn has a zero mean. While this assumption is not required, it supports the principles the hybrid architecture was built on.

Figure 5.6 displays a contour plot of the errors in path loss between SPLAT! and measured for a given trial. Many of the maps from other trials show similar behavior. The first important result to draw from this figure is that the shape of the error is closer to the linear disturbance

model than the normal model used in simulation (2). The error seems to grow larger the further from the from the source which is situated in the middle. This error growth has a sort of radial structure emanating outward from the source with variations in the behavior at select spots. This information can be helpful for future tuning of the RF simulations and hybrid architecture. The second result to draw from this figure is an intuition in what the mean and variance were saying in the previous figure (5.5). While the mean is close to 0, this is a result of equal number of errors being over predicted by SPLAT! to areas being under predicted by SPLAT!. This reveals itself in the variance values associated with the map and thus the variance statistic may provide better indication in performance than the mean.

5.3 July 2018 Data

In July of 2018 two missions were conducted during a single flight to map the RF environment with multiple 5.8GHz transmitters. The goal this flight was to learn the relative path loss of the signal and to investigate Gaussian process performance with real data. The Gaussian process would learn the deviation in the path loss from SPLAT!. In addition various Gaussian process models were experimented with and inspect if models learned with one transmitter could be used to predict behaviors of other transmitters.

For the two missions the aircraft flew lawnmower patterns over the Table Mountain Test Facility near Boulder, Colorado. The first mission provided training data for the GP while the second mission provided data to test the performance results against.

Figure 5.7 depicts the two mission flight patterns and the data samples that were collected. The second mission had significant issues logging data due to poor telemetry communication hence the sparser data points.

Six different models were tested by varying the input parameters in order to assess their relative impact on the prediction performance. Table 5.1 lists the set of variables used as state input for Gaussian process in addition to the path loss. For instance in Model 1 tells the GP that the path loss is only a function of latitude and longitude, while Model 3 says that the path

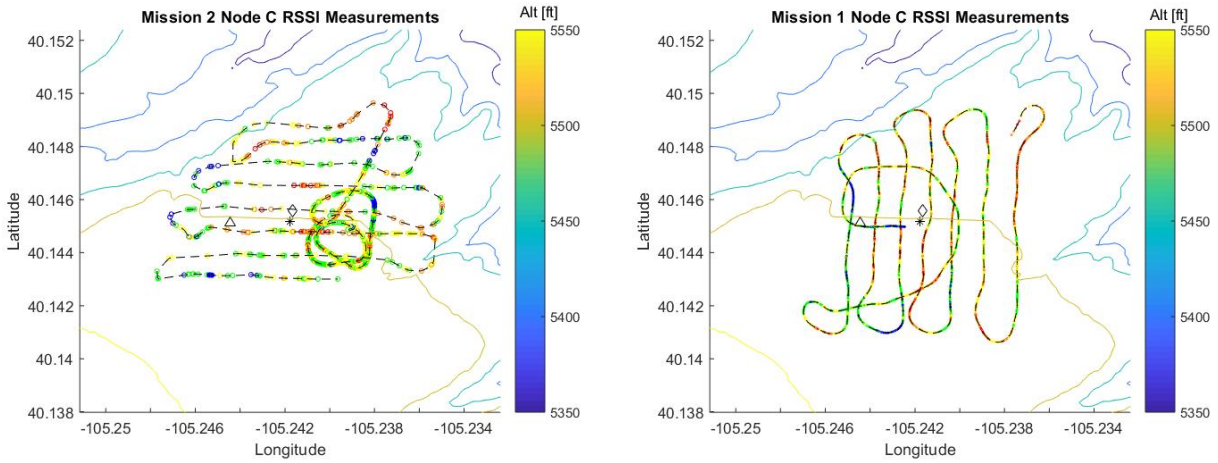

Figure 5.7: Plots of the mission trajectories along with the RSSI data samples that were measured overlayed on top of topographical contours. The three markers represent the locations of the three transmitter beacons.

