ABSTRACT

Title of dissertation: Rotorcraft Flight Dynamics and Control in Wind for Autonomous Sampling of Spatiotemporal Processes

Nitin Sydney, Doctor of Philosophy, 2015

Dissertation directed by: Professor Derek A. Paley
Department of Aerospace Engineering

In recent years, there has been significant effort put into the design and use small, autonomous, multi-agent, aerial teams for a variety of military and commercial applications. In particular, small multi-rotor systems have been shown to be especially useful for carrying sensors as they have the ability to rapidly transit between locations as well as hover in place. This dissertation seeks to use multi-agent teams of autonomous rotorcraft to sample spatiotemporal fields in windy conditions. For many sampling objectives, there is the problem of how to accomplish the sampling objective in the presence of strong wind fields caused by external means or by other rotorcraft flying in close proximity. This dissertation develops several flight control strategies for both wind compensation, using nonlinear control techniques, and wind avoidance, using artificial potential-based control. To showcase the utility of teams of unmanned rotorcraft for spatiotemporal sampling, optimal algorithms are developed for two sampling objectives: (1) sampling continuous spatiotemporal fields modeled as Gaussian processes, and (2) optimal motion planning for coordinated target detection, which is an example of a discrete spatiotemporal field. All

algorithms are tested in simulation and several are tested in a motion capture based experimental testbed.

ROTORCRAFT FLIGHT DYNAMICS AND CONTROL IN WIND FOR AUTONOMOUS SAMPLING OF SPATIOTEMPORAL PROCESSES

by

Nitin Sydney

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2015

Advisory Committee:
Professor Derek Paley, Chair/Advisor
Professor Inderjit Chopra
Professor Sean Humbert
Professor Raymond Sedwick
Professor Benjamin Shapiro

Dedication

To my family

Acknowledgments

Firstly, I would like to thank my advisor, Professor Derek A. Paley, for guiding my research through my graduate career and for granting me the opportunity to pursue my research interests. The experiences I have gained by working in the Collective Dynamics and Control Laboratory have been invaluable.

I would also like to thank the other members of my dissertation committee, Professors Inderjit Chopra, Sean Humbert, Raymond Sedwick, and Benjamin Shapiro, for their help in shaping my dissertation.

My thanks go to my fellow CDCL lab mates, namely Levi DeVries, Seth Napora, Sachit Butail, Cammy Peterson, Amanda Chicoli, Frank Lagor, and Daigo Shishika for helping me throughout the years. This dissertation would not be here today without their constant discussions and expertise.

The last few years I have spent at the Naval Research Laboratory have introduced me to several people that I would like to thank, namely Don Sofge, Keith Sullivan, and Michael Kuhlman, who have all helped support my developing interest in the field of mobile robotics.

My family has been a constant source of strength and support for me during this process. I would like to thank my parents, Rekha and Mukesh, for their unrelenting belief in what I can achieve; I wouldn't be who I am without them. I would especially like to thank my brother Anish. He's pushed me to work harder than I ever thought I could, and he truly deserves to be called the better twin because of it.

Finally, I would like to thank my wife, Lauren. She's put up with all my late nights, listened patiently to all my problems, and has provided constant love and support for the last eight years. Thank you for everything.

Table of Contents

List of Figures

Chapter 1: Introduction

1.1 Motivation

In the past decade, there has been growing interest in the use of unmanned aerial vehicles for both government [1] and commercial applications [2]. Small unmanned aerial vehicles present new opportunities for multi-agent sensor deployment as they can transit rapidly between locations. Rotorcraft in particular can be used to hover sensors in areas of interest and can redeploy based on collected information. The ability to rapidly transit opens new possibilities for information gathering and multi-agent coordination. As such, this dissertation explores the possibility of using multiple, small, unmanned rotorcraft for the purpose of sampling spatiotemporal processes.

Figure 1.1 shows a flowchart of how spatiotemporal fields are represented and how multi-agent teams of unmanned rotorcraft can be used to estimate them. Spatiotemporal processes are split into two categories: continuous and discrete processes. Examples of continuous processes include scalar fields such as temperature, humidity, pollution, etc. Conversely, discrete processes encompass applications such as target tracking and moving vehicle surveillance. The methods for estimating these processes generally fall into one of two categories: interpolation and proba-

bilistic. Interpolation includes strategies such as Gaussian process and least squares estimation, whereas probabilistic strategies include distributed parameter estimation and recursive Bayesian inference. With regards to using unmanned rotorcraft for sampling spatiotemporal processes, there are two levels of control. First is a path planning level. In this level, vehicles coordinate motions to accomplish a desired global goal, e.g., to maximize information gain. The second level is rigid-body motion, i.e., the ability to follow the motion prescribed by the path planner.

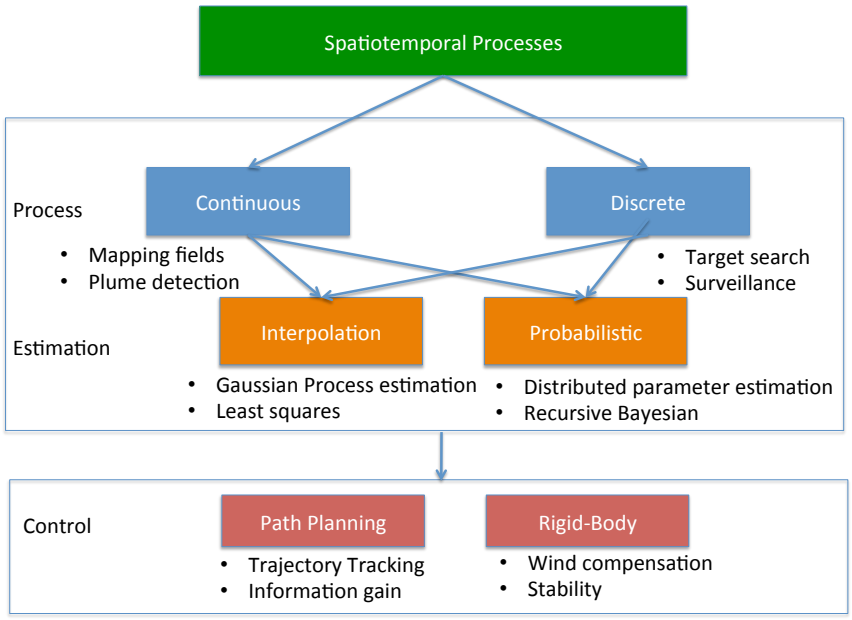


Figure 1.1: Representation of estimation of spatiotemporal fields for the purpose of sampling and estimation using multi-agent teams of unmanned rotorcraft.

One contribution of this dissertation is dealing with the issue of rigid-body control of the rotorcraft. One of the concerns with small unmanned aerial vehicles such as quadrotors is their susceptibility to wind fields and gusts, which can not only degrade the performance of the vehicle but can also make them dangerous to

operate in populated areas. Even without external wind gusts, quadrotors operating in proximity to one another are affected by the downwash from other quadrotors. Therefore, controllers are designed to stabilize rotorcraft in the presence of strong wind fields caused both by external means as well as other vehicles in close proximity.

The focus of this dissertation is to showcase how rotorcraft can be used in real-world, windy environments for sampling spatiotemporal processes. As such, another contribution of this dissertation is to design path planning algorithms and control teams of rotorcraft to sample two types of spatiotemporal processes. The first type of process is scalar, spatiotemporal Gaussian processes with two spatial dimensions and one temporal dimension. Specifically, the scope of the Gaussian process sampling is to focus on optimally sampling nonstationary fields, i.e., fields whose variability is nonuniform. This dissertation also seeks to solve the explore vs. exploit problem for Gaussian processes, i.e., where the statistics of the field must be estimated before the field can be mapped. The second type of spatiotemporal process is the discrete process of target detection. In the problem of target detection, the focus is on vehicles with low quality cameras, for which vehicle coordination is crucial to optimally detect an unknown number of potentially moving targets.

1.2 Related Work

The related work is split into three sections according to subproblem. The three sections are: multi-rotor flight in wind, distributed target detection, and sampling of continuous Gaussian processes.

1.2.1 Multi-Rotor Flight in Wind

Many prior works in the area of quadrotor control, e.g., [3–5], approximate the quadrotor dynamics by a linear system, for which standard linear controllers can be designed. More recent dissertations [6–8] use nonlinear control techniques like feedback linearization, backstepping, and sliding mode control. It is common for linear and nonlinear control techniques applied in the literature to use a six-degree-of-freedom (6DOF) model for the quadrotor dynamics that neglects the effects of aerodynamic forces on the vehicle. Some notable exceptions [9,10] model rotor effects such as blade flapping and induced thrust, which are important to the results in this dissertation. Other papers [11, 12] account for wind gust disturbances, but only as linear perturbations.

The use of flow sensing has been employed with great success on naturally evolved flyers [13]. Using distributed measurements of relative wind, sufficiently large creatures such as birds align themselves with their intended direction of travel and account for the effects of wind gusts. By detecting fine details of the ambient flow field, small creatures like insects improve their flight performance by finely tuning their flap stroke to suit flight conditions [13]. Gewecke and Woike [14] showed that directing airflow over avian feathers could cause steering impulses, and as shown in more recent work by Brown and Fedde [15], birds have the necessary sensor-feather mechanisms in the wing to predict stall and measure airspeed.

In contrast to natural flyers, the current paradigm of small unmanned system instrumentation is to integrate inertial measurements supplemented by (scalar) air-

speed. A five-hole probe providing air-data measurements that include airspeed, angle of attack, and sideslip has been successful in applications involving conventional fixed wing flight within the traditional flight envelope [16–18]. These platforms provide a baseline capability for more advanced tests in areas such as cooperative control [19] and ocean-borne operations [20] for both fixed-wing and rotary-wing vehicles [21, 22].

Seeking improved platform performance, researchers have looked to expand the notion of onboard flow sensing and apply it to various levels of vehicle control. For example, flow information can be used to fine-tune aerodynamic parameters for performance. Patel and Corke [23, 24] considered the time-domain response from a high-bandwidth pressure sensor to predict incipient flow separation at the wing leading edge and trigger a plasma flow actuator to alleviate flow separation.

Flow sensing can also improve flight control. Xu et al. [25] implemented arrays of micro-machined shear-stress sensors on the leading edge of a low aspect-ratio delta wing. The sensor system was developed to support control strategies that effected aerodynamic flight control through boundary-layer manipulation [26]. The AVOCET project [27] aims to continuously tailor the pressure distribution and resulting forces and moments across the wing using advanced micro-tuft sensors and hybrid fluidic flow actuators. Under attached flow conditions, NASA has supported wind tunnel-based implementation and testing of a distributed actuation and sensing array for use on a blended-wing-body UAV, using a series of pressure measurements to study the effectiveness of a morphing-wing control strategy [28, 29]. More recently, Watkins et al. have demonstrated improved attitude control of a small fixed-wing

vehicle in turbulent flow conditions through the integration of pressure sensors embedded in the main wings.

1.2.2 Distributed Target Detection

There are many approaches to multiple target tracking in the literature. Probabilistic approaches such as [30–36] use Bayesian inference combined with maximum likelihood or multiple hypothesis trackers to track multiple (moving) targets. Although the above approaches address how to combine sensor measurements to track and detect targets, only a few techniques [37–39] effectively move the mobile platforms to find targets and most of these techniques focus on the use of a single sensor platform. Other approaches [40] focus on optimal sensor selection. Recently, several authors, e.g., [41, 42], have used information-based approaches to design optimal sensor placement/control, but it is difficult to scale these approaches to a large number of vehicles due to the complexity in calculating mutual information. Most similar to the work presented here are physics-inspired swarm controllers [43–45], where vehicles switch behaviors based on their "temperature" to improve the speed and quality of information gathering. However, the behavior in these approaches is explicitly imposed, leading to a large number of tuning parameters. Additionally, these strategies tend to be heuristic, and hence may not have performance guarantees.

1.2.3 Sampling of Continuous Gaussian Processes

There are several approaches in the literature pertaining to sampling a spatiotemporal field using a mobile sensor network. In distributed parameter estimation, the field is modeled by an underlying process, usually a partial differential equation, for which the parameters are estimated by routing vehicles to maximize an objective function based on optimal experimental design [46–48]. In Bayesian estimation [49–52], the mean of the field is modeled as a combination of known basis functions and a (partially) known covariance function. A maximum-likelihood, recursive (gridded or particle) filter is used to determine the unknown coefficients of the covariance function as well as regression coefficients for the basis functions. In the coverage control approach [53–57], vehicles are routed to maximize the area that the sensor measurements cover, often using heuristic measures for coverage. Another approach is based on mapping error [58, 59], where vehicles are routed to minimize the error in the estimation of the field.

Unlike prior work that focuses on sampling stationary spatiotemporal fields, e.g., [58, 60, 61]; this dissertation studies the problem of sampling a spatiotemporal field in which the second-order statistics may change as a function of space and time, i.e., the field is nonstationary [62]. Some recent works on sampling a nonstationary field focus on coverage control [63, 64], where vehicles are controlled such that there are no gaps in the coverage of the field; these works use a heuristic measure of sampling need to measure sampling performance. For example, Leonard and Olshevskyi in [64] sample a time-invariant field by routing vehicles to fixed locations. Sampson

et al. and Schmidt *et al.* estimate the covariance structure of a time-invariant non-stationary field using fixed measurement locations as in [65,66]. In [65], a coordinate transformation is used to ensure the covariance structure of the transformed field is stationary. The covariance function is parameterized by its characteristic spatial and temporal length scales, called decorrelation scales. The approach in [65] is to numerically map the locations of several fixed sampling stations to a new set of coordinates such that the spatial dispersion between locations in the new coordinates is approximately stationary. The new coordinates are found by numerically minimizing a cost function that includes a heuristic measure of stationarity. While not identical, this approach strongly influenced the development of the main result for Gaussian process estimation, which extends the approach to permit mobile sensors.

1.3 Contributions

The contributions of this dissertation are as follows:

1. Created a layered, feedback linearization controller for a 6DOF quadrotor model including the effects of blade flapping and aerodynamic moment. Unlike other models in the field, this controller allows the wind field to be different over each rotor, which can cause severe and undesired motions of the vehicle. The controller uses an estimated wind field found using a recursive Bayesian filter in the feedback loop.

2. Designed and implemented a path planner that estimated the position of another quadrotor and avoids the downwash. The path planner uses an estimate

of the position of another quadrotor as input to the path planner, which uses artificial potential field control to guide the vehicle safely around the impinging downwash. The estimate of the proximal quadrotor is found using a recursive Bayesian estimator.

3. Created a feedback linearization controller for a 1DOF pitch stand that includes blade flapping and aerodynamic drag. Unlike the simplified momentum theory based models used for the quadrotor controller, this controller uses an aerodynamic model where motor RPM is the input. This allows for a finer level of control, which may be required in strong wind fields.

4. Developed physics-inspired multi-sensor path planner for detecting an unknown number of targets. The path planner uses only local information for guidance, and hence scales linearly with the number of vehicles in the field.

5. Proved that the physics-inspired path planner locally maximizes the expected information gain. By assuming a binary sensor model, it is shown that locally moving vehicles along the gradient of a Likelihood Ratio Surface maximizes the mutual information between the collected measurement and likely target location.

6. Designed optimal control for sampling along a closed path in a stationary field. The control design is based on the concept of *sampling speed*, where the statistics of the field are used to determine how to space vehicles to provide full coverage. The control design is a speed control on the path to acheive a

splay formation amongst vehicles.

7. Developed a coordinate transformation to render a nonstationary field locally stationary. The transformation assumes a Gaussian covariance function between space-time locations. This coordinate transformation is crucial to vehicle path planning design as it allows for the use of coverage algorithms based on sampling stationary fields, for with the sampling need is uniform.

8. Created a path planner that maximizes sensor coverage of a two dimensional nonstationary field. The path planner the Spanning Tree Coverage algorithm to create a path through the domain that concentrates measurements in areas where there is a higher sampling need. The control is then a combination of path following plus speed control to maintain a splay formation.

9. Developed a strategy for sampling a stationary field with unknown decorrelation scales. The strategy for sampling uses the concept of explore versus exploit. Vehicles first travel in a formation that is beneficial for determining the decorrelation scales (using a recursive Bayesian estimator), then transition into an STC based mapping formation.

1.4 Outline of Dissertation

The rest of the dissertation is outline as follows. Chapter 2 introduces several mathematical frameworks that are necessary for the understanding of the contributions made in the remaining chapters. This includes a detailed discussion of

recursive Bayesian estimation and likelihood ratio tracking, estimation of Gaussian processes, as well as an introduction to nonlinear feedback linearization.

Chapter 3 designs several controllers for flight stability for multi-rotor vehicles in the presence of unknown wind fields. This includes the development of both stability controllers, as well as a path planning strategy for flight near other multi-rotor vehicles.

Chapter 4 presents a framework for using aerial vehicles with low-quality downward facing cameras to detect an unknown number of potentially moving targets. In this chapter, it is shown that the physics-inspired coordination strategy that is presented is locally optimal in the sense of maximizing information gain.

Chapter 5 develops the theory for sampling stationary and nonstationary spatiotemporal Gaussian processes. It details how to sample one dimensional and two dimensional spatial fields with both known and unknown decorrelation scales.

Chapter 6 summarizes the dissertation and provides suggestions for future work.

Chapter 2: Background Information

In this chapter, I discuss three mathematical frameworks used in this dissertation. The first is recursive Bayesian estimation, which is a probabilistic parameter estimation technique. The second is the estimation of Gaussian processes. Gaussian processes are used to model continuous spatiotemporal field in Chapter 5. The final framework is feedback linearization, which is a nonlinear control design technique that is used to globally linearize nonlinear systems. This technique is used in Chapter 3 to control the dynamics of multi-rotor vehicles.

2.1 Recursive Bayesian Estimation

A Bayes filter is a probabilistic approach for assimilating noisy measurements into a probability density function over a given state space. The filter implementation comprises the discrete steps of predicting and updating. Let θ_k denote the state to be estimated at time step k and z_k denote the measurement at time k. The predict step involves computing the conditional probability [32]

$$p(\theta_k|z_{k-1}) = \int_\Omega p(\theta_k|\theta_{k-1})p(\theta_{k-1}|z_{k-1})d\theta_{k-1}. \tag{2.1}$$

The measurement update is proportional to the product of the measurement likelihood $p(z_k|\theta_k)$ and the predicted state [32]:

$$p(\theta_k|z_k) = \frac{p(z_k|\theta_k)p(\theta_k|z_{k-1})}{p(z_k|z_{k-1})}, \tag{2.2}$$

where

$$p(z_k|z_{k-1}) = \int_\Omega p(z_k|\theta_k)p(\theta_k|z_{k-1})d\theta_k$$

is the integral of the numerator. $p(z_k|z_{k-1})$ is referred to as the prior, i.e., the information known from previous time steps, and $p(\theta_k|z_k)$ is the posterior, which takes into account information provided by the likelihood function. The posterior at time step k is used as the *prior* at time step $k+1$, and thus, (2.2) can be applied recursively to estimate θ. Note that measurements from multiple sensors are assimilated by executing multiple, independent update steps. The importance of the Bayesian filter for the work in this dissertation is that it allows for nonlinear state dynamics and measurement equations.

For the work in Chapter 4, a likelihood ratio tracker is used to detect targets from unmanned rotorcraft. A likelihood ratio tracker is exactly like the recursive Bayesian filter except the measurement likelihood function is replaced by the measurement likelihood ratio, i.e., the ratio of two likelihood functions. The numerator of the likelihood ratio represents the conditional probability of the measurement given that the target is in state θ_k^+, whereas the denominator represents the conditional probability of the measurement given that the target is not in state θ_k^-. Thus,

the log likelihood ratio is

$$\log \mathcal{L}(z_k|\theta_k) = \log \frac{p(z_k|\theta_k^+)}{p(z_k|\theta_k^-)} \tag{2.3}$$

$$= \log(p(z_k|\theta_k^+)) - \log(p(z_k|\theta_k^-)). \tag{2.4}$$

Let $P = \log(p)$. The update step in the log likelihood ratio tracker becomes

$$P(\theta_k|z_k) = \log \frac{\mathcal{L}(z_k|\theta_k)p(\theta_k|z_{k-1})}{p(z_k|z_{k-1})}$$

$$= P(z_k|\theta_k^+) - P(z_k|\theta_k^-) + P(\theta_k|z_{k-1}) - P(z_k|z_{k-1}). \tag{2.5}$$

The first term in (2.5) represents the new, positive information, whereas the second term represents the new, negative information. The third term represents the prior information and the fourth term is a normalization constant. When the probability reaches a critical threshold, the target is declared detected; otherwise, the sub-threshold target probabilities are maintained as hypotheses. Note, the inverse log likelihood ratio posterior represented by temperature in Chapter 4 is $-P(\theta_k|z_k)$.

2.2 Estimation of Gaussian Processes

Let $A(r_k)$ be the value of an unknown, continuous spatiotemporal field at $r_k \triangleq [x_k,\ y_k,\ t_k]^T \in \mathbb{R}^2 \times \mathbb{R}^+$, which is the space-time location of point k. (The subscript on r is dropped to indicate the entire field.) The field is statistically described by its mean $\bar{A}(r_k)$ and the covariance function $C(r_i, r_j)$ between any two points r_i and r_j. A covariance function is a positive-definite function that describes the variability of the field between the i^{th} and j^{th} location, as described in [67]. A field is stationary if its covariance function depends only on the difference $r_i - r_j$ and is non-

stationary if it depends on r_i and r_j independently. There are a number of choices for the form of $C(r_i, r_j)$, e.g., Matern, rational quadratic, Ornstein-Uhlenbeck and squared-exponential forms as described in [62]. For the sampling strategy in this dissertation, a covariance function that is a product of a spatial covariance function and a temporal covariance function is required, i.e., $C(r_i, r_j) = C_s(x_i, y_i, x_j, y_j)C_t(t_i, t_j)$, where C_s and C_t are the spatial and temporal portions of the covariance, which depend on space and time, respectively.

To represent spatiotemporal fields with non-uniform coverage requirements, a nonstationary squared exponential covariance function of the following form is adopted, as introduced in [62]:

$$C(r_i, r_j) = \frac{|\Sigma(r_i)|^{1/4}|\Sigma(r_j)|^{1/4}}{\left|\frac{\Sigma(r_i)+\Sigma(r_j)}{2}\right|^{1/2}} \exp\left[-\frac{1}{2}(r_i - r_j)^T \left(\frac{\Sigma(r_i) + \Sigma(r_j)}{2}\right)^{-1} (r_i - r_j)\right], \tag{2.6}$$

where $\Sigma(r_k) \in \mathbb{R}^{3\times 3}$ is a positive definite symmetric matrix that is continuous in r_k. I require that $\Sigma(r_k) = \text{diag}\{\sigma_x^2(x_k, y_k), \sigma_y^2(x_k, y_k), \tau^2(t_k)\}$. The square roots of the diagonal elements of $\Sigma(r_k)$ are the spatial and temporal decorrelation scales of the field. The decorrelation scales dictate the spatial and temporal separations between the i^{th} and j^{th} locations at which the covariance function evaluates to $1/e \approx 0.368$ and are the characteristic scales of the field. Note, for a stationary field, the decorrelation scales are constant, but for a nonstationary field they may vary in space and time. The Gaussian covariance function (2.6) is used to derive the coordinate transformation in Section 5.3, which clusters measurements in space-time regions with shorter decorrelation scales, and spreads measurements where

the decorrelation scales are larger. Other covariance functions, e.g., Matern, may contain these decorrelation scales, but they are embedded in a different function, e.g., a Bessel function. The extension of the coverage strategy to the case where $\Sigma(r_k)$ is not diagonal is not considered here.

To determine the mapping error, optimal interpolation is employed as in [68–70], which treats the field as a discrete random field in space and time. Let $\tilde{r}_d = [\tilde{x}_d,\ \tilde{y}_d,\ \tilde{t}_d]^T$ be the space-time location of measurement $d = 1, ..., D$ and ϵ_d be the measurement noise, so that the value of measurement d is $z_d = A(\tilde{r}_d) + \epsilon_d$. It is assumed that $E[\epsilon_m \epsilon_l] = \tilde{\sigma}_0 \delta_{ml}$, where $E[\cdot]$ denotes the expected value, $\tilde{\sigma}_0$ is the standard deviation of the measurement noise, and δ_{ml} is the Kronecker delta, which implies that the noise from any two distinct measurements is uncorrelated and the variance of the noise is $\tilde{\sigma}_0$.

Let C_e denote the covariance of the error in the estimate $\hat{A}(r)$ of the field $A(r)$ after assimilating the set of measurements $\tilde{r} = [\tilde{r}_1, \ldots, \tilde{r}_D]^T$. I have

$$C_e(r_i, r_j; \tilde{r}) = C(r_i, r_j) - \sum_{d=1}^{D} \sum_{l=1}^{D} C(r_i, \tilde{r}_d)(M^{-1})_{dl} C(r_j, \tilde{r}_l), \tag{2.7}$$

as explained in [69, 71], where M^{-1} is the inverse of the measurement covariance matrix whose entries are

$$M_{dl} = E[z_d z_l] = C(\tilde{r}_d, \tilde{r}_l) + \tilde{\sigma}_0 \delta_{dl},$$

as from [72]. The mapping error is the diagonal $C_e(r_i, r_i; \tilde{r})$ of the error covariance matrix. The average (resp. maximum) mapping error is computed by averaging (resp. finding the maximum of) all of the elements of the mapping error. The

(mean-square) optimally interpolated estimate $\hat{A}(r_i)$ of $A(r_i)$ is

$$\hat{A}(r_i) = \bar{A}(r_i) + \sum_{d=1}^{D} C(r_i, \tilde{r}_d) \left(\sum_{l=1}^{D} (M^{-1})_{dl} z_d \right),$$

as provided in [67], where $\bar{A}(r_i)$ is the assumed known mean (first-order statistics) of the field.

2.3 Feedback Linearization

In this dissertation I use input/ouput feedback linearization of the nonlinear dynamics of the quadrotor and a one degree of freedom pitch stand [73]. Let $\mathbf{x} \in \mathbb{R}^n$ be the state vector of the quadrotor, $\mathbf{u} \in \mathbb{R}^m$ be the control, and $\mathbf{y} \in \mathbb{R}^m$ be the output. Consider (nonlinear) system dynamics of the form

$$\dot{\mathbf{x}} = f(\mathbf{x}) + \sum_{i=1}^{m} g_i(\mathbf{x}) u_i \tag{2.8}$$

$$\mathbf{y} = [y_1, ..., y_m], \tag{2.9}$$

where $f, g \in \mathbb{R}^n$. The feedback linearization technique described next allows us to design $\mathbf{u}$ to linearize (2.9) using state feedback control.

Let r_k be the relative degree of the kth output, where $k = 1, ..., m$, and let $L_f^k(y_i(\mathbf{x}))$ denote the kth Lie derivative of $y_i(\mathbf{x})$ with respect to f [73]. A control that linearizes the system (2.8)–(2.9) is [73]

$$\mathbf{u} = \Delta^{-1}(\mathbf{x})(\boldsymbol{\nu} - b(\mathbf{x})), \tag{2.10}$$

where

$$\Delta_{ij}(\mathbf{x}) = L_{g_j} L_f^{r_i - 1} y_i(\mathbf{x}), \text{ and } b_i(\mathbf{x}) = L_f^{r_i - 1} y_i(\mathbf{x}). \tag{2.11}$$

$\Delta(\mathbf{x})$ is the decoupling matrix for the system and the (linear) control $\boldsymbol{\nu}$ can be designed to achieve the desired closed-loop behavior of the new system [73]

$$\begin{bmatrix} \dfrac{d^{r_1}y}{dt^{r_1}} \\ \vdots \\ \dfrac{d^{r_m}y}{dt^{r_m}} \end{bmatrix} = \begin{bmatrix} \nu_1 \\ \vdots \\ \nu_m \end{bmatrix}.$$

Note that $\mathbf{u}$ requires the inverse $\Delta^{-1}(\mathbf{x})$ of the decoupling matrix, which means that $\Delta(\mathbf{x})$ must be invertible for all $\mathbf{x} \in \mathbb{R}^n$ in order for (2.8) to be feedback linearizable.

2.4 Wind Field Modeling

I adopt an approach [74] to model a separated flow behind the ship using a complex potential in a simplified domain, called the Z-domain. The flow in the Z-domain contains elementary potentials, such as uniform flows, to represent ambient motion and vortices to represent separation. This flow is transformed to the physical domain, called the z-domain, using a conformal mapping designed so that vortices in the Z-domain are placed after sharp bends or in notches in the z-domain. Note that without vortices, this technique would not produce separated flow in the z-domain.

Locations in the Z-domain are represented in complex coordinates as $Z = X + iY$. Adding the potential of a vortex pair and a uniform flow yields

$$\Omega(Z) = \frac{U_\infty}{2\pi}\left\{Z - p - i\frac{(p+1)^2 + s^2}{2s}\log\left[\frac{Z-p+is}{Z-p-is}\right]\right\}, \tag{2.12}$$

where $p \pm is$ are the locations of the vortex centers. Since the vortices are of equal and opposite strength, there is a virtual surface on the real line through which flow

cannot pass. The uniform flow represents the ambient flow and the vortex at $p+is$ produces the separated flow in the z-domain.

Let $z = F(Z)$ be a conformal map from the Z-domain to the z-domain. The mapping $F(Z)$ takes the real line and shapes it to whatever the physical domain looks like. For example, for a backwards step, which is an approximation to the geometry of the deck of a ship suitable for landing an unmanned aerial system, the map $F(Z)$ is [74]

$$z = \frac{h}{\pi}\{\sqrt{Z^2-1} + \log[Z + \sqrt{Z^2-1}]\}. \tag{2.13}$$

From [74], the magnitude q and direction θ of the flow at location Z are

$$q = \mathrm{Re}\left\{\log\left[\frac{d\Omega}{dZ}\left(\frac{dF}{dZ}\right)^{-1}\right]\right\} \tag{2.14}$$

$$\theta = -\mathrm{Im}\left\{\log\left[\frac{d\Omega}{dZ}\left(\frac{dF}{dZ}\right)^{-1}\right]\right\}. \tag{2.15}$$

For the backwards step [74]

$$\frac{d\Omega}{dz} = \frac{(Z-2p-1)\sqrt{Z+1}\sqrt{Z-1}}{(Z-p+is)(Z-p-is)}. \tag{2.16}$$

Using (2.14) and (2.16) the flow magnitude and direction are [74]

$$q = \sqrt{\frac{(X-2p-1)^2+Y^2}{[(X-p)^2+(Y+s)^2][(X-p)^2+(Y-s)^2]}} \times \left([(X+1)^2+Y^2][(X-1)^2+Y^2]\right)^{1/4} \tag{2.17}$$

$$\theta = \tan^{-1}\left(\tfrac{Y+s}{X-p}\right) + \tan^{-1}\left(\tfrac{Y-s}{X-p}\right) - \tfrac{1}{2}\tan^{-1}\left(\tfrac{Y}{X-1}\right) + -\tan^{-1}\left(\tfrac{Y}{X-2p-1}\right) - \tfrac{1}{2}\tan^{-1}\left(\tfrac{Y}{X+1}\right). \tag{2.18}$$

Figure 2.1 illustrates the flowfield in the z-domain generated by this transformation. Assuming that the complex plane represents the horizontal (x) and vertical (z)

directions, the mean wind field is

$$w_{x,m} = q\cos\theta \tag{2.19}$$

$$w_{y,m} = 0 \tag{2.20}$$

$$w_{z,m} = q\sin\theta. \tag{2.21}$$

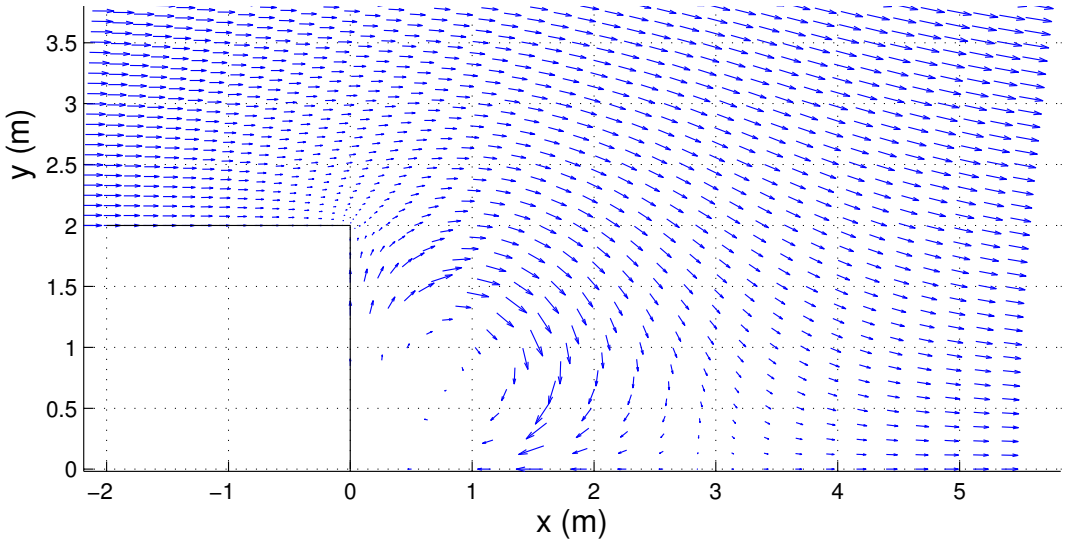


Figure 2.1: Flowfield in the z-domain

For the application of proximity flight, an expression for the downwash of a quadrotor based on [75] momentum theory derived here to describe the downwash of each rotor. Let v_i be the induced velocity of a rotor, which is the vertical velocity directly underneath it. Then, the vertical velocity w_c at any point z below the rotor height z_r is [75]

$$w_c = v_i + v_i \tanh\left(-k\frac{z_r - z}{h}\right), \tag{2.22}$$

where k and h are shaping parameters that control how rapidly the area of the streamtube below the rotor contracts to its steady-state value. Momentum theory

[76] assumes that w_c is uniform over the xy plane for a given z position. However, the boundary of the downwash contracts as it speeds up, so the radial condition for when the vertical velocity goes to zero can be identified. Let ρ be the radial distance of the downwash boundary from the rotor center $(x_r,\, y_r,\, z_r)$ and R be the radius of the rotor. Using momentum theory [74]

$$w_{z,m} = \begin{cases} w_c & \text{if } \rho < R/\sqrt{1+\tanh\left(-k\frac{z_r-z}{h}\right)} \\ 0 & \text{otherwise.} \end{cases} \tag{2.23}$$

To get the full flowfield for all four rotors the flowfield generated by all the rotors are added together. Figure 2.2 shows an example of the quadrotor downwash with $v_i = 4$ m/s, where negligible or zero velocity vectors have been omitted.

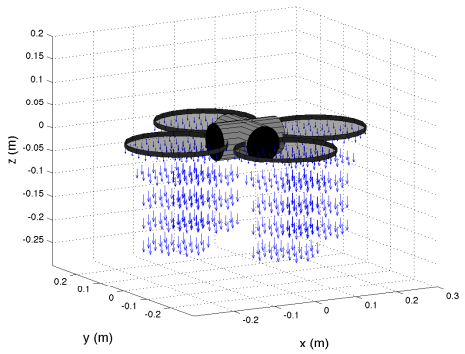


Figure 2.2: Flow beneath a quadrotor using (2.23) as the downwash model.

Note that it is assumed that the downwash has no radial component of velocity. The justification for this assumption lies in the observation that the downwash tends to contract to its steady-state condition within 1–2 rotor radii, after which the flowfield is primarily vertical [76]. Thus, the model is valid only for proximity flight with separation greater than 1–2 rotor radii.

Since the wind fields described above are smooth, which is unrealistic for the air wake behind a ship or the downwash of a quadrotor, a frozen (spatially constant) Dryden gust model is added to the mean wind field to simulate turbulence. The turbulent component of $w_{(\cdot)}$ is [77]

$$w_{(\cdot),t} = \sum_{n=1}^{N} \sqrt{\Delta\omega \Phi(\omega_n)} \sin(\omega_n t + \phi_{(\cdot),n}), \tag{2.24}$$

where ω_n is the frequency of the nth mode, $\Delta\omega$ is the spacing of the frequencies, $\phi_{(\cdot),n}$ is a random phase shift, and

$$\Phi(\omega_n) = \sigma_t^2 \frac{2L_t}{\pi} \frac{1}{1 + (L_t \omega_n)^2}$$

is the power spectral density for a Dryden gust model [77]. Thus, the total wind field is

$$w_{(\cdot)} = w_{(\cdot),m} + \sum_{n=1}^{N} \sqrt{\Delta\omega \Phi(\omega_n)} \sin(\omega_n t + \phi_{(\cdot),n}), \tag{2.25}$$

where the phase shifts are different for each spatial dimension.

In general, the wind field is not known *a priori*, thus a recursive Bayesian filter is employed to produce an estimated wind field for use in the controller. For this, a likelihood function, which relates the measurements to the wind and vehicle state, is needed. As an example, consider a uniform wind field. Assume the wind field is parameterized by parameter set $\boldsymbol{\beta}$, e.g. wind magnitude and direction for a uniform wind. Assume the ground velocity $\mathbf{v} = \dot{x}\mathbf{e}_x + \dot{y}\mathbf{e}_y + \dot{z}\mathbf{e}_z$ is available from GPS measurements (outdoors) or motion capture (indoors) and air velocity $\mathbf{v}_{rel} = u\mathbf{b}_x + v\mathbf{b}_y + w\mathbf{b}_z$ is measured by a pitot tube array (pitot tubes pointing along all axes so relative wind can be determined in each direction). (If the GPS

updates are too slow for realtime control, a combination of GPS and an IMU could be used to determine position.) The expected air velocity measurement is

$$\mathbf{v}_{rel,e}(\boldsymbol{\beta}, \mathbf{v}) \triangleq \begin{bmatrix} u_e \\ v_e \\ w_e \end{bmatrix} = {}^{\mathcal{I}}R^{\mathcal{B}} \begin{bmatrix} w_{x,m}(\boldsymbol{\beta}) - \dot{x} \\ w_{y,m}(\boldsymbol{\beta}) - \dot{y} \\ w_{z,m}(\boldsymbol{\beta}) - \dot{z} \end{bmatrix}, \tag{2.26}$$

where the vehicle ground speed is defined as the wind velocity plus the vehicle velocity relative to the wind. It is assumed that the pitot tubes are mounted orthogonally to the rotation of the vehicle such that vehicle rotation does not induce a velocity in the sensors. Using (2.26), the following (Gaussian) likelihood function is chosen

$$p(\mathbf{z}_i(t_k)|\boldsymbol{\beta}) = \exp\left[-(\mathbf{v}_{rel} - \mathbf{v}_{rel})^T(\mathbf{v}_{rel} - \mathbf{v}_{rel,e})/\sigma^2\right], \tag{2.27}$$

where $\mathbf{z}_i(t_k) = [\dot{x} \quad \dot{y} \quad \dot{z} \quad u \quad v \quad w]^T$ and σ^2 represents measurement noise variance. This form for the likelihood function produces the desired behavior, namely that the likelihood increases when the parameters are close to the true value, and decreases when they are not.

As an example of using (2.27) to estimate a uniform wind, Figure 2.3 shows the estimator performance for a single quadrotor. In the simulation, the quadrotor is subjected to a uniform wind of 1 m/s directed along the positive y axis ($\pi/2$ rad). Figure 2.3 shows the probability marginal distributions of wind speed (left) and direction (right) as measurements are incorporated in time. The dashed white line indicates the maximum likelihood estimate, which converges to the correct value, despite a noise level of $\sigma = 0.1$ m/s.

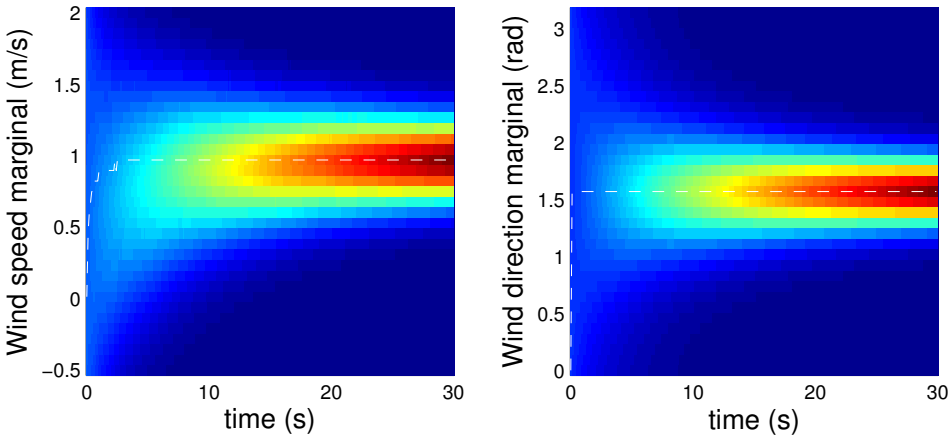


Figure 2.3: Estimation marginals of wind speed (left) and direction (right).

Chapter 3: Rotorcraft Guidance and Control in Wind

For small, unmanned rotorcraft to be viable for widespread use, it is imperative that they be a able to handle real-world, outdoor conditions. In this chapter, I discuss several strategies for operating small-scale rotorcraft in the presence of strong wind fields. Here, strong wind fields are quantified as winds that are at least of the order of the induced velocity of the rotor.

3.1 Quadrotor Dynamics and Control in Wind

This section presents a nonlinear controller for a model quadrotor flying in an unknown, turbulent wind field. In this scenario, the rotor forces are commanded to counteract the effect of an unknown windfield. I start by deriving the dynamics of the vehicle in Section 3.1.1, where I include aerodynamic loads and perturbations not normally considered in the literature. In Section 3.1.2 I derive a layered feedback linearization controller that determines the thrust for each rotor. For this controller, I assume that the desired pitch and roll angles are small. In the final section, I present a Bayesian estimator that determines an estimate of an unknown wind field given (noisy) measurements of groundspeed, airspeed, and position of the vehicle.

3.1.1 Quadrotor Dynamics in Wind

The following 6DOF model for quadrotor flight in wind includes the effects of rotor blade flapping, induced thrust due to translational flight, and aerodynamic drag. Also novel is the individual treatment of blade flapping and induced thrust on each of the rotors, which is crucial to flight in the downwash of another vehicle or in the presence of a spatially varying flowfield.

The free-body diagram of the quadrotor in Figure 3.1 shows six forces acting on the quadrotor (gravity, aerodynamic drag, and the four thrust forces) and two reference frames (the inertial reference frame $\mathcal{I}$, which is centered at origin O and has unit vectors $\mathbf{e}_x$, $\mathbf{e}_y$, and $\mathbf{e}_z$, and the body reference frame $\mathcal{B}$, centered at G with unit vectors $\mathbf{b}_x$, $\mathbf{b}_y$, and $\mathbf{b}_z$). The inertial orientation of the body frame is described using a 3-2-1 Euler angle sequence of yaw (ψ), pitch (θ), and roll (ϕ) angles.

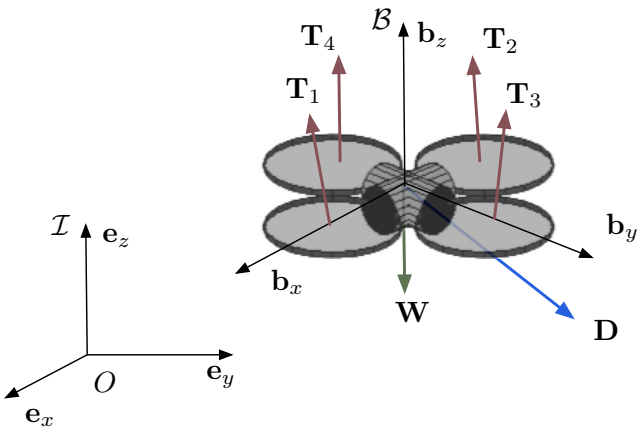


Figure 3.1: Reference frames and free body diagram for quadrotor.

Figure 3.2 shows how the thrust vector $\mathbf{T}_k$ is deflected away from the $\mathbf{b}_z$ axis

due to rotor blade flapping [76], which can be understood as follows. As the relative wind hits the rotor, the advancing side of the blade experiences increased flow and lift, whereas the retreating side sees decreased flow and lift, causing the blades to tilt due to the change in aerodynamic load. However, since the loading cycles occur at the same frequency as the rotation of the blade, there is a resonance effect that causes the maximum deflection to occur 90 degrees out of phase with the load location [76]. Thus, the rotor plane tilts away from the relative velocity vector, and redirects the thrust force.

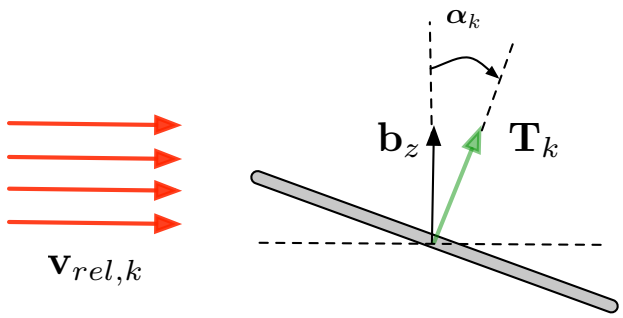


Figure 3.2: Flap angle of the rotor due to a relative wind.

The air-relative velocity incident upon the kth rotor is denoted $\mathbf{v}_{rel,k} = u_k\mathbf{b}_x + v_k\mathbf{b}_y + w_k\mathbf{b}_z$. Blade flapping occurs due to the component of $\mathbf{v}_{rel,k}$ in the body xy plane [9]. Following [78], the approximation is made that the blade flap angle α_k is proportional to the magnitude of the velocity in the body xy plane. Thus, $\alpha_k = k_f\sqrt{u^2 + v^2}$, where k_f is a proportionality constant that is common to all four rotors. To find $\mathbf{T}_k$, the geometry in Figure 3.2 with the flap angle α_k is used to

determine

$$\mathbf{T}_k = T_k(\bar{u}_k \sin\alpha_k \mathbf{b}_x + \bar{v}_k \sin\alpha_k \mathbf{b}_y + \cos\alpha_k \mathbf{b}_z), \tag{3.1}$$

where $\bar{u}_k \triangleq u_k/\sqrt{u_k^2+v_k^2}$, $\bar{v}_k \triangleq v_k/\sqrt{u_k^2+v_k^2}$, and T_k is the magnitude of the thrust of the kth rotor.

In addition to the tilt of the thrust vector due to blade flapping, I also model the change in the magnitude of the thrust due to the relative wind velocity. There are two effects that cause this change [76]: the first is an increase in thrust due to horizontal translation, which is known as induced thrust; the second is a decrease in thrust if the component of relative wind in the $\mathbf{b}_z$ direction is negative. Essentially, the rotor loses thrust when it is in climb and gains thrust in horizontal translation. Note that the tilt in the rotor plane due to blade flap is ignored, which is justified because flap angles tend to be on the order of one degree in moderate relative wind [9]. Let v_h be the induced velocity of the rotor in hover, is assumed known (either using momentum theory or experimentally). The modified thrust can be calculated using [76]

$$v_{i,k} = \frac{v_h^2}{\sqrt{u_k^2+v_k^2+(v_{i,k}+w_k)^2}} \quad \text{and} \tag{3.2}$$

$$T_k = \frac{F_k v_{i,k}}{v_{i,k}+w_k}, \tag{3.3}$$

where F_k is the thrust applied to the airmass for a given power (the control input) to the rotor. Equations (3.2)–(3.3) assume that the motor dynamics are sufficiently fast so that their transient behavior can be ignored. Note also that the equation for $v_{i,k}$ produces a fourth-order polynomial in induced velocity citehoffman. For the case of climb, there is only one positive root, which represents the physical solution

for induced velocity and can be found numerically or analytically [76]. In hover, the induced velocity is v_h, whereas in descent the momentum theory solution above is not valid as the airflow through the rotor is not steady.

An important limitation of the model (3.2)–(3.3) is that it only holds when $w_k \leq 0$ or $w_k > 2|v_h|$ [9]; otherwise, the rotor is in Vortex Ring State (VRS), which is a region in which the aerodynamics are unsteady and momentum theory results due not apply. In practice, helicopters fly quickly through VRS to avoid any dynamic instabilities. It has also been shown that the thrust tends not to vary greatly in descent [9]. Ground effect is also not modeled by (3.2)–(3.3).