Model	Input Variables			
Model 1:	Lat-Lon Position			
Model 2:	Lat-Lon Position	Terrain Height		
Model 3:	Lat-Lon Position	Elevation Angle	Azimuth Angle	
Model 4:	Lat-Lon Position	Elevation Angle	Azimuth Angle	Terrain Height
Model 5:	Elevation Angle	Azimuth Angle	Distance	
Model 6:	Elevation Angle	Azimuth Angle	Distance	Terrain Height

Table 5.1: Table of the various input models used for the Gaussian Process Regression. Red text represent transmitter dependent variables

	Model A	Model B	Model C
Tested with Node A:	AA	BA	CA
Tested with Node B:	AB	BB	CB
Tested with Node C:	AC	BC	CC

Table 5.2: Table of the cross model testing matrix

loss is a function of latitude, longitude, the elevation angle, and azimuth angle. The six models would be trained with their corresponding data and the six models would be created for each transmitter, designated as either *Model A/B/C*. Then each of these six models were applied to perform predictions with test points from all three transmitters. The matrix of this arrangement and notation are listed in table 5.2.

The results from the matrix for each of the individual models are shown in figures 5.8, 5.9, and 5.10 for Model A, Model B, and Model C respectively. Each figure has two images, one displaying the mean error for all samples in that dataset and the other for the mean variance of the error. The

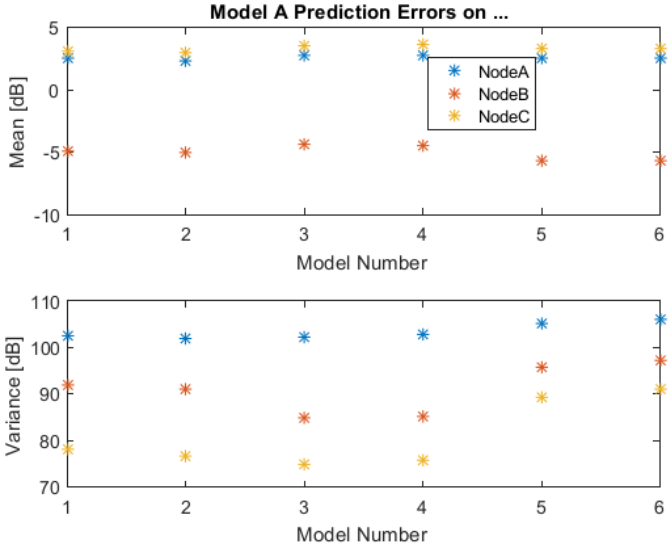

Figure 5.8: Figure depicts the error results between tested points for the six different input models from Node A and each node combination in the test matrix. The left chart shows the mean error and the right chart shows the variance of the error.

x-axis is labeled 1 through 6 for each of the models listed in table 5.1. In each chart, the results include the cross model comparison such as AA, AB, and AC in the previous matrix (table 5.2).

There are three main takeaways from these figures (5.8, 5.9, 5.10). The first is that there is no significant difference in terms of the mean performance between any of the different input models. However, there is some difference in the variance between the different model inputs. Models 3 and 4, which have the highest number of variables, tend to have lower mean variances. This is likely due to model fitting; since there are more parameters the problem is constrained more than the others and since the number of samples is low it allows for better fitting.

The second takeaway is that in terms of the mean, Node A and Node C fall closely together for all three figures. This suggests that Node A and Node C are very similar in the propagation parameters. This makes sense because both have similar local terrain. Nodes A and C are both positioned on the same plateau and spread out by about 200 meters along a road. While Node B