It is assumed that the drag force acts at the center of mass of the vehicle and hence causes no moments on the vehicle. The drag force is [9]

$$\mathbf{D} = C_d||\mathbf{v}_G||^2\hat{\mathbf{v}}_G, \tag{3.4}$$

where C_D is a drag coefficient and $\mathbf{v}_G = u_G\mathbf{b}_x + v_G\mathbf{b}_y + w_G\mathbf{b}_z$ is the relative velocity at the center of mass (the hat $\hat{}$ denotes a unit vector). Note that (3.4) is a point-particle assumption for the drag force, which is a reasonable approximation since the other aerodynamic effects dominate the moments [9].

To get the equations of motion for the translational dynamics all of the forces are expressed in terms of the unit vectors in the inertial frame using the rotation matrix

$$^{\mathcal{I}}R^{\mathcal{B}} = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix}. \tag{3.5}$$

Let

$$C_1 = \left(C_D ||\mathbf{v}_{rel,CM}|| u_{CM} + \sum_{k=1}^{4} \bar{u}_k \sin\alpha_k \frac{F_k v_{i,k}}{v_{i,k} + w_k} \right) \tag{3.6}$$

$$C_2 = \left(C_D ||\mathbf{v}_{rel,CM}|| v_{CM} + \sum_{k=1}^{4} \bar{v}_k \sin\alpha_k \frac{F_k v_{i,k}}{v_{i,k} + w_k} \right) \tag{3.7}$$

$$C_3 = \left(C_D ||\mathbf{v}_{rel,CM}|| w_{CM} + \sum_{k=1}^{4} \cos\alpha_k \frac{F_k v_{i,k}}{v_{i,k} + w_k} \right). \tag{3.8}$$

Using (3.5) and (3.6)–(3.7), the equations of motion for translation expressed in the inertial frame are

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix}_{\mathcal{I}} = {}^{\mathcal{I}}R^{\mathcal{B}} \begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix}_{\mathcal{B}} - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}_{\mathcal{I}}. \tag{3.9}$$

To determine the rotational dynamics , the moments on the quadrotor caused by the forces in Figure 3.1 are needed. Gravity acts through the center of mass, so it does not contribute to the moments on the vehicle. (It is also assume that the drag force acts through the center of mass, so it does not generate a moment.) Thus, the only moments are caused by the thrust forces and a yawing moment to counteract the rotor torques, such that the total angular momentum about the center of mass G is conserved. Assume a symmetric geometry, where all the rotors are a horizontal distance L and vertical distance d from the center of mass, and that rotors 1 and 2 are situated along the $\mathbf{b}_x$ axis and rotors 3 and 4 are along the $\mathbf{b}_y$ axis. In order to balance the moment produced by spinning the rotors, assume that rotors 1 and 2 spin in the opposite direction of rotors 3 and 4. Note that a controlled imbalance of the rotor moments is what achieves yaw rotation in a quadrotor.

Let $\boldsymbol{\omega} \triangleq p\mathbf{b}_x + q\mathbf{b}_y + r\mathbf{b}_z$ be the angular velocity of the quadrotor body frame with respect to the inertial frame. Let $I \triangleq \text{diag}(I_x, I_y, I_z)$ is the moment of inertia matrix for the quadrotor and M_y be a yawing moment produced by spinning the motors. (Note that the up-down symmetry of the quadrotor is sufficiently small enough that the off diagonal terms of the moment of inertia matrix can be ignored.) The rotational dynamics are

$$I\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times I\boldsymbol{\omega} + \sum_{k=1}^{4} \mathbf{r}_{k/G} \times \mathbf{T}_k + M_y \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \tag{3.10}$$

where $\mathbf{r}_{k/G}$ is the distance vector from G to the kth rotor. Performing the requisite cross products yields the following rotational equations of motion:

$$\begin{aligned} I_x\dot{p} &= -(I_z - I_y)qr - \sum_{i=1}^{4} d\bar{v}_k \sin\alpha_k \frac{F_k v_{i,k}}{v_{i,k} + w_k} \\ &\quad + L(\cos\alpha_3 \frac{F_3 v_{i,3}}{v_{i,3} + w_3} - \cos\alpha_4 \frac{F_4 v_{i,4}}{v_{i,4} + w_4}) \end{aligned} \tag{3.11}$$

$$\begin{aligned} I_y\dot{q} &= -(I_x - I_z)rp - \sum_{i=1}^{4} d\bar{u} \sin\alpha_k \frac{F_k v_{i,k}}{v_{i,k} + w_k} \\ &\quad + L(\cos\alpha_2 \frac{F_2 v_{i,2}}{v_{i,2} + w_2} - \cos\alpha_1 \frac{F_1 v_{i,1}}{v_{i,1} + w_1}) \end{aligned} \tag{3.12}$$

$$\begin{aligned} I_z\dot{r} &= -(I_y - I_x)pq - L(\bar{v}_1 \sin\alpha_1 \frac{F_1 v_{i,1}}{v_{i,1} + w_1} \\ &\quad - \bar{v}_2 \sin\alpha_2 \frac{F_2 v_{i,2}}{v_{i,2} + w_2}) - L(\bar{u}_3 \sin\alpha_3 \frac{F_3 v_{i,3}}{v_{i,3} + w_3} \\ &\quad - \bar{u}_4 \sin\alpha_4 \frac{F_4 v_{i,4}}{v_{i,4} + w_4}) + c_m \sum_{k=1}^{4} F_k, \end{aligned} \tag{3.13}$$

where c_m is a constant that relates the applied thrust to the moment induced by spinning the rotor. In order to simulate the motion of the vehicle, I use the rotational

kinematics for a 3-2-1 Euler angle sequence [79]:

$$\dot{\phi} = p + q \sin\phi \tan\theta + r \cos\phi \tan\theta \tag{3.14}$$

$$\dot{\theta} = q \cos\phi - r \sin\phi \tag{3.15}$$

$$\dot{\psi} = q \sin\phi \sec\theta + r \cos\phi \sec\theta. \tag{3.16}$$

Equations (3.11)–(3.13) along with (3.9) fully describe the translational and rotational motion of the quadrotor. In the case of hover with no external wind field, these equations of motion reduce to the standard equations for a (rigid) quadrotor, e.g., [7] (omitted due to space constraints).

3.1.2 Feedback Linearization Control

I now derive an input/output feedback linearization of the quadrotor control system. First, the system is transformed to the form of (2.8) using $\mathbf{x} \triangleq [x, y, z, \dot{x}, \dot{y}, \dot{z}, \phi, \theta, \psi, p, q, r]^T$ and $\mathbf{u} \triangleq [F_1, F_2, F_3, F_4]^T$. The outputs to be controlled are the 3D position and heading (yaw) of the vehicle, i.e., $\mathbf{y} = [x, y, z, \psi]^T$. A calculation of the relative degree [73] shows that each output has relative degree two (omitted for space constraints). Since the decoupling matrix $\Delta(\mathbf{x})$ defined in Section 3.1.2 is singular for this set of outputs, I cannot determine a control $\mathbf{u}$ to make the input-output system linear using position and heading as the outputs.

As an alternative, I first feedback linearize the system with output $\mathbf{y}_{in} = [\phi, \theta, \psi, z]^T$, i.e., the attitude and altitude of the vehicle, and then feedback linearize the planar position. The first system represents the inner loop, whose inputs are the desired attitude and altitude. In this case, the relative degree for each output is two

and the decoupling matrix $\Delta(\mathbf{x})$ is invertible everywhere except when $\phi = \pi/2$ or $\theta = \pi/2$. The feedback linearized system is $\ddot{\mathbf{y}}_{in} = \boldsymbol{\nu}_{in}$. I design $\boldsymbol{\nu}_{in}$ to stabilize the system to a desired altitude and (small) attitude $\mathbf{y}_{in,d} = [\phi_d, \theta_d, \psi_d, z_d]^T$ and then transform $\boldsymbol{\nu}_{in}$ to $\mathbf{u}$ using (2.10).

The outer loop takes as input the desired (planar) position and yaw, and outputs the desired pitch and roll values to be used by the inner loop. I seek planar position dynamics of the form $\ddot{x} = \nu_{out_1}$ and $\ddot{y} = \nu_{out_2}$, where $\boldsymbol{\nu}_{out}$ is the outer loop control. Let $\phi_d = \phi_d(\nu_{out_1}, \nu_{out_2})$ and $\theta_d = \theta_d(\nu_{out_1}, \nu_{out_2})$. Assuming that the desired pitch θ_d and roll ϕ_d are small, the desired mapping (using (3.9)) is

$$\phi_d = (\sin\psi\nu_{out_1} - \cos\psi\nu_{out_2} + C_2)/C_3 \tag{3.17}$$

$$\theta_d = \frac{(\cos\psi\nu_{out_1} + \sin\psi\nu_{out_2} - C_1)}{(C_2\sin\phi_d + C_3\cos\phi_d)}. \tag{3.18}$$

The full control architecture is shown in Figure 3.3. The inputs to the closed-loop system are x_d, y_d, z_d, and ψ_d. Given x_d and y_d I calculate $\boldsymbol{\nu}_{out}$, which are the (linear) control for the outer loop. Then I use $\boldsymbol{\nu}_{out}$ to calculate ϕ_d and θ_d according to (3.17)–(3.18), which are used along with z_d and ψ_d to calculate the linear control $\boldsymbol{\nu}_{in}$ for the inner-tracking loop.

The following two-vehicle simulation illustrates the benefit of accounting for the aerodynamics of the vehicle in the feedback linearization. The vehicles are commanded using PID controllers on the inner and outer loops to a waypoint at the origin. Both vehicles start at the waypoint, and encounter a wind with a profile of $u = 2$ m/s, $v = 2$ m/s, and $w = -2(1 - \cos(x))$ m/s. The first vehicle accounts for the aerodynamic loads, whereas the second vehicle does not, i.e., it neglects the

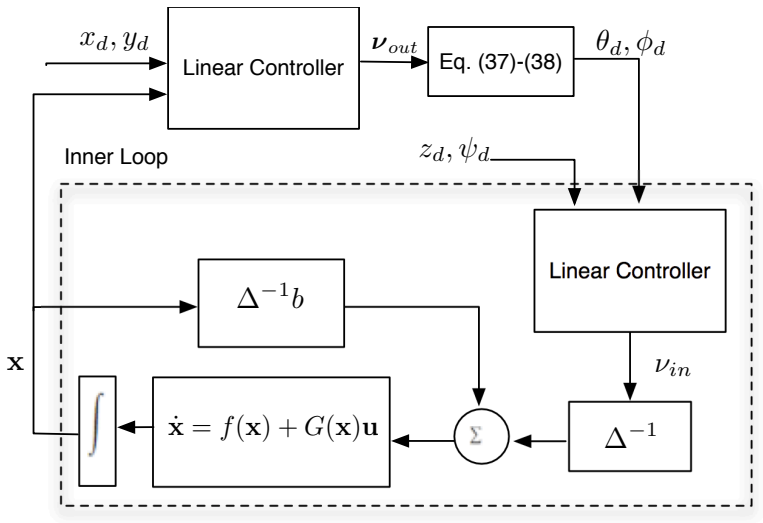


Figure 3.3: Two-layer feedback linearization control system for 6DOF quadrotor.

flap angles, drag, and the thrust modification. Figure 3.4 shows that the vehicle accounting for the aerodynamics has a lower overshoot and a smaller settling time.

3.1.3 Application: Autonomous Ship Landing

Autonomous ship landings is an area of great interest for military application [80]. The task of landing autonomously onboard a moving ship is made challenging by the separated flow that occurs on the deck of the ship, which can cause the autonomous vehicle to pitch and roll undesirably. The geometry of this situation is shown in Figure 3.5(a). Air flows over the forward section of the ship and then passes a sharp corner, where the flow separates and produces a vortex in the rear section of the ship where the quadrotor seeks to land. The flowfield parameters $\boldsymbol{\beta}$ are the speed of the prevailing wind and the location of the vortex to determine the mean (nonuniform) wind field. Frozen Dryden turbulence is added to the mean wind field

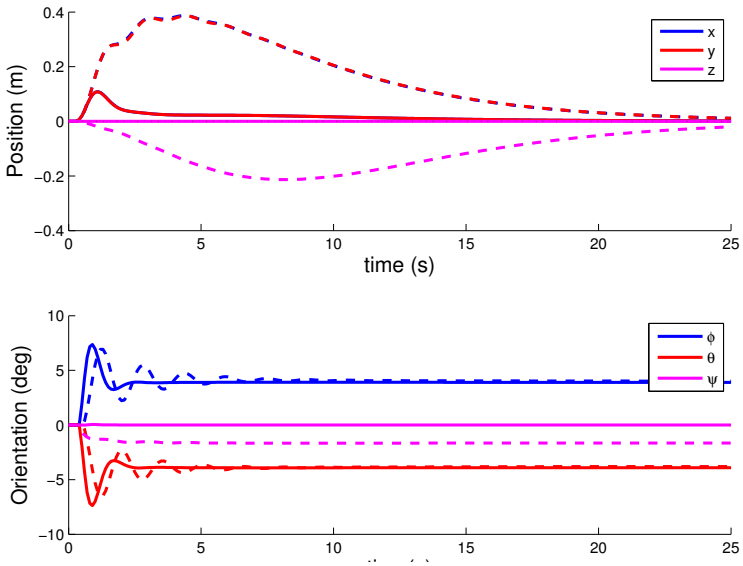


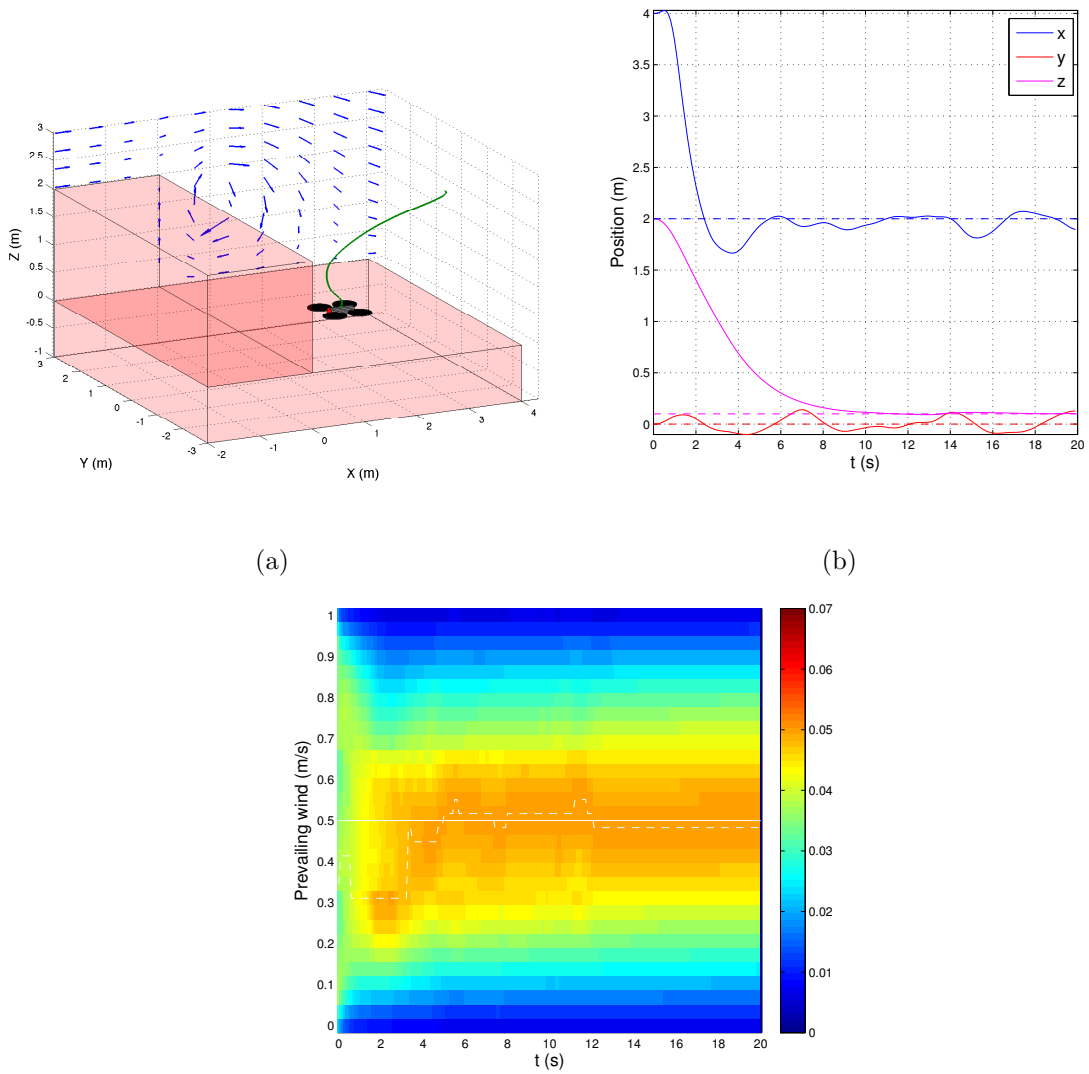
Figure 3.4: Waypoint holding with (solid) and without (dashed) accounting for aerodynamic loads.

to simulate random eddies and flow fluctuations with a range of frequencies from 1-10 rad/s and randomly generated phase shifts.

In the simulation, the vehicle is commanded to go from $(x, y, z) = (4, 0, 2)$ to a landing location of $(x_d, y_d, z_d) = (2, 0, 0)$. Figure 3.5 shows the controller performance. The colormap on the far right shows the estimation result for the wind speed. Observe that the vehicle converges to the desired landing location despite the unknown turbulence and the time it takes for the the parameter estimates to converge.

3.1.4 Application: Quadrotor Proximity Flight

In this application I show how the dynamic controller in Section 3.1 can be used to fly one quadrotor directly underneath another, e.g., for formation flight. In the simulation, one quadrotor is commanded to hover at $(x, y, z) = (0, 0, 1)$ and a



(a) (b)

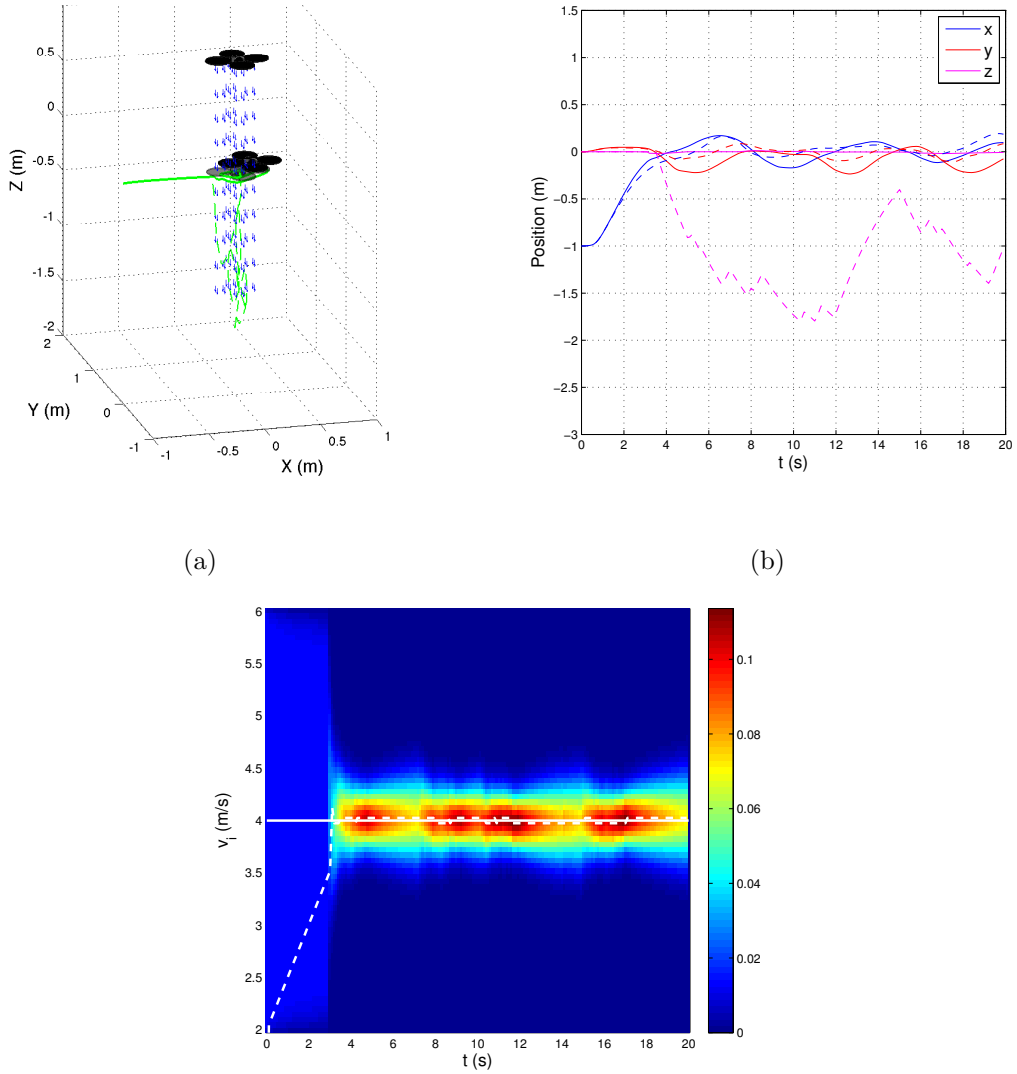
(c)

Figure 3.5: Simulation of an autonomous ship landing with unknown turbulence. (a) The vehicle trajectory in 3D; (b) vehicle trajectories vs. time; (c) probability marginal for prevailing wind speed estimation.

second quadrotor is commanded to fly underneath it to the origin. For comparison, I show a third (virtual) vehicle that does not compensate for the aerodynamics, i.e., the flap angles, drag coefficient, and climb velocity are all set to zero. In this simulation, $\boldsymbol{\beta}$ is the induced velocity of the hovering quadrotor. In addition, (frozen) Dryden turbulence affects all three vehicles. The results of the simulation are shown in Figure 3.6. The vehicle accounting for the aerodynamic loads settles at the appropriate location with no adverse affects, while estimating the induced velocity and shaping parameters of the downwash flowfield of the first vehicle. The probability density is flat for the first three seconds before the quadrotor enters the downwash, then converges quickly once it enters the field. The third vehicle, however, experiences a sharp decent rate once it hits the flowfield as it has no knowledge of the wind (dashed lines in Figure 5.2(b)). In reality, this causes a turbulent, unsteady aerodynamic condition over each rotor, which is not modeled here, and would cause the vehicle to become unstable.

3.2 Pitch Dynamics and Control

The previous sections studied the problem of flight stability for a full 6DOF quadrotor. As an intermediate step, this section develops the dynamics and formulates a feedback linearization based controller for a two-rotor system that is constrained to rotation about its pitch axis.



(a) (b)

(c)

Figure 3.6: Simulation of quadrotor proximity flight with unknown turbulence. (a) The vehicle trajectories in 3D with the shaded vehicle representing the quadrotor with no flow compensation; (b) vehicle trajectories vs. time; (c) probability marginal for induced velocity estimation.

3.2.1 Pitch Dynamics

I start by introducing several reference frames. See Figure 3.7 for a sketch of the test stand with the relevant forces and reference frames. Let $\mathcal{I} = \{O, \hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \hat{\mathbf{e}}_3\}$ be an inertial reference frame centered at the origin O. There is also a body fixed frame $\mathcal{B} = \{G, \hat{\mathbf{b}}_1, \hat{\mathbf{b}}_2, \hat{\mathbf{b}}_3\}$ that is attached to the test stand at the pivot point with the $\hat{\mathbf{b}}_1$ axis along the bar and the $\hat{\mathbf{b}}_3$ axis aligned with the propellor attachment rods. Two more reference frames, $\mathcal{C} = \{P_1, \hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2, \hat{\mathbf{c}}_3\}$ and $\mathcal{D} = \{P_2, \hat{\mathbf{d}}_1, \hat{\mathbf{d}}_2, \hat{\mathbf{d}}_3\}$, are attached to propellors 1 and 2 respectively. These last two frames are frame $\mathcal{B}$ tilted up by the flap angle of each propellor.

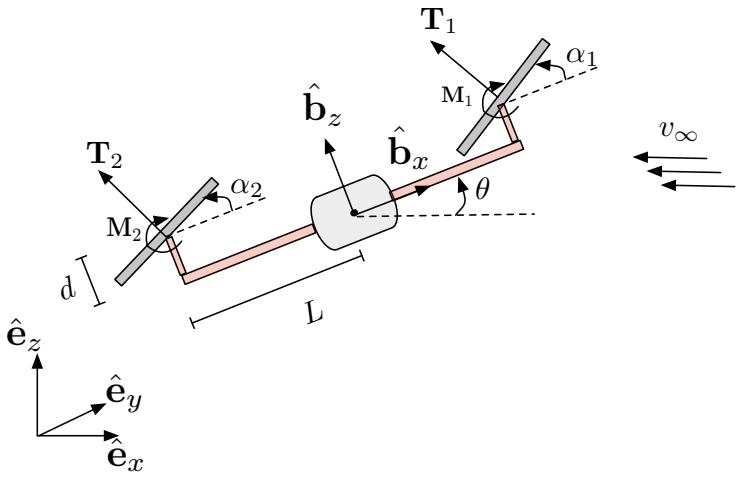


Figure 3.7: Free body diagram of pitch stand system.

Figure 3.7 shows a free body diagram of the system, where $\mathbf{T}_1$ and $\mathbf{T}_2$ are the thrust forces from the propellors, and $\mathbf{M}_1$ and $\mathbf{M}_2$ are pure moments induced by the blades as they flex due to blade flapping. The dynamics of the test stand are

determined through the relation

$$\frac{^{\mathcal{I}}d}{dt}\mathbf{h}_G = \mathbf{M}_{ext}, \tag{3.19}$$

which states that the time rate of change angular momentum of the body about the center of mass in the inertial frame is equal to the total external moment acting on the body.

To evaluate (3.19), the angular momentum of the system must be determined. Let I be the moment of inertia of the test stand and J be the moment of inertia of an individual rotor. Then, the angular momentum of the system is

$$\mathbf{h}_G = -I\dot{\theta}\hat{\mathbf{b}}_2 + J\Omega_1\hat{\mathbf{c}}_3 - J\Omega_2\hat{\mathbf{d}}_3, \tag{3.20}$$

where $\dot{\theta}$ is the pitch rate of the stand and Ω_1 and Ω_2 are the rates of revolution of the two propellors. Note that the angular velocity of the body is

$$^{\mathcal{I}}\omega^{\mathcal{B}} = -\dot{\theta}\hat{\mathbf{b}}_2. \tag{3.21}$$

Expanding the LHS of (3.19) and substitute (3.20) and (3.21) to yield

$$\begin{aligned} \frac{^{\mathcal{I}}d}{dt}\mathbf{h}_G &= \frac{^{\mathcal{B}}d}{dt}\mathbf{h}_G + {}^{\mathcal{I}}\omega^{\mathcal{B}} \times \mathbf{h}_G & (3.22)\\ &= I\ddot{\theta}\hat{\mathbf{b}}_2 + (-\dot{\theta}\hat{\mathbf{b}}_2) \times (-I\dot{\theta}\hat{\mathbf{b}}_2 + J\Omega_1\hat{\mathbf{c}}_3 - J\Omega_2\hat{\mathbf{d}}_3). & (3.23) \end{aligned}$$

To further evaluate this expression, $\hat{\mathbf{c}}_3$ and $\hat{\mathbf{d}}_3$ need to be expressed as components in the body frame. Using the flap angles α_1 and α_2 I have

$$\hat{\mathbf{c}}_3 = \cos\alpha_1\hat{\mathbf{b}}_3 - \sin\alpha_1\hat{\mathbf{b}}_1 \tag{3.24}$$

$$\hat{\mathbf{d}}_3 = \cos\alpha_2\hat{\mathbf{b}}_3 - \sin\alpha_2\hat{\mathbf{b}}_1. \tag{3.25}$$

Substituting the above expressions into (3.23) and evaluating the cross product yields

$$\frac{^{\mathcal{I}}d}{dt}\mathbf{h}_G = [-I\ddot{\theta}-\dot{\theta}(J\Omega_1\cos\alpha_1-J\Omega_2\cos\alpha_2)]\hat{\mathbf{b}}_2+\dot{\theta}(-J\Omega_1\sin\alpha_1+J\Omega_2\sin\alpha_2)\hat{\mathbf{b}}_3 \quad (3.26)$$

Making the substitution $\mathbf{M}_{ext} = M_2\hat{\mathbf{b}}_2 + M_3\hat{\mathbf{b}}_3$ yields the following two equations

$$-I\ddot{\theta} - \dot{\theta}(J\Omega_1\cos\alpha_1 - J\Omega_2\cos\alpha_2) = M_2 \quad (3.27)$$

$$\dot{\theta}(-J\Omega_1\sin\alpha_1 + J\Omega_2\sin\alpha_2) = M_3. \quad (3.28)$$

Note, equation (3.28) is not needed since it describes the yaw motion of the test stand, which is constrained to be fixed in that direction, i.e., this is a constraint that say that there is a reaction moment at the base of the test stand that has the value in (3.28).

The task now is to calculate M_2 for the system. There are four moments acting on the vehicle: two from the thrusts of each rotor and a pure bending moment caused by the structural flexing of the blades due to blade flap. As a first step, I calculate the thrust vector for each propellor. For this I need to account for the change in magnitude of the thrust do to relative climb/descent and the direction change due to forward velocity.

Let v_∞ be the magnitude of the velocity along the negative x direction (see Figure 3.7), then the speed of the flow through the rotor is

$$w_1 = v_\infty\sin(\theta+\alpha_1) + l\dot{\theta} \quad (3.29)$$

$$w_2 = v_\infty\sin(\theta+\alpha_2) - l\dot{\theta}. \quad (3.30)$$

Using momentum theory, the thrust from a propellor is

$$T_k = 2\rho A v_{i,k}(w_k + v_{i,k}), \tag{3.31}$$

where $v_{i,k}$ is the induced velocity through the kth propellor. From Blade Element Momentum Theory, the thrust is

$$T_k = k_m^{(1)}\Omega_k^2 - k_m^{(2)}(w_k + v_{i,k})\Omega_k. \tag{3.32}$$

Using equations (3.31) and (3.32), the induced velocity through the kth propellor for a given rpm can be calculated. Equation (3.32) also gives an expression that can be used for the magnitude of the thrust from the rotor as a function of rpm, which is the control input. Note that $k_m^{(1)}$ and $k_m^{(2)}$ are found experimentally.

For the direction of the thrust vector, I note that $\mathbf{T}_1$ and $\mathbf{T}_2$ are along the $\hat{\mathbf{c}}_3$ and $\hat{\mathbf{d}}_3$ axes respectively. For the flap angle, the approximation is used that the angle is proportional to the component of the relative wind parallel to the test stand, i.e.,

$$\alpha_k = k_f v_\infty \cos\theta. \tag{3.33}$$

Using the moment arms from the pivot points to the center of each rotor, the total moment due to the thrust vectors is

$$M_t = -(k_m^{(1)}\Omega_1^2 - k_m^{(2)}(w_1 + v_{i,1})\Omega_1)(l\cos\alpha_1 + d\sin\alpha_1) \tag{3.34}$$

$$+(k_m^{(1)}\Omega_2^2 - k_m^{(2)}(w_2 + v_{i,2})\Omega_2)(l\cos\alpha_2 - d\sin\alpha_2). \tag{3.35}$$

The second two moments are from the structural moment applied to the propellor hub due to the blades flexing when flapping, and is calculated as

$$M_s = -k_s(\Omega_1^2\alpha_1 + \Omega_2^2\alpha_2). \tag{3.36}$$

Substituting (3.34) and (3.36) into (3.27) and rearranging yields the equation of motion (in second order form)

$$\begin{aligned}\ddot{\theta} &= -\frac{1}{I}\Big[\dot{\theta}(J\Omega_1\cos\alpha_1 - J\Omega_2\cos\alpha_2) - (k_m^{(1)}\Omega_1^2 - k_m^{(2)}(w_1+v_{i,1})\Omega_1)(l\cos\alpha_1 + d\sin\alpha_1) + \\ &\quad (k_m^{(1)}\Omega_2^2 - k_m^{(2)}(w_2+v_{i,2})\Omega_2)(l\cos\alpha_2 - d\sin\alpha_2) - k_s\Omega_1^2\alpha_1 - k_s\Omega_2^2\alpha_2\Big]. \qquad (3.37)\end{aligned}$$

3.2.2 Control Design

To design a controller I start by simplifying (3.37) and putting it in state space form. First I assume that the rotor inertia terms are insignificant, i.e., $J = 0$. Furthermore assume the flap angle is equal for both rotors ($\alpha_1 = \alpha_2 = \alpha$) and that $d << l$. I also make a small angle assumption on α ($\cos\alpha = 1$ and $\sin\alpha = \alpha$). Then (3.37) reduces to

$$\begin{aligned}\ddot{\theta} &= -\frac{1}{I}\Big[-(k_m^{(1)}\Omega_1^2 - k_m^{(2)}l(v_\infty\sin(\theta+\alpha) + l\dot{\theta} + v_i)\Omega_1) + \\ &\quad (k_m^{(1)}\Omega_2^2 - k_m^{(2)}l(v_\infty\sin(\theta+\alpha) - l\dot{\theta} + v_i)\Omega_2) - k_s\Omega_1^2\alpha_1 - k_s\Omega_2^2\alpha_2\Big]. \qquad (3.38)\end{aligned}$$

Let $x = [x_1 \;\; x_2]^T \triangleq [\theta \;\; \dot{\theta}]^T$, then the modified equation of motion in state space form is

$$\dot{x}_1 = x_2 \qquad (3.39)$$

$$\begin{aligned}\dot{x}_2 &= -\frac{1}{I}\Big[-(k_m^{(1)}\Omega_1^2 - k_m^{(2)}l(v_\infty\sin(x_1+\alpha) + lx_2 + v_i)\Omega_1) + \\ &\quad (k_m^{(1)}\Omega_2^2 - k_m^{(2)}l(v_\infty\sin(x_1+\alpha) - lx_2 + v_i)\Omega_2) - k_s\Omega_1^2\alpha_1 - k_s\Omega_2^2\alpha_2\Big] \qquad (3.40)\end{aligned}$$

Now let $\Omega_1 = \Omega + u$ and $\Omega_2 = \Omega - u$. This effectively allows us to reduce the number of controls from two (the two rotation rates of the propellors) to one (a differential

rotation rate u). Then the state space form becomes

$$\dot{x}_1 = x_2 \tag{3.41}$$

$$\dot{x}_2 = -\frac{1}{I}\left[-2k_s\alpha u^2 + (2k_m^{(2)}l(v_\infty \sin(x_1+\alpha)+v_i) \right.$$
$$\left. -4\Omega l k_m^{(1)})u + 2l^2 k_m^{(2)}\Omega x_1 - 2k_s\alpha\Omega^2\right]. \tag{3.42}$$

Let

$$a = \frac{2}{I}(k_s\alpha)$$
$$b = -\frac{1}{I}(2k_m^{(2)}l(v_\infty \sin(x_1+\alpha)+v_i) - 4\Omega l k_m^{(1)})$$
$$c = -\frac{1}{I}(2l^2 k_m^{(2)}\Omega x_1 - 2k_s).$$

Then the control

$$u = -\frac{b}{2a} - \frac{\sqrt{b^2-4a(c-\nu)}}{2a} \tag{3.43}$$

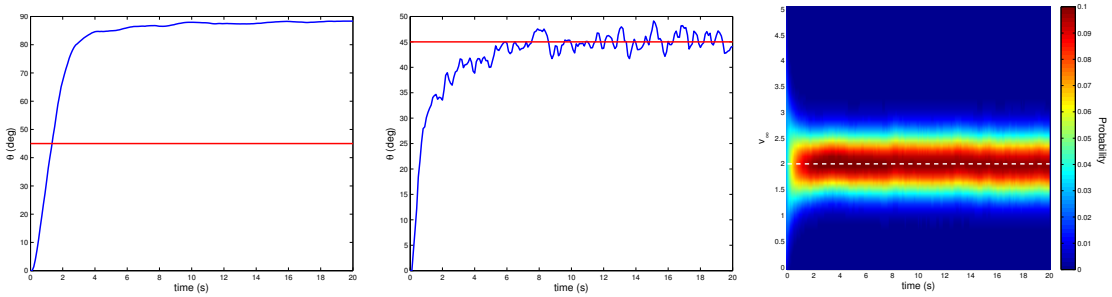
transforms the equations of motion to

$$\dot{x}_1 = x_2 \tag{3.44}$$

$$\dot{x}_2 = \nu. \tag{3.45}$$

which is a linear system.

To test the feedback system, simulations were conducted where the rotor system must stabilize to a predetermined pitch angle of $\pi/4$ in the presence of a uniform wind using a PID controller in the linearized system. In the simulation, the wind field is estimated using a recursive Bayesian filter. Without loss of generality, it is assumed that the body reference of zero pitch is the axis along which the uniform wind is acting, i.e., the wind direction is known, and only the wind magnitude must be estimated.



(a) (b) (c)

Figure 3.8: Simulation results of pitch dynamics with (a) no control enabled and (b) with feedback linearization control. (c) Results of recursive Bayesian estimator for wind speed

Figure 3.8 shows the results of the simulation. Figure 3.8(a) shows the results of the open-loop system with no feedback, for which the rotor system stabilizes to an angle of $\pi/2$, and Figure 3.8(b) shows the results of the closed loop system. The rotor-system quickly stabilizes to the reference angle, though there is a slight jitter due to simulated measurement noise. The results of the estimator with wind speed are shown in Figure 3.8(c).

3.3 Path Planning for Downwash Avoidance

The last two sections seemed to design flight controllers in the presence of unknown wind fields. However, it may be the case that it is desirable to avoid harmful wind fields all together. In multi-vehicle scenarios, such as those presented in Chapters 3 and 4, on such wind field is the downwash of other local vehicles. This section showcases the utility of a flow sensing and control system for proximity flight. The scenario involves multiple quadrotors operating at different altitudes. The

downwash of a quadrotor causes extreme, undesired changes in attitude and altitude of a quadrotor flying below [81]. The flight-path planner and flow measurement system were implemented on the lower quadrotor. The planner generates an estimate of the position of the higher quadrotor to determine the direction the lower quadrotor must travel to reach a desired waypoint while avoiding the downwash. The flow measurement system and path planner were validated in simulation and experiment.

3.3.1 Estimation of Nearby Rotorcraft

Consider a two-dimensional flow field with a rotor generating thrust along the z-axis. Velocity components w and v lie along the z and y coordinates respectively [82]. The momentum equation in the z direction is

$$\frac{\partial w}{\partial t} + w\frac{\partial w}{\partial z} + v\frac{\partial w}{\partial y} = g_z - \frac{1}{\rho}\frac{\partial p}{\partial z} + \frac{\mu}{\rho}\left(\frac{\partial^2 w}{\partial z^2} + \frac{\partial^2 w}{\partial y^2}\right) \tag{3.46}$$

where ρ is air density, μ is the dynamic viscosity of the air, and g_z represents body forces due to effects such as bouyancy. The following set of assumptions are applied as follows: (1) the mean flow field is unchanging, so $\partial w/\partial t$ is zero; although rotorwash is highly turbulent, a rotorcraft in a steady flight condition will generate a steady mean flow field velocity. (2) Cross-stream flow is small compared to the downstream velocity; this assumption is based on the intuition that even with the helical structure common to propeller-driven flow s, cross-stream velocity plays a secondary role in thrust generation. For simplicity, only the thrust-aligned velocity component is considered and $\partial w/\partial y$ is neglected. (3) At each z location, the stream-wise variation in w is small compared to the cross stream changes; hence, $\partial^2 w/\partial z^2$

is zero. (4) Lastly, bouyancy and external pressure gradients are not present, so g_z and $\partial p/\partial z$ are neglected.

Eq. (3.46) is further simplified by linearizing about a constant peak velocity W_0 based on the intuition that center-line flow velocities obey a $1/z$ decay. The result is

$$W_0 \frac{\partial w}{\partial z} = \frac{\mu}{\rho}\left(\frac{\partial^2 w}{\partial y^2}\right). \tag{3.47}$$

Solving (3.47) in cylindrical coordinates yields a Gaussian velocity profile reminiscent of the velocity variation observed in established turbulent jet profiles [83]. The expected vertical velocity measured at a lateral distance

$$r(\beta) = \sqrt{(x - x_s)^2 + (y - y_s)^2} \tag{3.48}$$

and a downstream distance z from the center of an idealized rotorcraft is

$$w(\beta) = \frac{C}{z} W_0 \exp\left(-\frac{W_0 r(\beta)^2 \rho}{4 z \mu}\right). \tag{3.49}$$

The measurement $\boldsymbol{z}^i(t_k)$ of the ith sensor at time t_k is assimilated into the Gaussian likelihood function

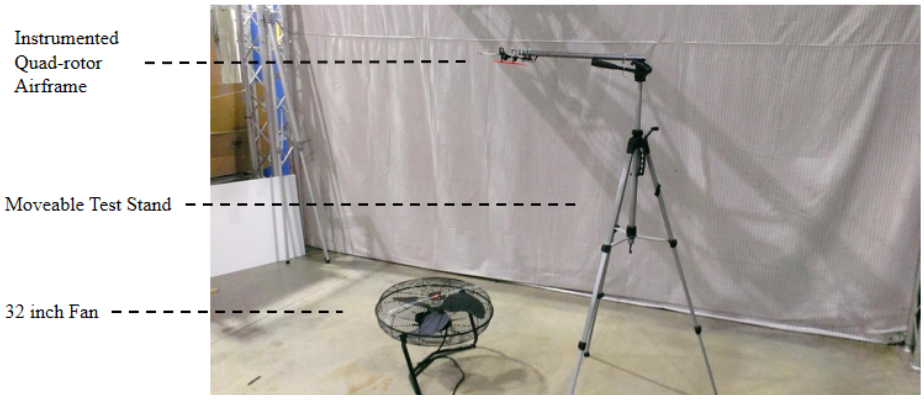
$$p(\boldsymbol{z}^i(t_k)|\beta) = \exp((z^i - w)/\sigma^2), \tag{3.50}$$

where σ^2 is the variance of the measurement noise (chosen based on sensor characteristics).

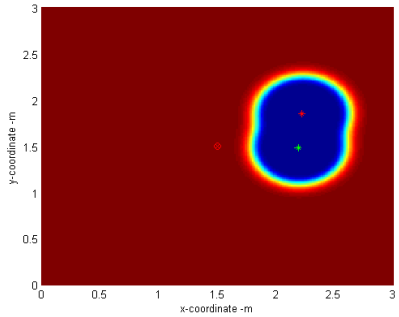
To validate the Bayesian methodology and evaluate the performance of the generalized flow model, a series of ground-based experiments were conducted. A

common 32 inch-diameter household fan, recast as a source of vertical flow disturbance, generated the flow. A test stand suspended the airspeed probes over the vertical source while data from an OptiTrack motion-capture system provided position information. To test the methodology in a more complex flow field than was modeled, experiments were conducted within the zone of flow establishment where velocity profiles have twin velocity peaks near the center line [84]. Even though this flow feature was unmodeled, a series of localization tests were successfully carried out using a single probe, followed by a set of tests with a quadrotor mockup. The quadrotor mock-up stand is shown in Fig. (3.9). Velocity measurements were taken at a number of probe locations, simulating a series of trajectories within the flow field. The spinning rotors simulated the self-noise generated in flight.

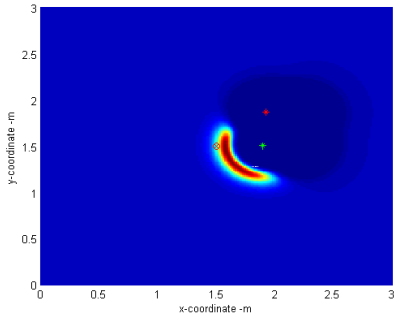
The idealized flow model and Bayesian framework provided accurate fan location estimates. The algorithm typically estimated the center of the fan to within 2 cm, or under 3% of the fan diameter. Results from a sample test run are presented in Fig. (3.9) showing estimates within 2% of fan diameter in four time steps. The colormap shows the posterior distribution of the location of the fan, with red indicating high probability and blue indicating low probability. The green and red asterisks denote probe location and the red circle indicates the actual location of the fan, unknown to the estimator.



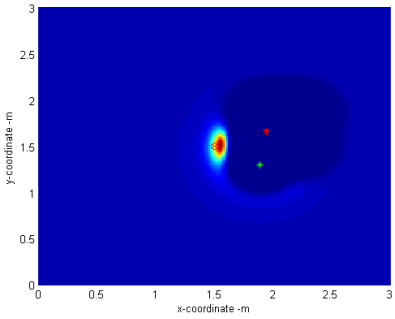
(a) Test Setup



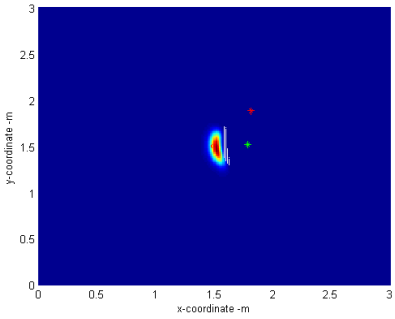
(b) Position 1



(c) Position 2



(d) Position 3



(e) Position 4

Figure 3.9: Quadcopter mock-up and sample localization results

3.3.2 Path Planning for Downwash Avoidance

Consider a vehicle with the dynamics

$$\dot{x} = u_x \tag{3.51}$$

$$\dot{y} = u_y, \tag{3.52}$$

where x and y are the coordinates the vehicle and u_x and u_y are control inputs. The goal is to design u_x and u_y to drive the vehicle to a waypoint and avoid the vertical jet produced by the higher quadrotor. The cost function $J(x, y, t_k)$ incorporates the desired goal and the presence of a vertical flow field, i.e.,

$$J(x, y, t_k) = p(\bar{\boldsymbol{z}}(t_k)|\beta) + k_J\sqrt{(x - x_d)^2 + (y - y_d)^2}, \tag{3.53}$$

where (x_d, y_d) is the location of the goal and k_J is a weighting variable. Intuitively, the cost is high when the vehicle is far away from the waypoint and/or near the downwash of the higher quadrotor. The goal is to find a path C through the domain such that the integral of J along the path is locally minimized. Formally, the problem is stated as

$$\begin{aligned} \underset{u_x,u_y}{\text{minimize}} \quad & \int_C J(x, y, t_k)ds \\ \text{subject to} \quad & \dot{x} = u_x \\ & \dot{y} = u_y, \end{aligned} \tag{3.54}$$

where ds is an increment along the path.