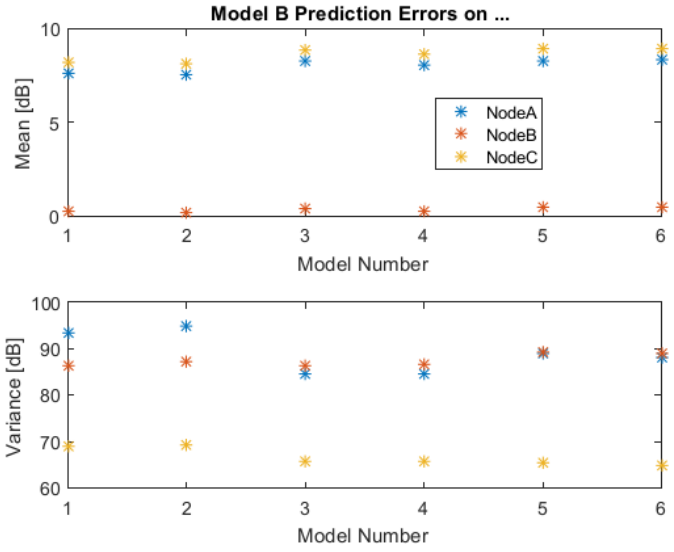

Figure 5.9: Figure depicts the error results between tested points for the six different input models from Node B and each node combination in the test matrix. The left chart shows the mean error and the right chart shows the variance of the error.

may be physically close to C, it is located at the bottom of a ridge for the plateau (approximately 20 feet vertically). This theory is further supported by the result that neither A nor C closely align with B. This result is very important because it provides insight for air to ground communication that local terrain around may play the largest role in the RF propagation of interest. In addition if there are strong similarities between multiple transmitters for the correction model, data collected for a one transmitter can be applied to others that have not been measured/learned. This could be further expanded to simplify the correction model to use the transmitter location as a function input rather than a single model for each transmitter. However, this result is larger than anticipated because RF propagations is complex and highly dependent on the paths the signal take. It is not reasonable in that input models 1 and 2 have zero inputs that are transmitter relative and are still very close to each other. A possible counter theory on this behavior is that the distribution and numbers of samples is not sufficient and it is just a coincidence that the two data sets are very

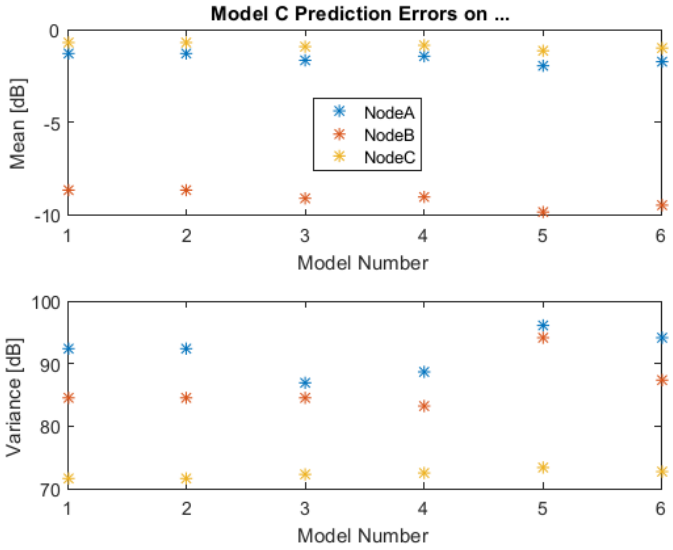

Figure 5.10: Figure depicts the error results between tested points for the six different input models from Node C and each node combination in the test matrix. The left chart shows the mean error and the right chart shows the variance of the error.

similar.

The last takeaway from the analysis is that the mean error and mean variance error are within the two bounds simulated (chapter 3) for the disturbance model. The results show that for nodes B and C when predicting on their own training data the mean errors were very close to 0 and with variance that is slightly better than the simulation results provided in Chapter 2. Only node A did not have a non zero mean and had larger variance than the other training data predictions. However the variance was still within the simulation results from earlier. Overall it supports the premise that the GP will most likely have a 0 mean prediction on average with larger variances depending on the environment and density of samples.