To decrease the computational complexity, a receding horizon version of (3.54) looks only one time step ahead. In this case the cost function reduces to

$$J_{RH} = J(x, y, t_k) + J(x + \Delta x, y + \Delta y, t_k) \tag{3.55}$$

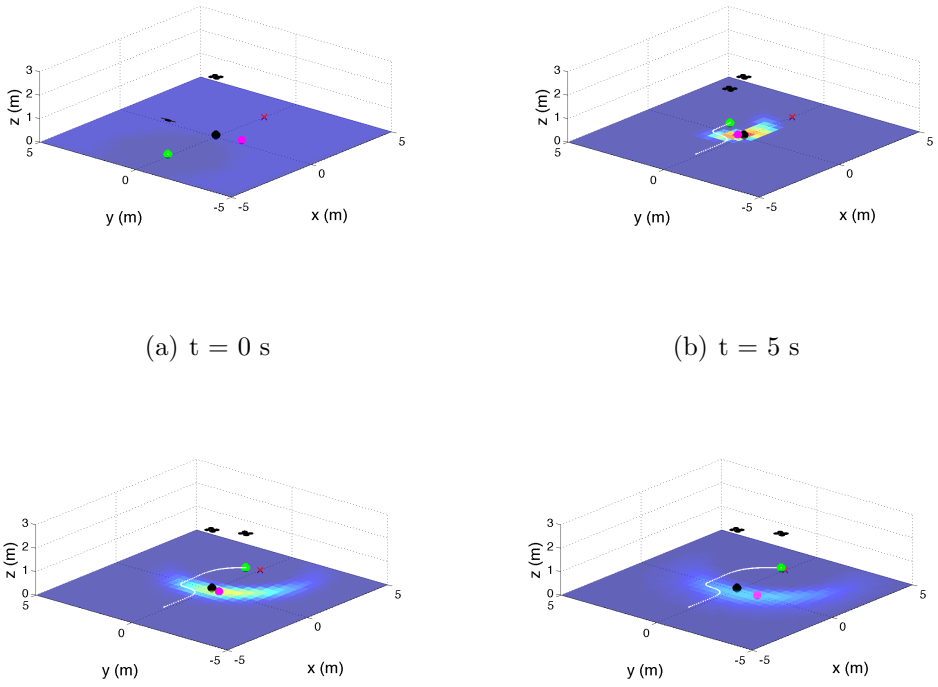
Since the first term on the right-hand side is fixed by the current vehicle location, the cost function is minimized by moving in the direction of greatest decrease. Thus, in the zero limit of Δx and Δy, (3.55) is minimized using the control

$$u_x = -K\frac{\partial J}{\partial x} \tag{3.56}$$

$$u_y = -K\frac{\partial J}{\partial y}, \tag{3.57}$$

where K is a control gain. This choice of control moves the vehicle in the direction of greatest decrease in cost.

The algorithm was first tested in simulation. The hovering quadrotor was commanded to hover at an altitude of 2.5 m, and the instrumented quadrotor was given a waypoint at (2.0, 0.0) m. Fig. 4.4 shows the results of the algorithm. Each subfigure shows a snapshot of the simulation at a different timestep, starting from the initial time until the vehicle reaches the goal. The colormap at the bottom shows the posterior distribution for the position of the hovering quad, with red and blue denoting high and low probability respectively. The black dot indicates the position of the hovering quadrotor, the green dot is the position of the instrumented quadrotor, and the red X shows the destination of the instrumented vehicle. The trajectory of the quadrotor is shown as a white line on top of the posterior distribution. The estimate of the position of the hovering quadrotor is shown as a magenta dot. The results show that the instrumented quadrotor travels towards the waypoint, then diverts once it detects the higher quadrotor using simulated flow measurement data. Note that the estimate does not achieve zero steady-state error due to noise in the sensor measurements.



(a) t = 0 s

(b) t = 5 s

(c) t = 10 s

(d) t = 15 s

Figure 3.10: Trajectory of an instrumented quadrotor in proximity flight. The colormap indicates the posterior distribution of the probability of the hovering quadrotor location.

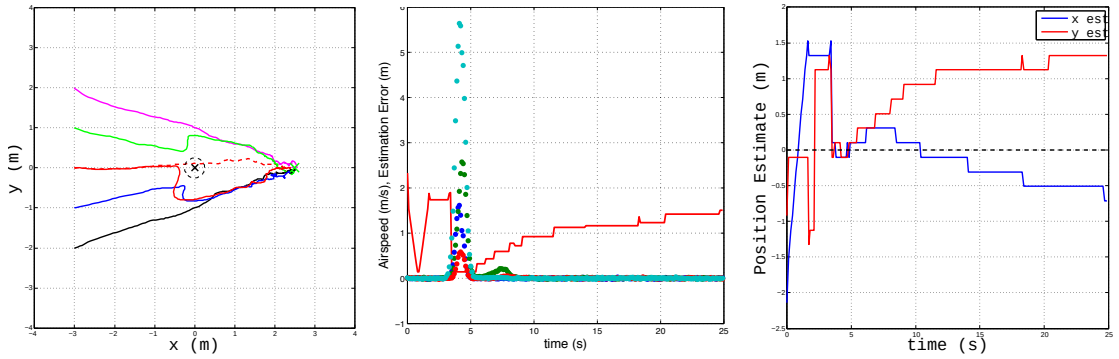
3.3.3 Experimental Results

To validate the flow measurment system and proximity flight path planner, experiments were performed at the Naval Research Laboratory in the Laboratory for Autonomous Systems Research (LASR). The experiments were performed in a motion-capture testbed in the Prototyping Highbay at LASR, which is 150 by 75 ft and equipped with 115 Vicon motion-capture cameras. Flight tests were conducted using two Ascending Technologies Pelican quadrotors. The Pelican has two onboard

computers, one for flight stabilization and a Linux computer for sensor integration and control calculations. The Linux computer runs the Robot Operating System (ROS), which is a message-passing architecture for autonomous robots.

In the experiment, the high-altitude quadrotor was commanded to hover at the origin at an altitude of 2.5 m. The instrumented quadrotor was commanded to go to the waypoint (2.5, 0.0) m while an altitude of 1.5 m from five separate initial conditions. Fig. 3.11 shows the results of the flight test for all of the runs. Fig. 3.11(a) shows the trajectory of two quadrotors. The black X indicates the position of the hovering quadrotor and the green X shows the desired waypoint. The dashed black circle shows the approximate area where the downwash of the hovering quadrotor is significant. The dashed red trajectory shows the path of the vehicle with the flow measurement system and control disabled to show the nominal trajectory the vehicle takes. (Note that for safety reasons the nominal trajectory was implemented without the presence of the hovering quadrotor.) The other trajectories show the quadrotor with the flow measurement system and control enabled. Fig. 3.11(b) shows the measurements (colored dots) taken by the flow measurement system for the red colored run as well as the estimation error (solid lines) of the hovering quadrotor's position for all five runs that included the hovering vehicle (the estimates are color coded to match the trajectories in Fig. 3.11(a)). Fig. **??** shows the x and y position estimates of the hovering quadrotor for the red run.

The experiments illustrates the necessity of the flow measurement system and the flight-path planner. A vehicle without the combined system travels in a straight trajectory towards the waypoint, as indicated by the dashed red line in Fig. 3.11(a).



(a) (b) (c)

Figure 3.11: (a) Trajectory of the quadrotor from five separate initial conditions with the hovering vehicle located at the origin and one initial condition without the hovering vehicle. (b) measurements from the airspeed probes and hovering quadrotor estimation error from red trial; (c) position estimate of the hovering quadrotor for the red run (middle trajectory)

This trajectory would place the vehicle directly in the downwash of the higher quadrotor. The vehicle with the flow measurement and flight-path planner detects the downwash of the higher quadrotor at approximately $t = 3.0$ seconds for the red, blue, and green trajectories, as seen in Fig. 3.11(b). As soon as the sensor measures a positive airspeed, the error in the estimate of the position of the higher quadrotor drops to close to the desired position. Note that the error in the estimate is likely due to a combination of sensor noise and uncertainty in the choice of parameters in the likelihood function. Once the estimate of the higher quadrotors position converges, the instrumented vehicle maneuvers to avoid the downwash, as shown by the red, blue, and green trajectory in Fig. 3.11(a). The position estimates for the red run seen in Fig. 3.11(c) show how the estimate converges to near the correct value when the instrumented vehicle passes close to the hovering quadrotor. Note that

the estimate drifts away once the downwash is no longer detected, due to process noise that is added to the recursive Bayesian filter to avoid probability collapse. The magenta and black trajectories do not travel close enough to the hovering quadrotor to even detect the downwash. This is apparent in both Figures 3.11(a) and 3.11(b) as the vehicle does not maneuver to avoid the downwash and the estimate never converges.

Chapter 4: Distributed Detection of Mobile Targets

In the previous chapter, controllers were designed to stabilize rotorcraft in strong wind fields. With this capability, it is now possible to design algorithms for rotorcraft for sampling spatiotemporal processes, as is done in this chapter. In this chapter, an aerial sensor network composed of quadrotors is used to sample a discrete spatiotemporal process, namely, to detect an unknown number of moving ground targets. First, the problem formulation is discussed, followed by the development of the multi-agent coordination strategy and experimental validation of the detection architecture.

4.1 Problem Formulation

Consider the following problem. There are N aerial vehicles equipped with downward facing cameras. They are searching for an unknown number of moving targets in a given domain. Furthermore, the sensors are of low quality, i.e., complex vision-based detection algorithms cannot be used. The proposed problem is: how does one coordinate the vehicle motion to maximize the probability of target detection?

The approach used in this dissertation consists of a two step process. First,

vehicles assimilate camera measurements into a Likelihood Ratio Tracker (LRT), which calculates the likelihood of target locations throughout the domain. Second, vehicles use the output of the LRT in a physics-inspired control strategy to coordinate their motions so as to maximize probability of detection. In this section, I discuss the details of the update and predict steps for the LRT.

4.2 Likelihood Ratio Tracker for Target Detection

4.2.1 Update step: The sensor measurement model

Consider a measurement data model based on an imperfect binary sensor. Suppose the criterion location for a positive response is chosen such that targets within range ρ_{target} are detected with probability P_d and false alarms occur with probability P_f per time step. The sensitivity m of each sensor is [85]

$$m = z(P_d) - z(P_f), \tag{4.1}$$

where $z(\cdot)$ represents the z-transformation into standard deviation units given by the quantile function [85]

$$z(p) = \sqrt{2}\text{erf}^{-1}(2p - 1).$$

For example, $P_d = 0.95$ and $P_f = 0.1$ yields $m = 2.92$. Let w_k represent unit-normal measurement noise in standard deviation units at time step k. When the target is absent, the measurement data is $z_k = w_k$, whereas when the target is present the measurement data is $z_k = m + w_k$. Assuming a zero-mean Gaussian sensor

model [86] yields

$$p(z_k|\theta_k^-) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{z_k^2}{2}\right) \tag{4.2}$$

$$p(z_k|\theta_k^+) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{(z_k-m)^2}{2}\right). \tag{4.3}$$

(The sensor noise variance is absent because z_k and m are already assumed to be in standard deviation units.) The log likelihood ratio (2.3) becomes

$$\log \mathcal{L}(z_k|\theta_k) = -\frac{(z_k-m)^2}{2} + \frac{z_k^2}{2} = m\left(z_k - \frac{m}{2}\right), \tag{4.4}$$

where m is a function of the sensor P_d and P_f given by (4.1).

While the methodology above would apply to any sensor characterized by its P_d and P_f, in this dissertation the sensor is a camera with a known field of view (FOV) and resolution. Using standard image processing techniques [87], the camera places a bounding box around targets within the FOV. When there is a target present, (4.4) with $z_k = m + w_k$ is applied to the prior uniformly inside a disc of radius ρ_{target} located at the center of the bounding box. In areas of the FOV where there is no target, (4.4) with $z_k = w_k$ is applied to the prior uniformly. P_d and P_f for a camera are complicated functions of the camera resolution, the size of the object in the FOV, and the quality of the image processing algorithm. For the experiments presented in this dissertation, P_d and P_f were treated as tuning parameters for the tracker and were chosen to improve the performance of the control and estimation/detection strategy.

4.2.2 Predict step: Integrating the diffusion equation

The predict step (2.2) involves updating the target probability density function in the absence of measurement information. Targets are allowed to move randomly, though I do not estimate their velocities. The target motion model is a random walk, which can be described mathematically by the diffusion equation with constant diffusivity, also called the heat equation.

Consider the two-dimensional heat equation with diffusivity α, i.e.,

$$\frac{\partial P}{\partial t} = \alpha \left(\frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial y^2} \right).$$

The alternating direction implicit (ADI) method [88] is a fractional-step method for integrating the heat equation. ADI employs the difference operators A_x and A_y representing the spatial derivatives in the x and y directions, respectively. For example, if probability P is discretized over $n \times n$ grid points, then $A_x P$ is a vector of length $n \times n$ with elements

$$\frac{P_{i+1,j} - 2P_{i,j} + P_{i-1,j}}{\Delta x^2}, \quad i = 1, \ldots, n, \quad j = 1, \ldots, n.$$

Let I be the $n \times n$ identity matrix. The solution is obtained from $P(\theta_{k-1}|z_{k-1})$ using matrix inversion to solve the following equation for $P(\theta_k|z_{k-1})$:

$$\left(I - \tfrac{\alpha \Delta t}{2} A_x\right) \left(I - \tfrac{\alpha \Delta t}{2} A_y\right) P(\theta_k|z_{k-1}) =$$

$$\left(I + \tfrac{\alpha \Delta t}{2} A_x\right) \left(I + \tfrac{\alpha \Delta t}{2} A_y\right) P(\theta_{k-1}|z_{k-1}).$$

To ensure that no information crosses the boundary of the search domain, I enforce Neumann boundary conditions around the perimeter of Ω by specifying that the normal gradient is zero.

4.3 Physics-Inspired Motion Planning

The agent motion is driven by a combination of two artificial forces. The first force guides the vehicle down the gradient of the inverse log likelihood surface, i.e., the temperature surface, in a manner that resembles a flowing liquid. The speed of movement down the gradient is determined by the temperature at the vehicle location, where colder temperatures correspond to slower speeds. The second force is the gradient of a Lennard-Jones potential [89] between the agents. The Lennard-Jones potential is a common approximation used in physical chemistry to model molecular gas dynamics. Figure 4.1 shows the vehicles moving like gas molecules in areas where there are no targets and gathering like solid molecules with other agents where there are likely to be targets. In this algorithm, as opposed to other physics-inspired swarm controllers, the states-of-matter behavior is emergent.

Let r_k^j denote the position of agent $j \in \{1, \ldots, N\}$ at time step k. The Lennard-Jones potential is [89]

$$V_k^j = \sum_{i \neq j}^{N} 4\epsilon \left(\sigma^{12} ||r_k^j - r_k^i||^{-12} - \sigma^6 ||r_k^j - r_k^i||^{-6} \right), \tag{4.5}$$

where ϵ is the depth of the well and σ is the distance at which the potential between two agents is zero. An example of the Lennard-Jones potential between two agents is shown in Figure 4.2. Note that the strength of the potential drops quickly to zero for large distances. Thus, the interaction between two agents only happens when they are close to one another.

Let $T_k^j = -P(\theta_k | z_k, j)$ in (2.5) be the temperature at vehicle j at time k and

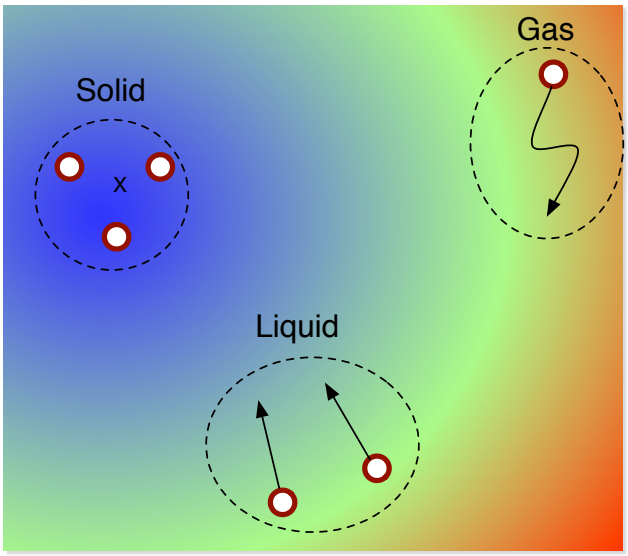


Figure 4.1: Diagram depicting the three emergent behaviors of the proposed algorithm. Vehicles in cold areas that are near a target will form a crystalline formation with nearby agents. Vehicles that are in warm areas will flow on the temperature surface like a liquid. Agents in hot areas will travel quickly in a random, gas-like fashion.

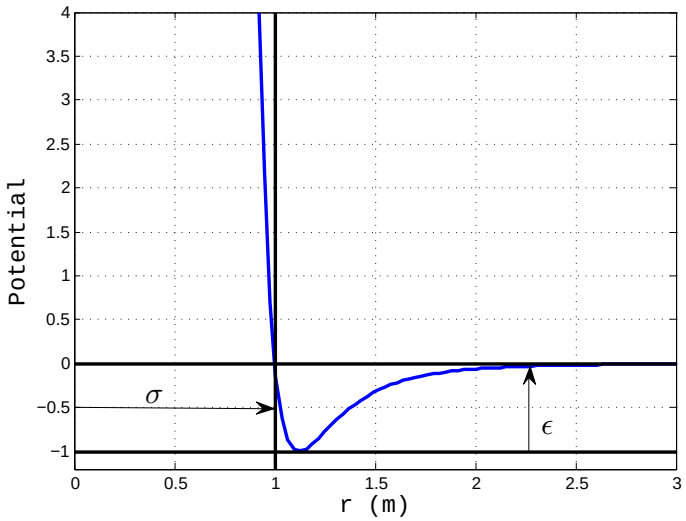


Figure 4.2: Example of a Lennard-Jones potential with $\epsilon = \sigma = 1.0$.

V_k^j be the Lennard-Jones potential (4.5) for the jth vehicle. IThe desired velocity for each vehicle is prescribed using a feedback controller. The desired velocity for the jth vehicle is given by

$$\dot{r}_k^j = -\min(1 + e^{T_k^j}, v_{max}) k_P \nabla T_k^j - k_V \nabla V_k^j, \tag{4.6}$$

where k_P and k_V are control gains and v_{max} is the maximum speed of the agent (assumed to be the same for all agents). The dependence on temperature causes the vehicles to slow down when they are near a possible target in order to collect more measurements in that area. This behavior, along with the sensor aggregation caused by the Lennard-Jones potential, seeks to increase the chances that a target will be detected. The algorithm works well in simulations when the gain on the Lennard-Jones potential is one order of magnitude smaller than the gradient-following gain. Additionally, σ is set to ensure that the vehicles do not collide with each other, which depends on the scale of the vehicle. Consequently, there are only three parameters to tune: either k_P or k_V, ϵ (the well depth), and T_0 (the detection threshold).

As seen in equation (4.6), the gradient of the temperature potential is multiplied by a heuristic scaling factor. This scaling factor, in combination with the Lennard-Jones potential, is what causes the states-of-matter behavior to be emergent. In the scaling factor, the speed of the vehicle is scaled by the temperature such that the speed increases when the temperature is high, and decreases when the temperature is slow. This causes vehicles to "freeze" in place when they are near an area of high probability and, conversely, to speed up when they are in areas of low probability. The full architecture (in pseudocode form) is shown in Algorithm 1.

Algorithm 1 Physics-Inspired Target Search and Detection Algorithm

Require: $\mathrm{N}, \epsilon, \sigma,\ k_P,\ k_V$

Initialize log likelihood ratio $P_0(\theta|z)$

for $k = 1$ till end of task **do**

  for $j \dots N$ **do**

    Get measurement z_k^j

    Calculate likelihood ratio $\log \mathcal{L}(z_k^j|\theta_k)$ using (4.4)

    Update: $P_k(\theta|z) = P_{k-1} + \log \mathcal{L}(z_k^j|\theta_k)$

  end for

  Normalize likelihood ratio: $P_k(\theta|z) = P_k(\theta|z) - \mathrm{sum}((P_k(\theta|z)))$

  Get r_k for every vehicle

  for $j = 1 \dots N$ **do**

    Compute $\nabla V_k^j(r_k)$ and $\nabla P(\theta_k|z_k)$

    Calculate desired velocity $\dot{r}_k^j$

    Apply control input to platform to achieve $\dot{r}_k^j$

  end for

end for

Unlike of other works in the field, the architecture shown in Algorithm 1 avoids local minima in an informative, non-heuristic fashion. A vehicle can fall into a local minima if it is surrounded by hills of low target probability. However, if there is no target present underneath the vehicle, the temperature will increase (from negative information) to the point where the vehicle is pushed away from its location. Note, this behavior is completely emergent and is provided inherently by connecting the detection scheme to the vehicle control. In addition to informatively handling local minima, there is the following theorem.

Theorem 4.3.1 *The mutual information between sensor measurement z_k and target location θ_k using the sensor model given by* (4.2)–(4.3) *is locally maximized when the vehicle moves along the gradient of the prior target distribution $p(\theta_k)$.*

Proof Let $\xi_k = (x_k, y_k)$ be the position of the kth quadrotor. For readability I drop the temporal subscript on all the variables in this proof. The mutual information between z and θ is

$$I(z,\theta) = \int_{z^+,z^-} \int_{\Omega} p(\theta, z|\xi) \log(p(z|\theta,\xi) d\theta dz - \int_{z^+,z^-} p(z|\xi) \log p(z|\xi) dz, \tag{4.7}$$

where $p(z|\xi)$ is the marginal probability density function of the observation in observation space. Since the sensor model is Gaussian, the logarithm in the first term on the right hand side of (4.7) evaluates to a constant. Hence, the double integral is independent of ξ. Thus, the mutual information is maximized by maximizing the second term, which is the entropy of the observation, $H(z|\xi)$, given the vehicle

location. The marginal probability $p(z|\xi)$ can be written as

$$p(z|\xi) = \int_{\theta \in D} p(z|\theta, \xi) p(\theta) d\theta. \tag{4.8}$$

Therefore, $H(z|\xi)$ can be expressed as

$$H(z|\xi) = -\int_{z^+, z^-} \int_{\theta \in D} p(z|\theta, \xi) p(\theta) d\theta \times \log \left(\int_{\theta \in D} p(z|\theta, \xi) p(\theta) d\theta \right) dz. \tag{4.9}$$

Evaluating the integral over z yields

$$\begin{aligned} H(z|\xi) &= -\int_{\theta \in D} p(z^+|\theta, \xi) p(\theta) d\theta \times \log \left(\int_{\theta \in D} p(z^+|\theta, \xi) p(\theta) d\theta \right) \\ &\quad - \int_{\theta \in D} p(z^-|\theta, \xi) p(\theta) d\theta \log \left(\int_{\theta \in D} p(z^-|\theta, \xi) p(\theta) d\theta \right). \end{aligned} \tag{4.10}$$

Both terms on the right hand side of (4.10) have the form $f(x) = x \log(1/x)$, which is an increasing function of x as long as $x \leq 1/e \approx 0.36$. In (4.10), the argument is the integral of $p(\theta)$ scaled by either $p(z^+|\theta, \xi)$ or $p(z^-|\theta, \xi)$. Since these are both less than $1/e$ (see (4.2)–(4.3)) and the integral of $p(\theta)$ is less than one in domain D, $H(z|\xi)$ is guaranteed to be an increasing function of the integral of $p(\theta)$.

Because the mutual information is an increasing function of the integral of $p(\theta)$, moving along the gradient of the integral will maximize the mutual information at the next time step. As a first-order approximation, assume that $p(\theta)$ is given by the first-order Taylor expansion

$$p(\theta) \approx a_0 + a_x x + a_y y, \tag{4.11}$$

where it is assumed that coordinates in D are given by the pair (x, y). Then I have

$$\int_{\theta \in D} p(\theta) d\theta \approx \int_{x_k - \frac{s}{2}}^{x_k + \frac{s}{2}} \int_{y_k - \frac{s}{2}}^{y_k + \frac{s}{2}} (a_0 + a_x x + a_y y) dy dx, \tag{4.12}$$

where I have assumed that D is a square region with side length s for simplicity. The gradient of (5.38) yields

$$\nabla \left(\int_{\theta \in D} p(\theta) d\theta \right) \approx a_x s^2 \hat{e}_x + a_y s^2 \hat{e}_y, \tag{4.13}$$

where $\hat{e}_x$ and $\hat{e}_y$ are unit vectors in the x and y directions respectively. Taking the gradient of $p(\theta)$ directly shows that

$$\nabla p(\theta) \approx a_x \hat{e}_x + a_y \hat{e}_y = \frac{1}{s^2} \nabla \left(\int_{\theta \in D} p(\theta) d\theta \right). \tag{4.14}$$

Thus, moving along the gradient of $p(\theta)$ is locally the same as moving along the gradient of its integral, which completes the proof.

Theorem 1 shows that the binary sensor model coupled with a gradient control allows each vehicle to greedily maximize the expected gain in information without explicitly having to calculate it. This property has significant benefit on the ability of the search strategy to scale with numbers of vehicles and targets, as compared to the calculation of mutual information, which generally scales exponentially. Theorem 1 only holds locally. Since the Lennard-Jones potential drops off dramatically with distance, each vehicle need only consider the position of other nearby agents. Therefore, the algorithm scales linearly with the number of vehicles. Note that if the algorithm was distributed, the complexity would be fixed by local vehicle density under a distance-based communication topology.