5.4 Data Analysis Summary

Data was collected from a series of sUAS flights in the summer of 2017 and 2018 and analyzed to investigate the performance of the a-priori model SPLAT! and the Gaussian process for the correction tool. The flights from summer of 2017 were conducted at Pawnee National Grasslands for an RF localization mission allowing for non-planned trajectories to be flown. RF mapping was performed during these missions resulting in 36 trials. These trials were used to assess the accuracy of SPLAT! and FSPL. The results of this demonstrated that SPLAT! is a magnitude more accurate that FSPL and had close precision. In addition these trials showed that the shape of the errors in a SPLAT! prediction is between that of the linear and normal disturbance models used in simulation. The July flight consisted of two missions flying lawn mower patterns to map the path loss from three transmitter beacons. The data points collected in the two missions were used to train and test the GP regression algorithm to evaluate the performance. Six different GP input models were constructed for each of the transmitters and were tested across transmitters for cross transmitter associations. The results from this flight analysis indicated that there were no major improvement between the various input models for the GP. The results also demonstrated an association between to he nodes that had similar local terrain profiles. This association is conflicting and needs further investigation in the future.

Chapter 6

Conclusion

This dissertation discussed the relevance and importance of creating communication-aware robotics, especially for sUAS. This work expands on previous work to create an architecture for a hybrid RF propagation model that is online, robust, and adaptable. The hybrid online architecture combines an a-priori estimate of the communication field with an online learning correction to update and fix deviations in the a-priori model from in-situ measurements. The architecture was assessed using simulations to understand the limitations and performance of the correction tool used in this approach. The architecture was then applied to an airborne wireless relaying application between two cooperative ground nodes. Lastly, flight data was collected and analyzed allowing an evaluation of the a-priori model SPLAT! and the Gaussian process correction tool.

6.1 Findings Summary

In the simulation assessments of the architecture it was found that the hybrid architecture may perform better when finding the deviation of the a-priori model than when learning the entire path loss field. The mean error of these predictions when compared to the nature runs had a mean error within 2 dBs of zero. However, the variance illuminated the fact that at many points the predicted path loss was substantially different than the simulated truth. While the sample number allows for better refinement of the path loss field's shape, too many may result in over-fitting causing an increase in the variance of the error. Lastly, it was demonstrated that strong correlation between points in the field leads to better learning performance by the Gaussian process.

The simulations performed for the wireless airborne relaying application provided information about how the architecture could perform when providing the path loss information to an application. The results of these simulations showed that the base hybrid architecture provides a small degree of exploration-exploitation behavior when used with simple guidance algorithms and kinematic models. This result highlighted the fact that there is more information that can be used from the hybrid architecture that could be leveraged to develop the desired level of exploration or exploitation behavior for a given application. In the simulations it was also shown that on average there was no significant difference between the various a-priori models used for the simple terrain environment. This would likely change depending on the environment being tested and the performance of the a-priori model may become more important. It is noted that the hybrid architecture works as well as using previous flight data and is at least as good as a naive approach on average in this application.

The data collected from the various flights provided information about SPLAT! and the Gaussian process regression. The data used to evaluate SPLAT! from Pawnee National Grasslands showed that SPLAT! was an order of magnitude more accurate and precise than Free Space Path Loss. It also showed that SPLAT! tended to have zero mean error with some variance. The maps of these errors depicted shapes that were more closely aligned with a linear disturbance model than a normal disturbance model with some radial aspects. The flight data that was used to evaluate the Gaussian process found two main points. The first is that for the given sample set the type of inputs provided did not significantly change the performance of the prediction. The second is that there can be correlation between similar transmitter sources that could provide interesting ways of expanding the use of data collected during the architectures use when trying to map several different RF fields at the same time from various transmitters. The exact correlation is unknown as the data set is likely disrupted by happenstance similarities.

In conclusion, the work in this thesis shows that the hybrid architecture can successfully learn the error of the a-priori model and can be used to provide information to a communication-aware application. It is also determined that the hybrid architecture in some cases is better able to learn

than the direct method.

6.2 Future Work

There are several aspects that were not addressed by this research or that need additional attention after these findings.

One aspect that was not investigated is utilizing the whole Gaussian distribution output at a given point from the GP regression. In the wireless relaying application only the mean predicted field was used for planning purposes, however it is expected that by utilizing the predicted variance in conjunction with the mean more advanced planning algorithms could be used to improve the performance through the exploration and exploitation properties. For example, the utility function could include a term that incorporates the lack of information in an area to do more exploration.