Algorithm performance is evaluated via simulation in the operating domain $\Omega = [-5 \quad 5] \times [-5 \quad 5]$ m. Agents and targets are constrained to remain in Ω at all times. Parameter definitions and values used in the simulation are provided in

Figure 4.3. The local temperatures of the agents and targets are obtained via linear interpolation between the grid points. The overall temperature surface is normalized by subtracting its mean every time step. To avoid overheating in the temperature surface, the individual sensor updates are scaled by N, i.e., the number of agents. This scaling has the effect of making the collective detection performance in terms of the cumulative number of targets detected over time roughly independent of N.

Figure 4.4 shows the results of the simulation. In Figures 4.4(a) and 4.4(b) there are snapshots of the vehicles, targets, and temperature (inverse log likelihood) surface at time steps $k = 125$ and 425 respectively. Vehicles are shown as red circles, undiscovered targets as white crosses, and discovered targets as red crosses. At any given time there are only three undiscovered targets in the domain. Once a target is detected, it turns red and another target appears with uniform probability somewhere in the domain. The heat map, which describes the temperature, is blue in areas of low temperature (high likelihood) and red in areas of high temperature (low likelihood).

In Figure 4.4(a), one target has been detected. There are several vehicles scanning the warmer regions like gas molecules, while others are clustering around potential targets in a semi-solid configuration. Figure 4.4(c) shows the cumulative number of targets captured during the simulation (solid line) and the amount of time it takes for the vehicles to fully cover the domain once (dashed line). The vertical red lines indicate the times the snapshots were taken. This algorithm follows a typical coverage curve seen for greedy algorithms, with approximately linear growth in coverage early on, followed by slower asymptotic convergence to full coverage.

Also of note is that the target capture rate is also approximately linear.

4.4 Experimental Results

I implemented the algorithm in hardware using a quadrotor testbed at the Laboratory for Autonomous Systems Research located at the Naval Research Laboratory. The testbed is composed of Ascending Technologies Pelican quadrotors operated in a facility with 115 Vicon motion capture cameras. Figure 4.5 shows the prototyping high bay where the experiments were conducted. The quadrotors are equipped with active LED motion capture markers in order to be seen by the Vicon cameras.

The quadrotors each have a downward facing Point Grey Chameleon 1.3 megapixel camera with a resolution of 1296×964 pixels. The camera is equipped with a Tamron CCTV manual lens to adjust zoom and focus. The camera has a global shutter to minimize motion blur and adjustable white balance and shutter speed. The quadrotors have an onboard Linux computer that processes camera frames at 15 Hz, which is limited by how fast images can be captured by the camera. The binary signal sent to the LRT is whether or not a blob of a predetermined color is in the image. For the experiments, the blobs were Sphero robots, a spherical rolling toy which has an orange covering. Figure 4.6 shows an example of the blob tracking software used to detect two Spheros in the field of view of the Point Grey camera.

Pose and target information from each vehicle is sent to a laptop, which uses

the Robot Operating System (ROS) [90] to process the data and compute control commands in a control node. ROS is an open-source software architecture that contains software libraries and tools to build robotic applications. The binary signal is generated by using a standard blob segmentation algorithm [87] in a camera processing ROS node onboard the vehicle. The vehicle also has a ROS node to handle passing measurements to the control node running on a laptop as well as receiving position and control commands from the external laptop and motion capture system. Control commands are sent to the vehicle through a WiFi connection to the message processing node at approximately 30 Hz. The full architecture for the experiment is shown in Figure 4.7.

4.4.1 Single-Vehicle, Single-Target Experiment

To verify the algorithm using the hardware testbed, experiments were first conducted with a single vehicle and a single moving target. Figure 4.8 shows a snapshot of an experiment. The targets are seen using the Point Grey camera, where the targets (and false alarms) were found in the image. The blob detections are fed into the LRT to calculate the inverse log likelihood surface, from which the desired control can be calculated. Once detected, any blob detected within a 0.5 m radius of the detected target are ignored by the LRT tracker. The Sphero was commanded to move in a random walk, so multiple detections of the target were possible.

In this experiment, the Sphero was detected three times as it performed the

random walk. The locations where the target was detected are indicated by the red crosses. Figures 4.8(a) and 4.8(b) show snapshots of the experiment at timesteps $k = 120$ after the first detection and $k = 220$ after the third detection. The number of targets captured over time is shown in Figure 4.8(c), where the vertical red lines indicate the times of the snapshot shown in Figures 4.8(a) and 4.8(b). In a real target-tracking application, once the target is detected using Algorithm 1 the vehicle would switch to another controller to maintain a line-of-sight on the detected target such that multiple detections become unnecessary. For illustrative purposes, the vehicle in this experiment continued using Algorithm 1 even after a detection was called. The single-vehicle experiments used the same gains that were used in simulation; the Lennard-Jones potential was omitted as it is a purely collaborative term.

4.4.2 Multi-Vehicle, Multi-Target Experiment

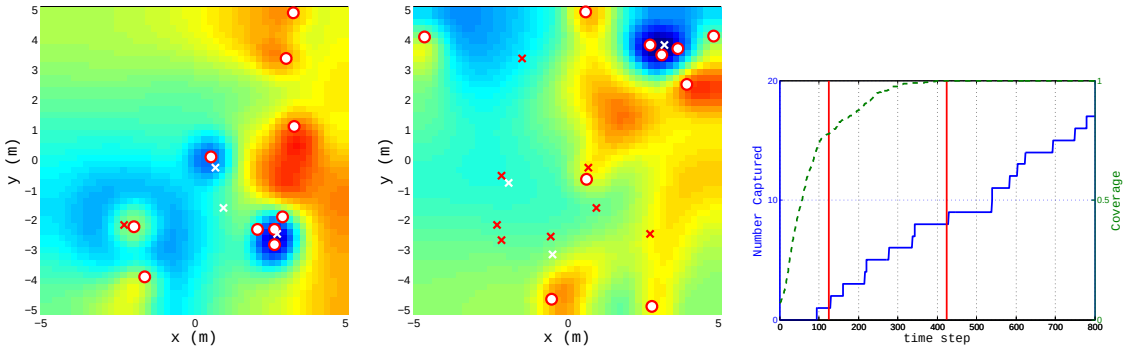
Multi-vehicle experiments were also conducted with stationary targets located at (-0.25, -0.35), (1.60, 0.40), and (1.90, -1.00) meters. Figures 4.9(a) and 4.9(b) show snapshots of the inverse log likelihood surface at time steps $k = 60$ and 155 with the position of the undetected targets shown as black crosses and detected targets shown as red crosses. At the depicted timestep, there is one target that has been recently detected, one that has yet to be discovered, and a third that in the process of being detected. Figure 4.9(c) shows the number of targets captured and the area covered over time. With two vehicles, all targets were captured within 200

time steps and the vehicles covered the full domain in approximately 400 time steps, which corresponds to 40 seconds.

Comparing Figure 4.4(c) to Figure 4.9(c), it is seen that the coverage rate in the experiment follows the same patterns as the simulated experiments, i.e., initial linear growth followed by asymptotic convergence. It was also determined that, in general, the vehicles must have lower gains than in simulation. As observed by comparing Figure 4.4(c) to Figure 4.9(c), lower gains result in a slower capture rate. With higher gains, the vehicles tended to overshoot the targets before the temperature was cool enough to slow down the vehicles. The discrepancy between simulation and experiment can be explained by observing that the dynamics of the quadrotors are inherently second-order, whereas the desired velocity stated in (4.6) is for vehicles with first-order dynamics. The Pelicans approximate this control by treating the output of (4.6) as a desired velocity to be achieved by the onboard flight controller.

Parameter	Value	Definition
N	10	number of agents
M	3	instantaneous number of targets
α	0.5	target diffusivity
σ	1.5 m	repulsive threshold
ϵ	0.2	Lennard-Jones potential depth
T_0	-15 °	target detection threshold
K_P	5	gradient-following gain
K_V	0.05	Lennard-Jones gain
$v_{\max}$	0.3 m/s	maximum agent speed
P_d	0.95	probability of detection
P_f	0.10	probability of false alarm
ρ_{target}	α	Sensor detection range
n	50	number of grid points in each dimension
Δx	$2/n$	grid spacing
Δt	0.03 s	time step

Figure 4.3: Parameter values and definitions



(a) Time step $k = 125$ (b) Time step $k = 425$ (c)

Figure 4.4: (a)(b) Snapshot of the simulation; (c) The number of targets captured throughout the simulation and the amount of time taken to cover the entire domain once. Vehicles cluster like solid molecules in cold regions near the target and explore randomly like gas particles in hot regions, while flowing down the temperature gradient from hot to cold.

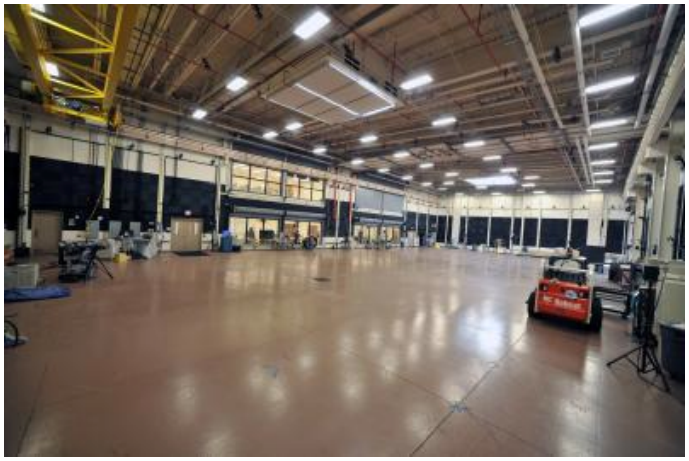


Figure 4.5: Prototyping high bay in the Laboratory for Autonomous Systems Research at the Naval Research Laboratory.

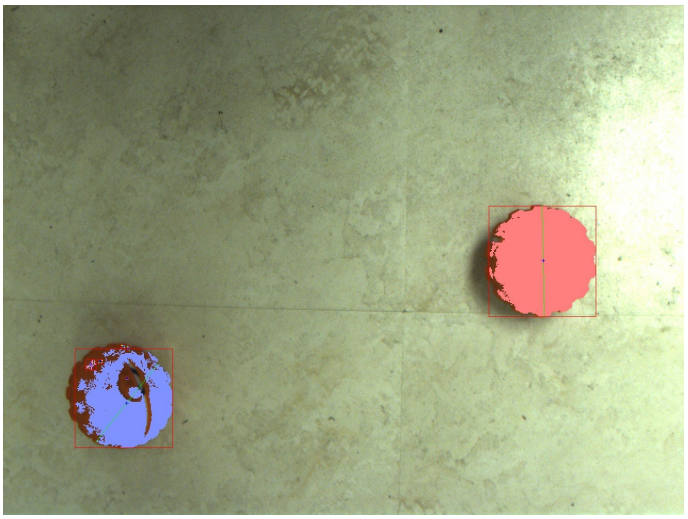


Figure 4.6: Processed camera image showing two orange Spheros identified using a color blob detector.

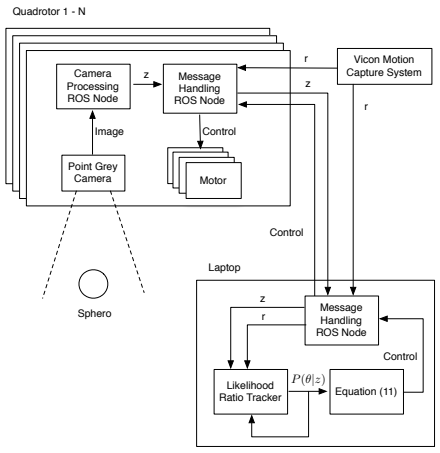
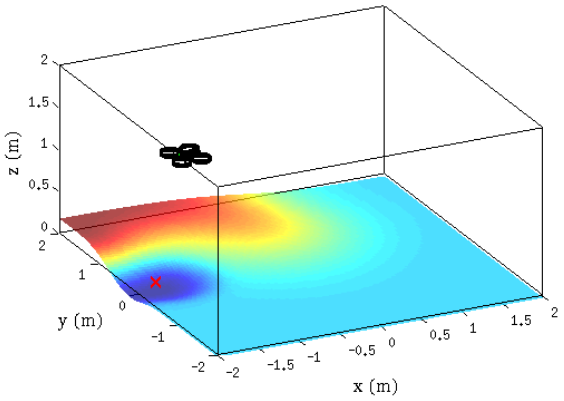
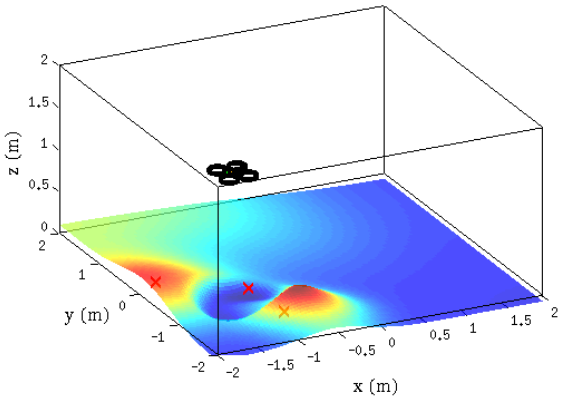


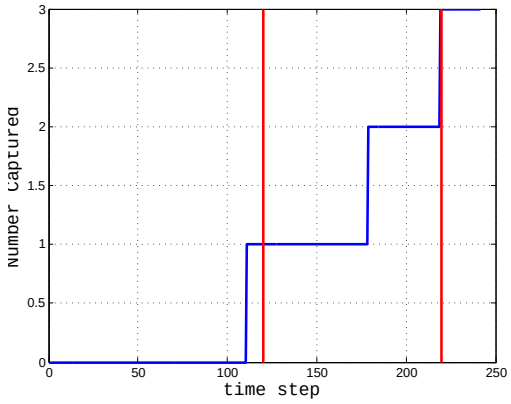
Figure 4.7: Hardware architecture for the experiments. The quadrotors run two ROS nodes to process camera data and handle incoming position measurements and desired velocity. An external laptop receives position and blob detection measurements to compute the temperature surface and calculate the desired velocity.



(a) Time step $k = 120$

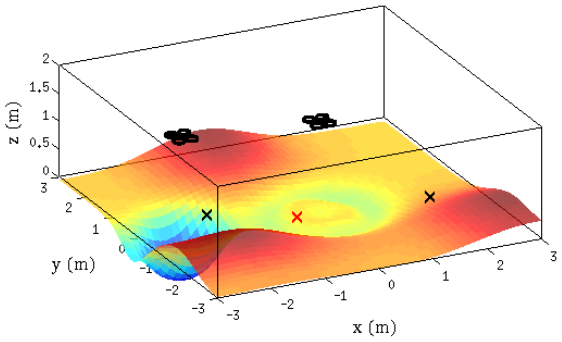


(b) Time step $k = 220$

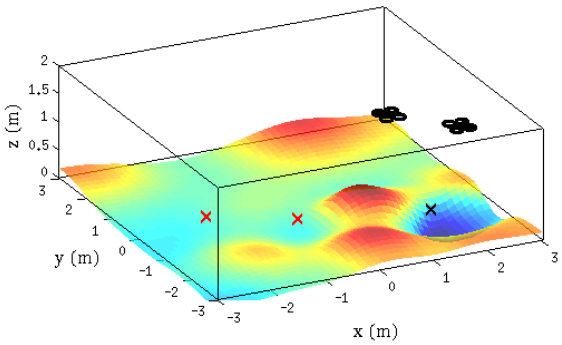


(c)

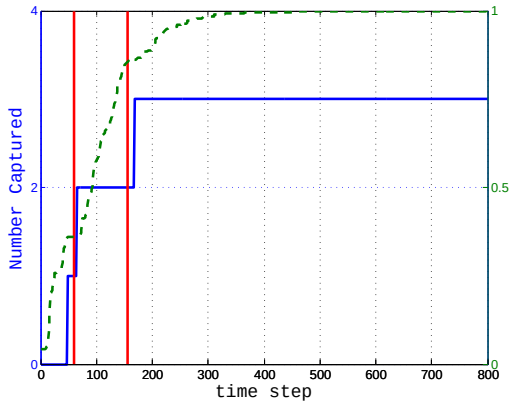
Figure 4.8: (a)(b) Snapshot of the single-vehicle experiment; (c) The number of targets captured throughout the experimetn and the amount of time taken to cover the entire domain once.



(a) Time step $k = 60$



(b) Time step $k = 155$



(c)

Figure 4.9: a)(b) Snapshot of the multi-vehicle experiment; (c) The number of targets captured throughout the experiment and the amount of time taken to cover the entire domain once.

Chapter 5: Distributed Sampling of Gaussian Processes

5.1 Problem Formulation

While the previous chapter used rotorcraft to sample a discrete process, this chapter samples a continuous spatiotemoral process. In particular, I investigate the use of an aerial, multivehicle, sensor network to sample a continuous spatiotemporal field $A(r)$, where the field is modeled as a Gaussian process with covariance function (2.6)is nonstationary. Specifically, I seek to provide optimal coverage of the field, where the coverage is optimal in the sense that the measurement density is proportional to the variability of the field. To determine when measurements are redundant, I consider the footprint of a measurement, which is defined as the volume in space and time contained in an ellipsoid centered at the measurement location with principle axes equal to the decorrelation scales of the field. The coverage task is illustrated in Figure 5.1, which depicts two vehicles taking measurements along a circular trajectory in the space-time domain. The red and blue tubes along the vehicles' trajectories are the sensor swaths created by the measurement footprints. Figure 5.1(a) shows the swaths for a stationary field and Figure 5.1(b) shows the swaths for a nonstationary field in which the temporal scales contract at $t = 10$ hours. The first goal is to design the vehicle trajectories $x_k(t)$ and $y_k(t)$ so that the

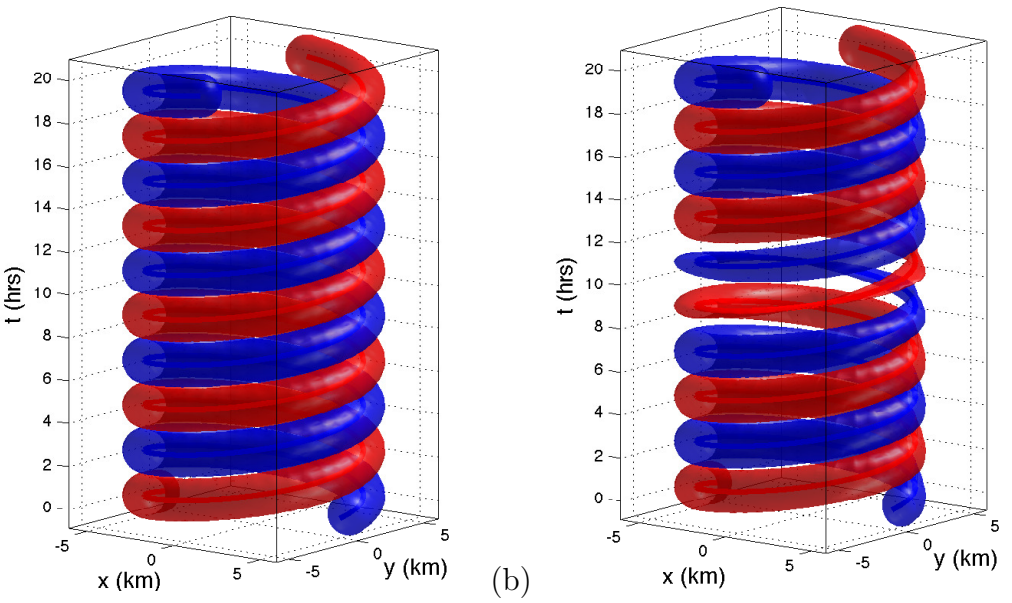


Figure 5.1: Two vehicles traveling through (a) stationary and (b) nonstationary spatiotemporal fields. The tubes represent the volume covered by sensor measurements.

swaths created by the set of all measurement footprints cover the entire field with minimal overlaps or gaps when the decorrelation scales of the field vary in space and time. Measurements contain Gaussian noise with zero mean and known variance. The second goal is to optimally sample the field to determine the decorrelation scales and the switch to a mapping formation, a notion known as explore versus exploit.

5.2 Sampling a Stationary Spatiotemporal Field

The sampling strategy presented in this dissertation relies on the ability to sample a one-dimensional spatiotemporal field, as considered in [60] and reviewed here. The field is modeled as a continuous spatiotemporal Gaussian process as described in 2.2. Consider a closed path, where $\theta_k \in \mathbb{S}^1$ is a path variable that denotes

the non-dimensional location of the k^{th} vehicle along the path. Each vehicle collects measurements at a sufficiently high rate such that the time between measurements is much less than the temporal decorrelation scale (to ensure that its sensor swath is approximately continuous). Under speed control u_k, the dynamics of the k^{th} vehicle is

$$\dot{\theta}_k = u_k(\theta_1, ..., \theta_N) = u_k(\theta),$$

where $\theta = [\theta_1, \ \cdots, \ \theta_N]^T$. Note, if the path is a circle of radius ρ, then $r_k = (\rho\cos\theta_k, \rho\sin\theta_k, t_k)$, where t_k is time. Note that for the remainder of this chapter, all dynamics are assumed to be first-order. In reality, multi-rotor systems behave as a second-order system. This means that the controllers developed in this chapter are treated as an input to a trajectory following controller or a flight stabilizer, which are discussed in Chapter 5.

Since the vehicles sample uniformly in time, the mapping error is minimized in a stationary field by traveling at maximum speed to place as many measurements as possible in the domain, as further illustrated in [58, 60]. To determine the number of vehicles necessary to optimally sample the domain, the sampling speed of the vehicle is used, defined as [72]

$$s_k = v_k \frac{\tau}{\sigma}, \tag{5.1}$$

where v_k is the vehicle speed and τ and $\sigma = \sigma_x = \sigma_y$ are the temporal and spatial scales of the field, respectively. Note that $\dot{\theta}_k = v_k$ only when the vehicle is exactly on the path.

In one spatial dimension (plus time), the sensor swath is the portion of the

domain covered by the set of coverage ellipses generated along the sampling trajectory. A sensor swath is considered *spatially constrained* if $s_k < 1$ and *temporally constrained* if $s_k > 1$, as illustrated in [60]. Figures 5.2(a) and 5.2(b) show the sensor swaths for spatially and temporally constrained sampling patterns, respectively. (Vehicles exiting on the right reappear on the left since the path is closed.) The

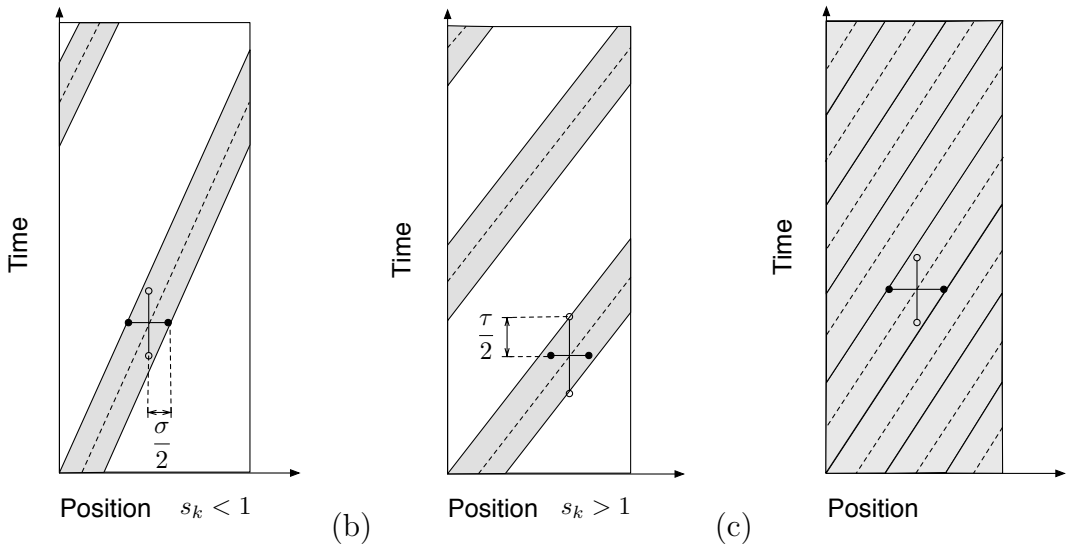


Figure 5.2: Sensor swaths for (a) spatially constrained and (b) temporally constrained vehicles in a stationary field; (c) Tiling the domain with three spatially constrained vehicles ensures complete coverage.

number of vehicles required to fully cover a closed path in a stationary domain is determined by the value of s_k as shown in [60].

Theorem 5.2.1 *The minimum number N of vehicles with sampling speed s_k required to cover a stationary, spatiotemporal field that is periodic in its one spatial*

dimension is

$$N = \begin{cases} \lceil 2\pi/\sigma \rceil & \text{if } s_k < 1, \\ \lceil T/\tau \rceil & \text{if } s_k \geq 1, \end{cases}$$

where $\lceil x \rceil$ *is the smallest integer larger than* x*, and* $T = 2\pi/v_k$ *is the period of a single revolution around the path that has nondimensional length* 2π *.*

Proof As depicted in Figure 5.2, if the vehicle is spatially constrained, then σ determines the swath width. Since (without loss of generality) the length of the path is taken as 2π, the minimum number of vehicles needed to cover the domain is the smallest integer greater than $2\pi/\sigma$. If the vehicle is temporally constrained, then τ determines the swath width and, if T is the period of one revolution, the number of vehicles needed to cover the domain is the smallest integer greater than T/τ.

To minimize the overlap in coverage, the vehicles are spaced evenly along the path, as depicted in Figure 5.2(c), using a splay controller from [91], as follows. Consider the splay phase potential

$$S(\theta) \triangleq \sum_{m=1}^{\lfloor \frac{N}{2} \rfloor} K_m U_m(\theta), \quad K_m > 0, \tag{5.2}$$

as used in [91], where $\lfloor x \rfloor$ is the largest integer less than x ,

$$U_m(\theta) \triangleq \frac{N}{2} |p_{m\theta}|^2, \quad \text{and} \quad p_{m\theta} \triangleq \frac{1}{mN} \sum_{j=1}^{N} e^{im\theta_j}.$$

Choosing the control law

$$u_k = -\frac{\partial S}{\partial \theta_k} = -\frac{K_m}{N} \sum_{j=1}^{N} \sum_{m=1}^{\lfloor \frac{N}{2} \rfloor} \frac{1}{m} \sin(m(\theta_k - \theta_j)), \tag{5.3}$$

from [91], locally stabilizes the set of critical points of S, which includes the set of splay formations [91, Theorem 7]. Note that, while the set of splay formations is only one of the critical points of S, all other critical points are unstable [91], thus, (5.3) will drive the vehicles to a splay formation. Also, adding a constant drift vector field to the right-hand side of (5.3) to represent motion around the path does not change the result [91], i.e., a term may be added to the right of (5.3) to represent the vehicle speed. Hence, adding a constant (representing the steady-state speed of the vehicle) to (5.3) allows us to drive the vehicles to a constant speed around the path while maintaining a splay formation.

5.3 Transformation to a Locally Stationary Field

In this section, a nonlinear coordinate transformation is defined that renders a nonstationary field locally stationary. The significance of this transformation is that it permits the use of multivehicle sampling algorithms designed for use in stationary fields. I derive two transformations, one for which the spatial decorrelation scales are coupled, i.e., $\sigma_x = \sigma_x(x_k, y_k)$ and $\sigma_y = \sigma_y(x_k, y_k)$, and another for which the spatial scales are decoupled, i.e., $\sigma_x = \sigma_x(x_k)$ and $\sigma_y = \sigma_y(y_k)$.

5.3.1 Coupled Decorrelation Scales

Let $r_k \triangleq [x_k,\ y_k,\ t_k]^T$ be the space-time coordinates in the original domain, called the r-domain, and $R_k \triangleq [X_k,\ Y_k,\ T_k]^T$ be a set of transformed coordinates in a new domain, called the R-domain. Also, let $G(V, E)$ be a lattice graph with

nodes V placed at each R_k and undirected edges E connecting adjacent cells (the four nearest neighbors in two dimensions). The lattice graph ensures that adjacent nodes are locally stationary under the transformation. The goal is to find R_k such that the field with covariance (2.6) becomes locally stationary, i.e., all adjacent nodes in G are (approximately) stationary, as in [65]. Intuitively, this corresponds to ensuring that the field is stationary in the neighborhood of each point R_k. Let $\bar{\Sigma}(r_i, r_j) = (\Sigma(r_i) + \Sigma(r_j))/2$, $r_{ij} \triangleq r_i - r_j$, and $R_{ij} \triangleq R_i - R_j$. Thus, the relation to satisfy is

$$\frac{|\Sigma(r_i)|^{1/4}|\Sigma(r_j)|^{1/4}}{\sqrt{\left|\bar{\Sigma}(r_i, r_j)\right|}} e^{\left[-r_{ij}^T(\bar{\Sigma}(r_i,r_j))^{-1}r_{ij}\right]} = e^{\left[-R_{ij}^T R_{ij}\right]} \tag{5.4}$$

between all adjacent points i and j. The left-hand side of (5.4) is the nonstationary covariance function in the r-domain and the right hand side is the covariance function in the R-domain, which is desire to be stationary. Without loss of generality, the spatial and temporal decrorrelation scales in the R-domain are chosen to be $\sigma_X = \sigma_Y = \tau_T = 1$. Since the spatial decorrelation scales are coupled, the transformations are also coupled; however, the temporal component of (5.4) is decoupled, by assumption. Let $\sigma_{xi} \triangleq \sigma_x(x_i, y_i)$, $\sigma_{yi} \triangleq \sigma_y(x_i, y_i)$, and $\tau_i \triangleq \tau(t_i)$. Then, (5.4) yields the following two relations:

$$(X_i - X_j)^2 + (Y_i - Y_j)^2 = \frac{2(x_i - x_j)^2}{\sigma_{xi}^2 + \sigma_{xj}^2} + \frac{2(y_i - y_j)^2}{\sigma_{yi}^2 + \sigma_{yj}^2}$$
$$+ \ln\left[\left(\frac{\sigma_{xi}^2 + \sigma_{xj}^2}{2\sigma_{xi}\sigma_{xj}}\right)\left(\frac{\sigma_{yi}^2 + \sigma_{yj}^2}{2\sigma_{yi}\sigma_{yj}}\right)\right] \quad \text{and} \tag{5.5}$$

$$(T_i - T_j)^2 = \frac{2(t_i - t_j)^2}{\tau_i^2 + \tau_j^2} + \ln\left(\frac{\tau_i^2 + \tau_j^2}{2\tau_i\tau_j}\right). \tag{5.6}$$

Equations (5.5) and (5.6) show how the space-time separation between two

points in the R-domain depends on the nonstationarity in the r-domain. However, to complete the spatial transformation another relationship is needed between $X_{ij} \triangleq X_i - X_j$ and $Y_{ij} \triangleq Y_i - Y_j$ to separate (5.5) into two equations. I propose to use the orientation between points in the R-domain and r-domain. The relation

$$\tan\left(\frac{Y_i - Y_j}{X_i - X_j}\right) = \tan\left(\frac{y_i - y_j}{x_i - x_j}\right) \tag{5.7}$$

preserves the relative orientation between points i and j, which ensures that the transformation will not rotate the coordinates in the R-domain, as it does in [65]. Substituting (5.7) in (5.5) yields

$$\begin{aligned} X_{ij} = \pm \Bigg[& \frac{2(x_i - x_j)^2}{\sigma_{xi}^2 + \sigma_{xj}^2} + \frac{2(y_i - y_j)^2}{\sigma_{yi}^2 + \sigma_{yj}^2} \\ & + \ln\left[\left(\frac{\sigma_{xi}^2 + \sigma_{xj}^2}{2\sigma_{xi}\sigma_{xj}}\right)\left(\frac{\sigma_{yi}^2 + \sigma_{yj}^2}{2\sigma_{yi}\sigma_{yj}}\right)\right]\Bigg]^{1/2} \times \\ & \left(1 + \frac{y_i - y_j}{x_i - x_j}\right)^{-1} \triangleq d_{X,ij}, \end{aligned} \tag{5.8}$$

$$\begin{aligned} Y_{ij} = \pm \Bigg[& \frac{2(x_i - x_j)^2}{\sigma_i^2 + \sigma_j^2} + \frac{2(y_i - y_j)^2}{\sigma_i^2 + \sigma_j^2} \\ & + \ln\left[\left(\frac{\sigma_{xi}^2 + \sigma_{xj}^2}{2\sigma_{xi}\sigma_{xj}}\right)\left(\frac{\sigma_{yi}^2 + \sigma_{yj}^2}{2\sigma_{yi}\sigma_{yj}}\right)\right]\Bigg]^{1/2} \times \\ & \left(1 + \frac{x_i - x_j}{y_i - y_j}\right)^{-1} \triangleq d_{Y,ij}, \quad \text{and} \end{aligned} \tag{5.9}$$

$$T_i - T_j = \pm \left[\frac{2(t_i - t_j)^2}{\tau_i^2 + \tau_j^2} + \ln\left(\frac{\tau_i^2 + \tau_j^2}{2\tau_i\tau_j}\right)\right]^{1/2} \triangleq d_{T,ij}. \tag{5.10}$$

Equations (5.8)–(5.10) represent the transformation from the r-domain to the R-domain. Note that the choice of plus or minus in (5.8)–(5.10) is arbitrary as it represents flipping the new coordinates about the corresponding axis. Thus, the positive root is used.

Let B be the incidence matrix of the lattice graph G. (5.8)–(5.10) are rewritten

as

$$B^T X = d_X, \quad B^T Y = d_Y, \text{ and } \quad B^T T = d_T, \tag{5.11}$$

where d_X is the vector of all terms $d_{X,ij}$ and d_Y and d_T are defined similarly. These equations are, in general, overdetermined. The least-squares solution to (5.11) is obtained using the pseudoinverse:

$$X = (BB^T)^{-1} B d_X, \tag{5.12}$$

$$Y = (BB^T)^{-1} B d_Y, \quad \text{and} \tag{5.13}$$

$$T = (BB^T)^{-1} B d_T. \tag{5.14}$$

A coordinate transformation is locally stationary if equations (5.12)–(5.14) are satisfied. The use of the lattice graph topology is justified since points with short separation (in space or time) have a stronger impact on the mapping error than points with large separation. I have also found the lattice topology to be particularly sensitive to nonstationarities in the r-domain; higher connected graphs tend to concentrate measurements less.

The inverse transformation from the R-domain to the r-domain is solved by creating a lookup table as follows. A uniform grid is created in the r-domain and mapped to the R-domain using (5.12)–(5.14), which produces a nonuniform grid of locations in the R-domain. Trajectories generated in the R-domain are discretized and mapped back to the r-domain by interpolating this grid.

Theorem 5.3.1 *The coordinate transformation given by* (5.12)–(5.14)*, with B as the incidence matrix of a lattice graph and the entries of d_X, d_Y, d_T given by* (5.8)–

(5.10), *renders a nonstationary field with covariance* (2.6) *locally stationary. Furthermore, the transformation exists if* $\sigma_x(x,y)$, $\sigma_y(x,y)$, *and* $\tau(t)$ *are positive; it is invertible if the derivatives* $d_{X,ij}/dx_i$, $d_{X,ij}/dx_j$, $d_{X,ij}/dy_i$, *and* $d_{X,ij}/dy_j$ *are greater than zero (and similarly for* $d_{Y,ij}$ *and* $d_{T,ij}$*).*

Proof The lemma is proved for (5.10); the proof is similar for (5.8) and (5.9). The transformation renders the field locally stationary because it satisfies (5.12)–(5.14). To prove existence, equation (5.10) is written as

$$T_i - T_j = \frac{1}{2}\ln\left(\frac{\tau_i^2+\tau_j^2}{2\tau_i\tau_j}\exp\left[\frac{2(t_i-t_j)^2}{\tau_i^2+\tau_j^2}\right]\right).$$

If τ_i and τ_j are positive, then the natural log is real, and hence the forward transformation exists. If $d_{T,ij}/dt_i, d_{T,ij}/dt_j > 0$, then $d_{T,ij}$ is a strictly increasing function of t_i and t_j. This implies that $d_{T,ij}$ is one-to-one and invertible.

Figure 5.3 shows an example of the two-dimensional spatial transformation using decorrelation scales σ_x and σ_y of the form

$$\sigma_x = \sigma_y = \sigma_0 - \mu_1 \sum_{a=1}^{A} e^{-\mu_2((x_k-x_{0,a})^2+(y_k-y_{0,a})^2)}, \tag{5.15}$$

where $a = 1, ..., A$ is the index of A nonstationarities, and $x_{0,a}$ and $y_{0,a}$ are their locations. The above function represents a varying spatial decorrelation scale where "dips" in the scale are centered at $(x_{0,a}, y_{0,a})$. On the right is the r-domain, which has a square boundary. When the r-domain boundary is mapped to the R-domain, it generates a new, curved boundary as shown in Figure 5.3. A uniform grid mapped from the R-domain to the r-domain shows how space is warped to concentrate measurements near the nonstationarity at (3,3).

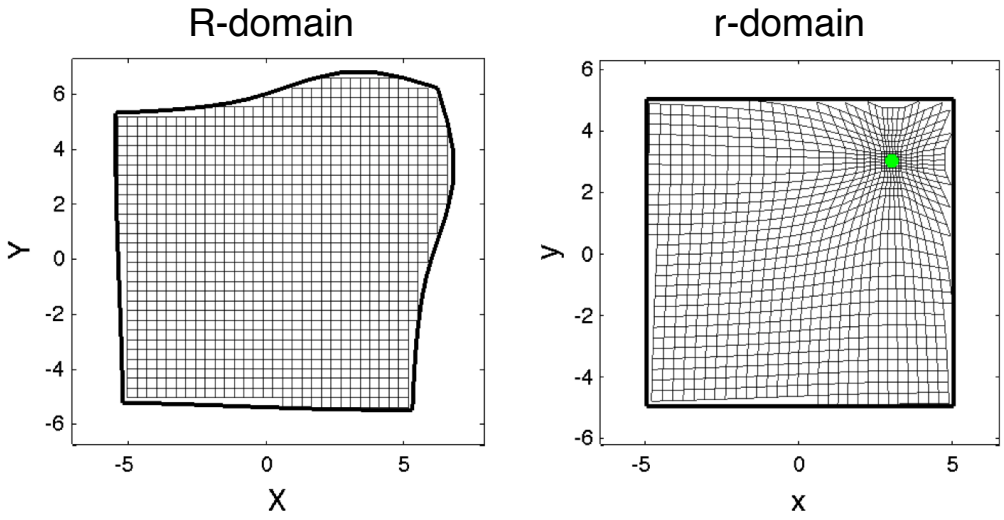


Figure 5.3: Example of the spatial transformation given by (5.12)–(5.13) with a nonstationarity of form (5.15).

5.3.2 Decoupled Decorrelation Scales

Next, consider when the spatial decorrelation scales are decoupled, i.e., $\sigma_{xi} = \sigma_x(x_i)$ and $\sigma_{yi} = \sigma_y(y_i)$. In this case, (5.8)–(5.10) reduce to

$$X_i - X_j = \left[\frac{2(x_i - x_j)^2}{\sigma_{xi}^2 + \sigma_{xj}^2} + \ln\left(\frac{\sigma_{xi}^2 + \sigma_{xj}^2}{2\sigma_{xi}\sigma_{xj}}\right)\right]^{1/2}, \tag{5.16}$$

$$Y_i - Y_j = \left[\frac{2(y_i - y_j)^2}{\sigma_{yi}^2 + \sigma_{yj}^2} + \ln\left(\frac{\sigma_{yi}^2 + \sigma_{yj}^2}{2\sigma_{yi}\sigma_{yj}}\right)\right]^{1/2}, \text{ and} \tag{5.17}$$

$$T_i - T_j = \left[\frac{2(t_i - t_j)^2}{\tau_i^2 + \tau_j^2} + \ln\left(\frac{\tau_i^2 + \tau_j^2}{2\tau_i\tau_j}\right)\right]^{1/2}. \tag{5.18}$$

For a lattice topology with decoupled decorrelation scales, $j = i + 1$, i.e., $X_j = X_i + \Delta X_i$, $x_j = x_i + \Delta x_i$ (and similarly for Y, y, T, and t). The following result provides an analytical representation of (5.16)–(5.18) in the continuous limit as the

lattice spacing goes to zero.

Theorem 5.3.2 *The following coordinate transformation for a nonstationary field with covariance* (2.6) *and decoupled decorrelation scales ensures the R-domain is locally stationary:*

$$X_k = \int_0^{x_k} \frac{1}{\sigma_x(x')} dx', \tag{5.19}$$

$$Y_k = \int_0^{y_k} \frac{1}{\sigma_y(y')} dy', \quad \textit{and} \tag{5.20}$$

$$T_k = \int_0^{t_k} \frac{1}{\tau(t')} dt'. \tag{5.21}$$

Furthermore, the transformation (5.19)–(5.21) *exists and is invertible if* $\sigma(x_k)$, $\sigma(y_k)$, *and* $\tau(t_k)$ *are positive, bounded, and continuous functions.*

Proof Assuming $j = i + 1$, i.e., $X_j = X_i + \Delta X$ and $x_j = x_i + \Delta x$, then (5.16) reduces to

$$\Delta X = \left(\frac{2(\Delta x)^2}{\sigma^2(x_i) + \sigma^2(x_i + \Delta x)} + \ln \left(\frac{\sigma^2(x_i) + \sigma^2(x_i + \Delta x)}{2\sigma(x_i)\sigma(x_i + \Delta x)} \right) \right)^{1/2}.$$

Dividing both sides by Δx and taking the limit as $\Delta x, \Delta X \to 0$ yields

$$\frac{dX_k}{dx_k} = \frac{1}{\sigma(x_k)}.$$

Integrating both sides yields the desired result. (The derivation of the temporal transformation is similar.) If σ_k and τ_k are bounded and continuous, then the integrals in both transformations exist. If the decorrelation scales are both positive, then X_k, Y_k, and T_k are increasing, invertible functions of x_k, y_k, and t_k.

Either the numerical or analytical form of the transformation presented above can be used for sampling a nonstationary spatiotemporal field, but the numerical transformation has the advantage that it is valid for any function of the decorrelation scales (as long as they are continuous and positive); however, it has (relatively) stringent conditions for invertibility. The analytical transformation is always invertible when the decorrelation scales are positive, but it is only valid when the spatial scales are decoupled. The analytical form of the transformation is of particular importance when regarding the sampling speed of the vehicles. In the next section, I show that performing control action in a domain that is transformed according to Lemma 5.3.2 drives vehicles to a constant sampling speed in the r-domain, which simplifies mission planning.

5.4 Multivehicle Coverage Control

In the previous section, a numerical transformation and its continuous limit (where the grid spacing is reduced to zero) are derived under which a nonstationary spatiotemporal field is rendered locally stationary. In this section, I design a multivehicle coverage algorithm that invokes the transformation to generate nonuniform sampling trajectories. Note, again, that the control laws developed here are used as input to a trajectory tracking and flight stability controller on the rotorcraft. I first implement a multivehicle controller that uniformly covers the R-domain. The resulting trajectories are mapped back to the r-domain.

To ensure that the R-domain is uniformly covered with no gaps or overlaps

between sensor measurements, I employ a coverage algorithm called Spanning Tree Coverage (STC), introduced in [92].[1] Intuitively, the algorithm works like a lawn-mowing algorithm along a closed path. Path closure is crucial due to the temporal nature of the field; locations are revisited to ensure that the temporal variation of the field is captured. The STC algorithm is summarized as follows. Let D be a spatial domain that is divided into square cells of size 2σ, where the constant σ will later be chosen to be the spatial decorrelation scale. Let $G(V, E)$ be a lattice graph with nodes V placed at the center of each cell and undirected edges E connecting adjacent cells. Starting in initial cell S, create a spanning tree for G, i.e., a graph with the maximal set of edges of G that contains no cycles. Then, the set of points that are a distance σ away from the spanning tree creates a closed path around the spanning tree. Since the spanning tree is formed using nodes that are 2σ apart, and the path is always $\sigma/2$ away from the tree, a vehicle with sensor size σ is guaranteed to cover the entire domain with no gaps or overlaps in the coverage. As an example, consider the spatial scale function given by (5.15) and the sampling boundary in Figure 5.3. Figure 5.4 shows the spanning tree and generated path in the R-domain (left) and the r-domain (right).

Next, I seek to steer a set of mobile vehicles to motion along the path generated

[1] Any algorithm that generates a single closed path that guarantees spatial coverage could potentially be used in lieu of the STC algorithm. I select this algorithm as opposed to classical lawn-mowing or milling [93], because it works for any simply connected spatial domain and allows us to apply intuition from sampling of stationary fields to nonstationary fields because the paths generated by the algorithm are closed.

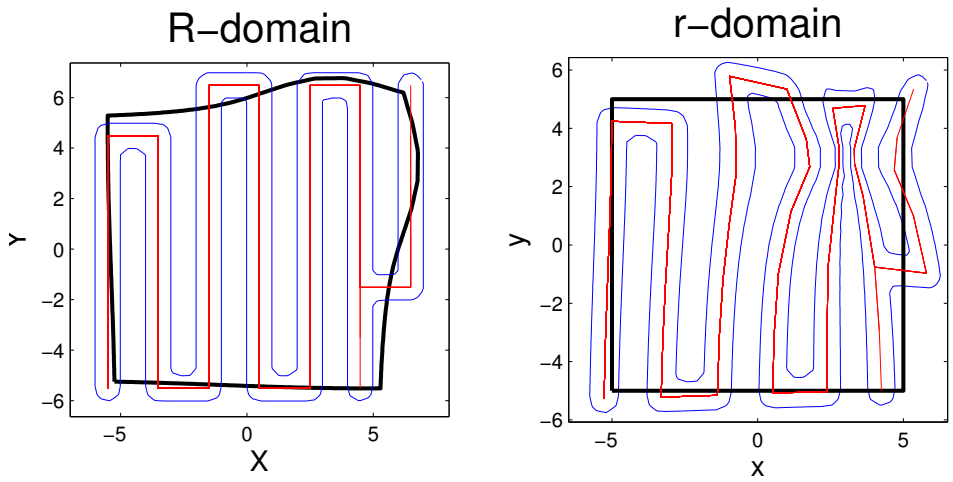


Figure 5.4: Example of the spatial transformation (5.12)–(5.13) with the generated STC path.

by the STC algorithm. Assume there is a mapping that converts the path coordinate θ_k to the location on the path (X_k^p, Y_k^p), i.e., $X_k^p = f(\theta_k) = f_k$ and $Y_k^p = g(\theta_k) = g_k$, where f_k and g_k are differentiable everywhere. Given the path from the STC algorithm, f_k and g_k can be found numerically using a lookup table from the path coordinate to X_p and Y_p. The lookup table is created by discretizing the path and associating with each discrete point a value of θ_k such that θ_k increases in the counterclockwise direction. Consider the spatial coordinates (l_k, θ_k) in the R-domain, where l_k is the shortest distance between the vehicle and the path and θ_k is the distance along the path from an arbitrary fixed point θ_0. These coordinates are illustrated in Figure 5.5.

Using the (l_k, θ_k) coordinates I design a distributed gradient controller to stabilize a set of vehicles to the desired cross-path and along-path motion. Let the

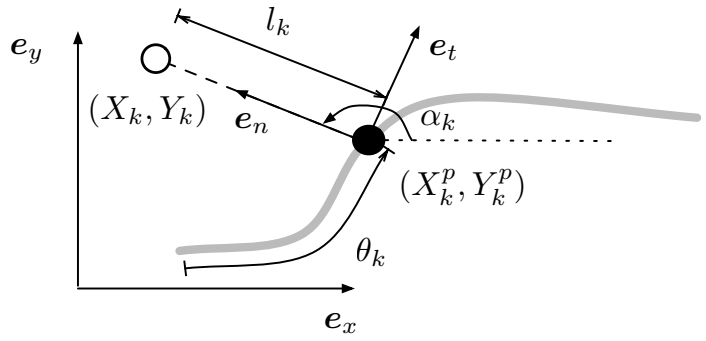


Figure 5.5: Control coordinates in the R-domain.

kinematics of the kth vehicle be

$$\frac{dl_k}{dT_k} = u_{l,k} \quad \text{and} \quad \frac{d\theta_k}{dT_k} = u_{\theta,k}. \tag{5.22}$$

Accordingly, there is the following result.