Another aspect was that the architecture's evaluation was not fully completed with validation from flight tests. The algorithms and software needed to perform these tests were created but due to weather and flight crew availability the flights have been postponed. Part of these validation flight tests include multiple mission scenarios of the wireless relaying application.

One area that this research did not focus on is the complexity of the terrain. While initially planned, logistical, legal, and technical issues arose when discussing flying missions in the mountain regions of Colorado. The areas of operation that were available for flying with CU's COAs were not satisfactory for the pilots with a fixed wing aircraft from a weather and obstacle perspective. While this thesis was able to demonstrate the effectiveness of the architecture in simple environments, it is expected that this architecture will be very beneficial in more complicated environments. In addition there does not appear to be sufficient literature about mountainous terrain communication models and data.

Lastly, more application research is needed to demonstrate the adaptability of the algorithm. There appears to be potential use for this architecture being implemented in flight safety software for sUAS such as Unmanned Traffic Management (UTM) by utilizing the architecture to provide a C2 link geofence that prevents the aircraft from going beyond reasonable limits of communication.

There is ongoing work in the RECUV Lab at CU Boulder using this algorithm as a way of providing communication information for multi-UAS operations.

The architecture defined in this work has the potential to advance the communication aspect of sUAS applications. The architecture is more robust and adaptable than other online methods presented but still needs to be further expanded on.

Bibliography

[1] Darryl Jenkins and Bijan Vasigh. The Economic Impact of Unmanned Aircraft Systems Integration in the United States. Association for Unmanned Vehicle Systems International, (March):1–40, 2013.

[2] Brian Ferris, Dirk Hähnel, and Dieter Fox. Gaussian Processes for Signal Strength-Based Location Estimation. 2006.

[3] S R Dixon, C D Wickens, and Dervon C Chang. Comparing quantitative model predictions to experimental data in multiple-UAV flight control. Proceedings of the Human Factors and Ergonomics Society 47th Annual Meeting, pages 104–108, 2003.

[4] Yasamin Mostofi. Communication-aware motion planning in fading environments. Proceedings - IEEE International Conference on Robotics and Automation, pages 3169–3174, 2008.

[5] Fumie Ono, Hideki Ochiai, Kenichi Takizawa, Mikio Suzuki, and Ryu Miura. Two-Way Relay Networks Using Unmanned Aircraft Systems. 2013 IEEE 77th Vehicular Technology Conference (VTC Spring), pages 1–5, 2013.

[6] Chris A. Wargo, Gary C. Church, Jason Glaneueski, and Mark Strout. Unmanned Aircraft Systems (UAS) research and future analysis. IEEE Aerospace Conference Proceedings, 2014.

[7] Yong Zeng, Rui Zhang, and Teng Joon Lim. Wireless communications with unmanned aerial vehicles: Opportunities and challenges. IEEE Communications Magazine, 54(5), 2016.

[8] Lav Gupta, Raj Jain, and Gabor Vaszkun. Survey of Important Issues in UAV Communication Networks. IEEE Communications Surveys and Tutorials, 18(2), 2016.

[9] Sebastian Rohde, Markus Putzke, and Christian Wietfeld. Ad hoc self-healing of OFDMA networks using UAV-based relays. Ad Hoc Networks, 11(7), 2013.

[10] Ilker Bekmezci, Ozgur Koray Sahingoz, and amil Temel. Flying Ad-Hoc Networks (FANETs): A survey, 2013.

[11] Yiannis Kantaros and Michael M Zavlanos. Communication-Aware Coverage Control for Robotic Sensor Networks. 2014.

[12] Philip B.' Charlesworth. Networked Multiple UAS. Encyclopedia of Aerospace Engineering, pages 1–9, 2015.

[13] Kamesh Namuduri, Yan Wan, and Mahadevan Gomathisankaran. Mobile ad hoc networks in the sky. Proceedings of the second ACM MobiHoc workshop on Airborne networks and communications - ANC '13, page 25, 2013.

[14] Ozgur Koray Sahingoz. Mobile networking with UAVs: Opportunities and challenges. 2013 International Conference on Unmanned Aircraft Systems, ICUAS 2013 - Conference Proceedings, pages 933–941, 2013.

[15] David Shallcross. Autonomous Location of Mobile Wireless Relay Nodes. pages 15–24, 2016.

[16] Illya Stepanov and Kurt Rothermel. Simulating mobile ad hoc networks in city scenarios. Computer Communications, 30(7):1466–1475, 2007.

[17] Jerome Le Ny, Alejandro Ribeiro, and George J. Pappas. Adaptive communication-constrained deployment of unmanned vehicle systems. IEEE Journal on Selected Areas in Communications, 30(5):923–934, 2012.

[18] C.R. Dixon and E.W. Frew. Cooperative electronic chaining using small unmanned aircraft. Collection of Technical Papers - 2007 AIAA InfoTech at Aerospace Conference, 1(May):339–346, 2007.

[19] D Satyajeet and A R Deshmukh. Heterogeneous Approaches for Cluster based Routing Protocol in Vehicular Ad Hoc Network (VANET). 134(12):1–8, 2016.

[20] Kai Daniel, Sebastian Rohde, Niklas Goddemeier, and Christian Wietfeld. A communication aware steering strategy avoiding self-separation of flying robot swarms. 2010 IEEE International Conference on Intelligent Systems, IS 2010 - Proceedings, pages 254–259, 2010.

[21] Caleb Phillips, Douglas Sicker, and Dirk Grunwald. A survey of wireless path loss prediction and coverage mapping methods. IEEE Communications Surveys and Tutorials, 15(1):255–270, 2013.

[22] U S Marine Corps. Mountain Combat: Hard to move, hard to shoot, even harder to communicate by LTC (R) Lester W. Grau USA and LT Jason Falivene USMC. pages 1–6.

[23] Julius Goldhirsh and Wolfhard J Vogel. Propagation effects for land mobile satellite systems: Overview of experimental and modeling results. NASA STI/Recon Technical Report N, 92(February):20404, 1992.

[24] Aleksandar Neskovic, Natasa Neskovic, and George Paunovic. Modern approaches in modeling of mobile radio systems propagation environment. IEEE Communications Surveys & Tutorials, 3(3):2–12, 2000.

[25] Stylianos Kasampalis, Pavlos I. Lazaridis, Zaharias D. Zaharis, Aristotelis Bizopoulos, Spyridon Zettas, and John Cosmas. Comparison of ITM and ITWOM propagation models for DVB-T coverage prediction. IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, BMSB, pages 1–4, 2013.

[26] Caleb Phillips, Douglas Sicker, and Dirk Grunwald. Bounding the practical error of path loss models. International Journal of Antennas and Propagation, 2012, 2012.

[27] Ashok M. Kanthe, Dina Simunic, and Ramjee Prasad. Effects of Propagation Models on AODV in Mobile Ad-hoc Networks. Wireless Personal Communications, 79(1):389–403, 2014.

[28] Cory Dixon and Eric W. Frew. Maintaining optimal communication chains in robotic sensor networks using mobility control. Mobile Networks and Applications, 14(3 SPEC. ISS.):281–291, 2009.

[29] Maciej Stachura and Eric W. Frew. Cooperative Target Localization with a Communication-Aware Unmanned Aircraft System. Journal of Guidance, Control, and Dynamics, 34(5):1352–1362, 2011.

[30] Hyukseong Kwon and Daniel J. Pack. Cooperative target localization by multiple unmanned aircraft systems using sensor fusion quality. Optimization Letters, 6(8):1707–1717, 2012.

[31] Esten Ingar Grøtli and Tor Arne Johansen. Path Planning for UAVs Under Communication Constraints Using SPLAT! and MILP. Journal of Intelligent & Robotic Systems, 65(1-4):265–282, jan 2012.

[32] Usman Ali, Hong Cai, Yasamin Mostofi, and Yorai Wardi. Motion and communication co-optimization with path planning and online channel prediction. Proceedings of the American Control Conference, 2016-July:7079–7084, 2016.

[33] Junting Chen and David Gesbert. Local Map-assisted Positioning for Flying Wireless Relays. 2018.

[34] Boris Galkin, Jacek Kibiłda, and Luiz A. DaSilva. Deployment of UAV-mounted access points according to spatial user locations in two-tier cellular networks. IFIP Wireless Days, 2016-April, 2016.

[35] Pawel Ladosz, Hyondong Oh, and Wen Hua Chen. Optimal positioning of communication relay unmanned aerial vehicles in urban environments. 2016 International Conference on Unmanned Aircraft Systems, ICUAS 2016, pages 1140–1147, 2016.

[36] Torsten Andre, Karin Anna Hummel, Angela P. Schoellig, Evsen Yanmaz, Mahdi Asadpour, Christian Bettstetter, Pasquale Grippa, Hermann Hellwagner, Stephan Sand, and Siwei Zhang. Application-driven design of aerial communication networks. IEEE Communications Magazine, 52(5), 2014.

[37] Mehrzad Malmirchegini and Yasamin Mostofi. On the spatial predictability of communication channels. IEEE Transactions on Wireless Communications, 11(3):964–978, 2012.

[38] Akram Al-Hourani, Sithamparanathan Kandeepan, and Abbas Jamalipour. Modeling Air-to-Ground Path Loss for Low Altitude Platforms in Urban Environments Modeling Air-to-Ground Path Loss for Low Altitude Platforms in Urban Environments. In Symposium on Selected Areas in Communication: Globecom, number December, pages 2898–2904. IEEE, 2014.

[39] Yuan Yan and Yasamin Mostofi. Communication and Path Planning Strategies of a Robotic Coverage Operation. American Control Conference (ACC), 2013, 2013.

[40] Jonathan Fink and Vijay Kumar. Online methods for radio signal mapping with mobile robots. In Proceedings - IEEE International Conference on Robotics and Automation, 2010.

[41] Magnus Minnema Lindhé. Communication-Aware Motion Planning for Mobile Robots. 2012.

[42] Neeti Wagle and Eric Frew. Spatio-temporal characterization of airborne radio frequency environments. 2011 IEEE GLOBECOM Workshops, GC Wkshps 2011, pages 1269–1273, 2011.

[43] Anthony J. Carfang, Neeti Wagle, and Eric W. Frew. Improving data ferrying by iteratively learning the radio frequency environment. IEEE International Conference on Intelligent Robots and Systems, (Iros):1182–1188, 2014.

[44] John Magliacane. SPLAT! Because the world isn't flat!, 1997.

[45] Carl Rasmussen and Christopher Williams. Gaussian Processes for Machine Learning. MIT Press, 2006.

[46] Tapan K. Sarkar, Zhong Ji, Kyungjung Kim, Abdellatif Medouri, and Magdalena Salazar-Palma. A Survey of Various Propagation Models for Mobile Communication. IEEE Antennas and Propagation Magazine, 45(3):51–82, 2003.

[47] J B Andersen and T S Rappaport. Propagation measurements and models for wireless communications channels. IEEE Communications Magazine, 33(1):42–49, 1995.

[48] National Instruments, Prentice Hall, P T R Rf, Digital Communications, Communications Resource, Main Page, Statistical Models, Multipath Fading Channels, Flat Fading, Gans Fading Model, Level Crossing, Fading Statistics, Rayleigh Fading Model, Valenzuela Indoor, Statistical Model, Smrcim Indoor, Outdoor Statistical Models, Relevant Ni, and Although Ossana. Statistical Models for Multipath Fading Channels. pages 1–11, 2016.

[49] Theodore Rappaport. Wireless Communications: Principles and Practice. Prentice Hall, 2nd edition, 2001.

[50] Eric Frew, Cory Dixon, Steve Borenstein, Katherine Glasheen, Ramya Kanlapuli Rajasekaran, Spencer Watza, and Andrew B. Mills. Lessons Learned from Field Testing Swarming Unmanned Aircraft at the University of Colorado. In AUVSI Xponential, Denver, CO, 2018.

[51] Joe Bardwell. Converting Signal Strength Percentage to dBm Values. WildPackets, Inc, (November):1–12, 2002.

[52] Constantine A. Balanis. Antenna Theory: Analysis and Design. Wiley, 4th edition, 2016.

[53] Randal W Beard and Timothy McLain. Small Unmanned Aircraft: Thoery and Practice. Princto University Press, 2012.

Appendix A

Simulated Aircraft Model

In Beard and McLain's book [53] they provide an in depth discussion about the control systems and algorithms that might run in a sUAS. The following equations were used from this text for the aircraft simulation.

The one angle (1A) kinematics model utilizes the holonomic behavior of fixed wing aircraft. The aircrafts state is primarily determined by the course angle. The 1A model seven variable state is defined in table A.1 in a NED frame.

The 1A kinematic model inputs are defined in A.2. The derivatives for the first two states, position, are simple trigonometry calculations with the course angle and airspeed. The other three inputs come from simplified linear equations using tuned parameters of the aircraft dynamic and the commanded values. No wind was assumed in the following lists.

The parameters in this work for the aircraft are specified from the Ttwistor shown in figure A.1. The parameters are listed in table A.3 with the corresponding values.

The state derivatives are only a function of the inputs and states allowing for integration

State	Description
p_n	position North
p_e	position East
h	altitude
$\dot{h}$	climb rate
χ	heading angle
$\dot{\chi}$	heading angle rate
V_a	airspeed

Table A.1: Table of the One Angle kinematic model's states

Model Inputs
$\dot{p_n} = V_a * \cos(\chi)$
$\dot{p_e} = V_a * \sin(\chi)$
$\ddot{h} = -\dot{b_h}\dot{h} + b_h * (h_{cmd} - h)$
$\ddot{\chi} = \dot{b_\chi}(\dot{\chi_{cmd}} - \dot{\chi}) + b_\chi(\chi_{cmd} - \chi)$
$\dot{V_a} = b_{Va}(V_{a-cmd} - V_a)$

Table A.2: Table of the One Angle kinematic model inputs

Figure A.1: The TTwistor Unmanned Aircraft that CU Boulder has flown for a variety of scientific missions

Parameter	Value
$b_{\dot{\chi}}$	0.5
b_χ	0.1
$a_{\dot{h}}$	1.0
$b_{\dot{h}}$	0.011
b_h	0.00115
b_{V_a}	0.01

Table A.3: Table of the One Angle kinematic model parameters for the Ttwistor unmanned aircraft

by whatever method desired. The simulations conduced utilized a simple Runge-Kutta (RK4) approach. Two different guidance algorithms were used in the simulations. A heading guidance algorithm to steer the aircraft to point to the next waypoint, used for the lawn mower patterns. Second a simple guidance loiter algorithm to get the aircraft to circle a point, used in the wireless relay application. The heading algorithm is not listed as it is trivial. The loiter algorithm used is presented MATLAB code below.

```
k_orbit = 2;
 lambda = 1; %CCW or CW
 m = 1;
 radius = waypoint.radius;


 cn = waypoint.x;
 ce = waypoint.y;
 pn = aircraft_state(1);
 pe = aircraft_state(2);


 chi = aircraft_state(5);


 psi = atan2(pe-ce,pn-cn) + 2 *pi * m;
 d = sqrt((pn-cn)^2+(pe-ce)^2);
 while(psi - chi) < -pi
     psi = psi + 2*pi;
 end
 while(psi-chi) > pi
     psi = psi - 2*pi;
 end
 chi_c = psi + lambda*(pi/2 + atan(k_orbit*(d-radius)/radius));
 Va_c = Va;
 h_c = waypoint.h;
```

```
chi_rate_c = Va/radius;

hdot_c = 0;
```

Appendix B

Flight Experiment Software

The python code for hardware/software in the loop simulation and flight experiments is hosted on two different github projects; */swatza/PyUAS* and /swatza/CommAware. The CommAware maintains the main applications for running the architecture while PyUAS provides a library to do support operations for simulation, autopilot interface, message handling, and many other parts.