Theorem 5.4.1 *Consider the vehicle dynamics* (5.22)*, where* $k = 1, ..., N$*, along a smooth, closed path. Assuming that* l_k *and* θ_k *are unique, choosing*

$$u_{l,k} = -\frac{dV}{dl_k} \quad \textit{and} \quad u_{\theta,k} = s_0 - \frac{dS}{d\theta_k}, \tag{5.23}$$

where $V = \frac{1}{2}\sum_{k=1}^{N} l_k^2$ *and* S *is given by* (5.2)*, stabilizes the set of splay formations along the path with constant speed* s_0*.*

Proof The dynamics of l_k and θ_k are decoupled, so their stability properties are determined independently. The closed-loop θ_k dynamics are studied in [91], which establishes that the system converges to the critical set of S, including the set of splay formations. For the l_k dynamics, the derivative of V with respect to time T_k is negative semi-definite and zero only when l_k is zero. Application of the invariance principle from [73] completes the proof.

Note that θ_k may not be uniquely defined depending on $f(\theta_k)$ and $g(\theta_k)$. The STC algorithm gives paths that resemble lawnmower patterns (as shown in Figure 5.4), so the non-unique points sit on lines that are equidistant from two tracks. If a vehicle does happen onto one of these locations, the value of θ_k that is closest to the value of θ_k at the previous time step is used to ensure continuity. I have found, in simulation, that vehicles pass unaffected through these problematic points when using this strategy.

In order to implement (5.22), the control is converted to the (X_k, Y_k) coordinates. I start by defining the normal vector $\mathbf{e}_n$ (using a Frenet-Serret path frame as in [94]) at the kth location:

$$\mathbf{e}_n = -\frac{dg_k}{d\theta_k}\mathbf{e}_x + \frac{df_k}{d\theta_k}\mathbf{e}_y,$$

where $\mathbf{e}_x$ and $\mathbf{e}_y$ are orthogonal unit vectors in an inertial frame. Using Figure 5.5,

$$X_k = X_k^p + l_k \cos\alpha_k \quad \text{and} \quad Y_k = Y_k^p + l_k \sin\alpha_k, \tag{5.24}$$

where α_k is the angle between $\mathbf{e}_n$ and $\mathbf{e}_x$.

Taking the derivative of (5.24) with respect to T_k yields

$$\frac{dX_k}{dT_k} = \frac{df_k}{d\theta_k}\frac{d\theta_k}{dT_k} + \frac{dl_k}{dT_k}\cos\alpha_k - l_k\frac{d\alpha_k}{d\theta_k}\frac{d\theta_k}{dT_k}\sin\alpha_k \text{ and}$$
$$\frac{dY_k}{dT_k} = \frac{dg_k}{d\theta_k}\frac{d\theta_k}{dT_k} + \frac{dl_k}{dT_k}\sin\alpha_k + l_k\frac{d\alpha_k}{d\theta_k}\frac{d\theta_k}{dT_k}\cos\alpha_k,$$

where

$$\frac{d\alpha_k}{d\theta_k} = -\left(1 + \left(\frac{\frac{df_k}{d\theta_k}}{\frac{dg_k}{d\theta_k}}\right)\right)^{-1}\left[\frac{\frac{dg_k}{d\theta_k}\frac{d^2f_k}{d\phi^2} - \frac{df_k}{d\theta_k}\frac{d^2g_k}{d\theta_k^2}}{\left(\frac{dg_k}{d\theta_k}\right)^2}\right].$$

The control in the R-domain is

$$\frac{dX_k}{dT_k} = \frac{df_k}{d\theta_k} u_{\theta,k} + u_{l,k} \cos\alpha_k - l_k \frac{d\alpha_k}{d\theta_k} u_{\theta_k} \sin\alpha_k = u_{X,k}, \tag{5.25}$$

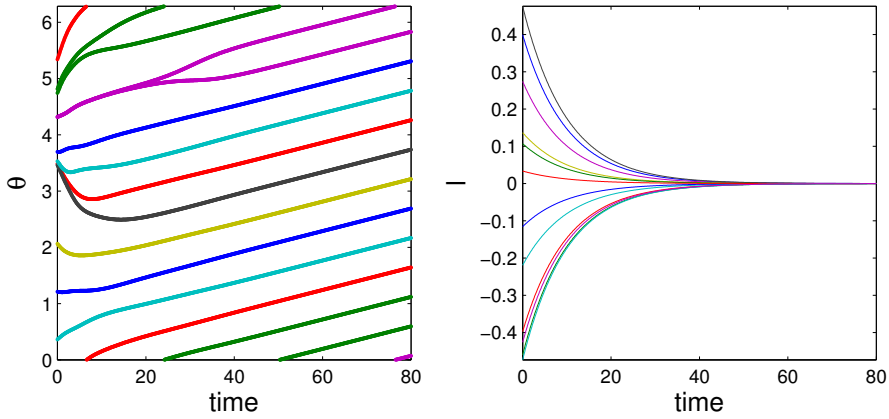
$$\frac{dY_k}{dT_k} = \frac{dg_k}{d\theta_k} u_{\theta,k} + u_{l,k} \sin\alpha_k + l_k \frac{d\alpha_k}{d\theta_k} u_{\theta,k} \cos\alpha_k = u_{Y,k}. \tag{5.26}$$

Table 5.1 contains a pseudocode version of the Multivehicle Coverage Control Algorithm. The inputs are the boundary of the r-domain, the decorrelation scales and the initial positions of all the vehicles. The output is the set of trajectories that the vehicles follow in the r-domain.

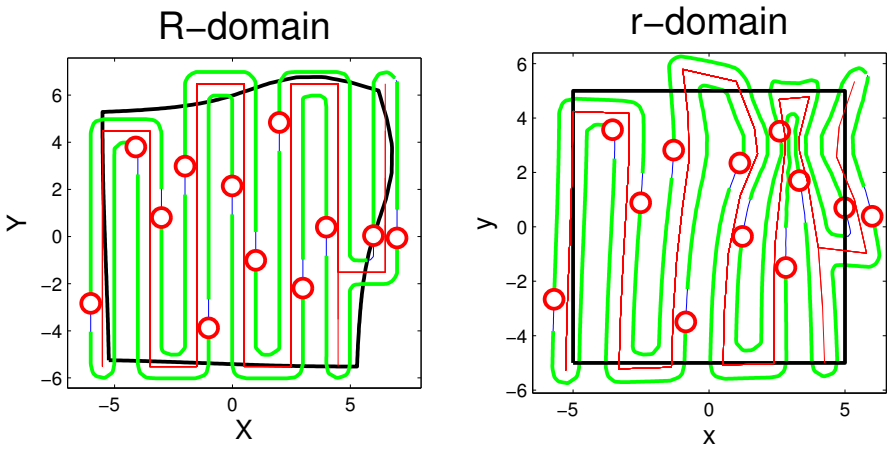
Table 5.1: Multivehicle Coverage Control Algorithm

Inputs: Boundary of r-domain, $\sigma_x(x_k, y_k)$, $\sigma_y(x_k, y_k)$, $\tau(t_k)$, vehicle initial conditions for all $k = 1, \ldots, M$
Outputs: Vehicle trajectories in the r-domain
1: Calculate R-domain boundary using (5.12)–(5.14)
2: Transform initial conditions to R-domain using (5.12)–(5.14)
3: Create lattice graph within boundary of R-domain
4: Use STC algorithm to construct path in the R-domain
5: Transform R-domain initial conditions to path coordinates (θ_k, l_k)
6: Integrate (5.22) using gradient control (5.23)
7: Transform trajectories from path coordinates (l_k, θ_k) in R-domain to (X_k, Y_k) coordinates
8: Transform trajectories to r-domain by numerically inverting using (5.12)–(5.14)

To illustrate the algorithm, I performed a simulation of twelve vehicles with random initial conditions. Figure 5.6(a) shows the values of θ_k and l_k for each vehicle, which reach a splay formation along the path. Figure 5.6(b) shows the steady-state trajectories in the R- and r-domains. The red circles represent vehicles and the green lines show their recent trajectories. In steady-state, in the R-domain (on the left of Figure 5.6(b)), it is seen by examining the green tail that the vehicles reach a splay formation around the blue path. The trajectories in the R-domain are



(a) R-domain path coordinates



(b) Steady-state trajectory

Figure 5.6: Coverage algorithm example using (5.25) and (5.26).

transformed back to the r-domain by numerically inverting transformation (5.12)–(5.14), shown on the right of Figure 5.6(b). In the r-domain, vehicles cluster their measurements in the area of the dip to ensure that the full domain is covered, even in areas where the spatial and temporal decorrelation scales are small.

In the special case where the spatial scales are decoupled, the following theorem shows that this coverage strategy corresponds to preserving the along-path steady-state sampling speed in the r-domain.

Theorem 5.4.2 *If the spatial decorrelation scales are decoupled and the R-domain coordinates are given by* (5.19)–(5.20)*, then the coverage-control strategy in Proposition 5.4.1 preserves the steady-state sampling speed* $s_k = s_0$ *for all* k*.*

Proof In steady state, $u_{l,k} = l_k = 0$ and $u_{\phi,k} = s_0$. The controls $u_{X,k}$ and $u_{Y,k}$ become

$$u_{X,k} = s_0 \cos\alpha_k \quad \text{and} \quad u_{Y,k} = s_0 \sin\alpha_k.$$

In the R-domain,

$$\dot{x} = \frac{dx_k}{dX_k}\frac{dX_k}{dT_k}\frac{dT_k}{dt_k} = s_0 \cos\alpha_k \frac{\sigma_x(x_k)}{\tau(t_k)},$$
$$\dot{y} = \frac{dy_k}{dY_k}\frac{dY_k}{dT_k}\frac{dT_k}{dt_k} = s_0 \sin\alpha_k \frac{\sigma_y(y_k)}{\tau(t_k)}.$$

The speed and direction of motion of the kth vehicle are

$$v_k = \frac{s_0}{\tau(t_k)}\sqrt{\sigma_x^2(x_k)\cos^2\alpha_k + \sigma_y^2(y_k)\sin^2\alpha_k}, \tag{5.27}$$

and its direction of motion is

$$\beta_k = \arctan\left(\frac{\dot{y}_k}{\dot{x}_k}\right) = \arctan\left(\frac{\sigma_y(y_k)}{\sigma_x(x_k)}\tan\alpha_k\right). \tag{5.28}$$

The sampling speed for a vehicle in a nonstationary field is

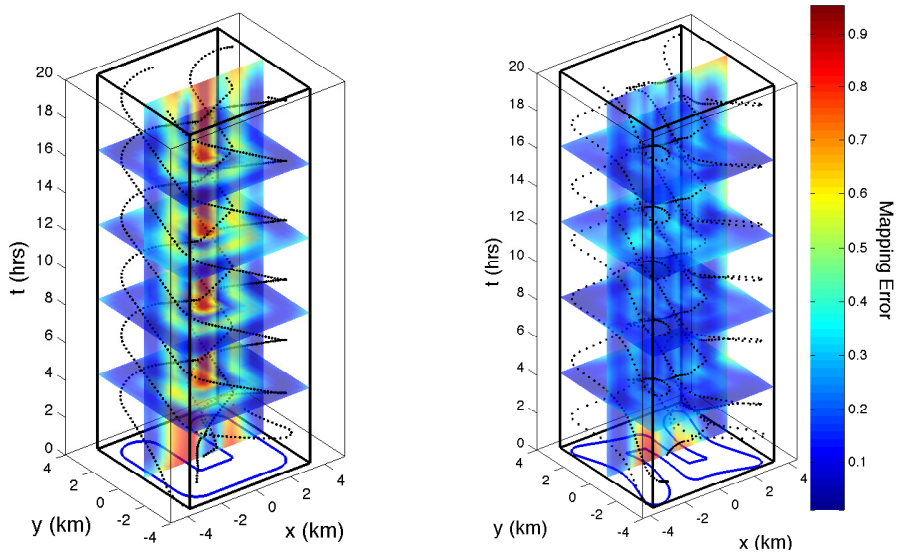
$$s_k = v_k \frac{\tau(t_k)}{\sigma_c(x_k, y_k)}, \tag{5.29}$$

where σ_c is the cross section of the measurement footprint along the direction of motion of the vehicle. Using (5.28) and trigonometric relations, the along-path spatial decorrelation scale is

$$\sigma_c = \sqrt{\sigma_x^2(x_k)\cos^2\alpha_k + \sigma_y^2(y_k)\sin^2\alpha_k}. \tag{5.30}$$

Substituting (5.27) and (5.30) into (5.29) shows that $s_k = s_0$ for all k, which completes the proof.

In steady-state, the vehicles travel around a closed path, so the number of vehicles can be selected as described in Section 5.2. In particular, the steady-state sampling speed is chosen so that the maximum required vehicle speed does not exceed the vehicle speed limit. Figure 5.7 shows a simulated sampling task and the resulting mapping error. There is a dip in the decorrelation scales centered at $x_{0,1} = y_{0,1} = 0$ km, and $t = 12$ hours. Five vehicles travel along a closed path in the r-domain. The blue curve on the $t = 0$ plane shows the spatial path that the vehicles track. The average mapping error is 0.271 using uniform coverage (Figure 7(a)) and 0.178 when using the nonstationary sampling algorithm (Figure 7(b)). Likewise, the maximum error is reduced from 0.983 to 0.768.



(a) (b)

Figure 5.7: Mapping error of a 3D nonstationary spatiotemporal field using (a) uniform sampling and (b) the nonstationary sampling algorithm in Table 1.

5.5 Lag Space for Formation Control

The covariance function is dependent on the spatial separation (lag) between r_i and r_j, defined as $\alpha_k = r_i - r_j$. In order to quantify the performance of a given mapping mission the mapping error is calculated, for which the spatial and temporal decorrelation scales of the field are needed. However, this means that I want the vehicles to be placed at a variety of locations so that the lags between vehicles completely cover the range in which the true decorrelation scale lies. This desire is contrary to the goal of generating an accurate map of the field, because optimal mapping requires all of the lags to be at one location; specifically, mapping vehicles should be placed in a splay formation around the path so that they are

spaced equally [58, 60]. Thus, there is a need to explore the field to determine the decorrelation scales before exploiting the knowledge of the scales to optimally cover the field.

Figure 5.8 shows hows a plot of the covariance function (2.6) in lag space, where the temporal portion of the function has been neglected for simplicity. Lag space is defined as the domain of distances between vehicles. The blue + shows where all the lags should be placed for optimal mapping. For optimal mapping, the vehicles are in splay, i.e., at a lag of $2\pi/N$, where $N = 6$ is the number of vehicles. The red circles show where they should be placed to estimate the spatial decorrelation scale if there is no prior information about σ, i.e., the vehicles are spaced at specific intervals so that they are in splay in lag space. The decorrelation scales must first be determined before transitioning the vehicles into a splay formation for accurate mapping. Note, for a stationary field it is not necessary to cover the field while estimating the scales since the scales are the same everywhere, but it does become important for mapping nonstationary fields.

To optimally determine the decorrelation scale of the fields, the vehicles need to be in a splay formation in lag space. Figure 5.9 shows the difference between lag splay (Figure 5.9(a)) and vehicle splay (Figure 5.9(b)) on a closed path. For determining the decorrelation scale, it is also desirable to cover the entire domain. Thus, in this section I design a controller to drive vehicles to lag splay on a circle that is off-center in a rotating reference frame, i.e., the circular path to which the vehicles are driven rotates around the origin as depicted in Figure 5.10. In the inertial frame, the vehicles follow a spirograph trajectory. The end result is that the

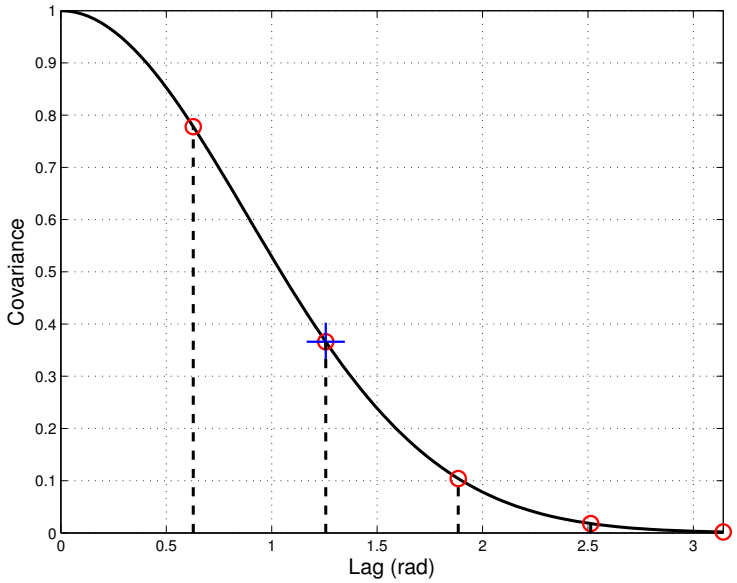
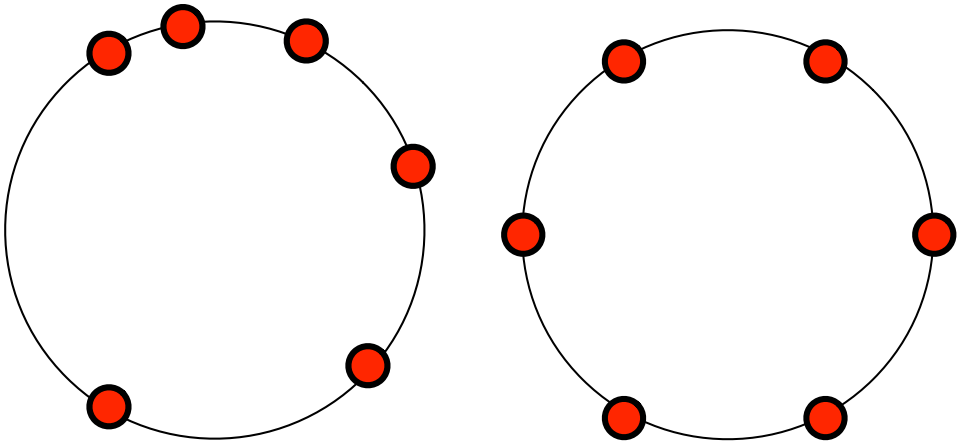


Figure 5.8: The covariance function (2.6) depicted in lag space. The red circles depict where lags should be placed for scale estimation and the blue plus shows where the lags should be for mapping.



(a) Lag splay

(b) Vehicle splay

Figure 5.9: Difference between (a) lag splay and (b) vehicle splay.

vehicles maintain lag splay, but also cover the space due to the rotation of the path around the origin.

To stabilize to the lag splay formation on a circle, the vehicles use a controller that is similar to that used for following the STC generated path. The cross path controller is the same as the one proposed in Theorem 5.4.1. However $u_{\theta,k}$ must be modified to achieve lag splay. The following controller is used in the along-track direction to stabilize to lag splay

$$u_{\theta,k} = -k \sin(\alpha_k - \alpha_{d,k}), \tag{5.31}$$

where $\alpha_{d,k}$ is the desired lag between vehicles $k+1$ and k. This drives the vehicles to a formation where the lags between vehicles are in splay as depicted in Figure 5.9(a).

Both the along-track and cross-track controllers must compensate for the rotation of the circle around a fixed point, denoted $r_c = (x_c,\ y_c)$, at rotation rate ω_c. To do this (5.25)–(5.26) are modified as follows:

$$\dot{X}_k = u_{X,k} + \omega_c ||r_c - r_k|| (x_c - X_k) \tag{5.32}$$

$$\dot{Y}_k = u_{Y,k} + \omega_c ||r_c - r_k|| (y_c - Y_k) \tag{5.33}$$

The second term in the above equations account for the rotation of the circular path.

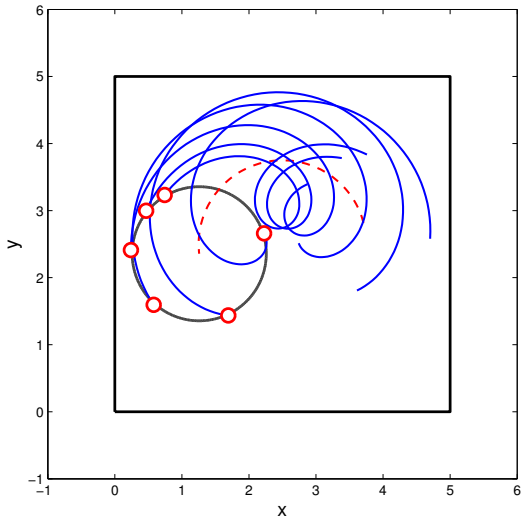


Figure 5.10: Spirograph formation used for maintaining lag splay while covering the domain.

5.6 Bayesian Estimation of Decorrelation Scales

I use a recursive Bayesian filter to estimate the decorrelation scales at each time step [95]. Consider the likelihood function [96]

$$P(z|\sigma,\tau) = e^{z^T M^{-1}(\sigma,\tau) z}, \tag{5.34}$$

where $z = [z_1, \cdots, z_D]^T$ contains all of the available measurements and $M^{-1}(\sigma,\tau)$ is the measurement covariance (see section 2.2. The likelihood function (5.34) assumes that the random field A is a Gaussian process with zero mean; it is commonly used in the prediction of Gaussian processes from collected measurements [96]. The maximum likelihood estimate of σ and τ, denoted $\hat{\sigma}$ and $\hat{\tau}$, is the mode of the posterior distribution [96]

$$P(\sigma,\tau|z) = aP(z|\sigma,\tau)P(z), \tag{5.35}$$

where $P(z)$ is the prior and $a = \int P(z|\sigma,\tau)P(z)dz$. The marginal probability of σ over all $\tau_{min} \leq \tau \leq \tau_{max}$ is

$$P_\sigma = \int_{\tau_{min}}^{\tau_{max}} P(z|\sigma,\tau)d\tau, \tag{5.36}$$

For details on the recursive Bayesian estimator, please see Section 2.1.

5.7 Optimal Exploration vs. Exploitation

A metric is needed in order to switch from decorrelation scale estimation to mapping. The metric used in this dissertation is based on the Kullbach–Leibler divergence between the posterior distribution of the scale estimate and a Dirac delta function. The metric is

$$K = \frac{\int P_\sigma \log\left(P_\sigma/P_\delta\right)d\sigma}{\int P_\sigma \log\left(P_\sigma/P_1\right)d\sigma}, \tag{5.37}$$

where $P_\delta = \delta(\sigma - \hat{\sigma})$ is a Dirac delta function centered at the mode of P_σ and P_1 is a uniform distribution. (A narrow Gaussian distribution is used to approximate P_δ because locations of zero probability yields a singular KL divergence.) Equation (5.37) is the KL divergence between P_σ and P_δ normalized by the KL divergence between P_σ and a uniform distribution. Thus, if there is little or no known information about the decorrelation scales, the numerator and denominator of (5.37) will be similar and K will be close to one, whereas if P_σ is narrow, then the numerator will be much less than the denominator and $K \approx 0$. To switch from scale estimation to mapping, (5.37) is compared to a threshold, i.e., when $K < K_{thresh}$, the formation switches to mapping using the STC algorithm with the estimated scale.

Figures 5.11 and 5.12 show snapshots a simulation of the estimation/mapping system for a 2D stationary field. In each figure, the top left shows the mapping error (2.7), the top right shows the posterior distribution over time, the bottom left shows the KL divergence metric, and the bottom right shows the quantities

$$P_\theta = \frac{1}{N}\sum_{j=1}^{N}(\exp(i\theta_j)) \tag{5.38}$$

$$P_{lag} = \frac{1}{N}\sum_{j=1}^{N}(\exp(il_j)), \tag{5.39}$$

where l_j is the jth lag. Equations (5.38) and (5.39) are when the vehicles are in splay and lag splay respectively. Figure 5.11 shows the simulation at time step 89, when the vehicles are in explore mode, and Figure 5.12 shows when the vehicles at the final time step when they are in exploit mode.

There are several interesting features of the algorithm seen in the simulation. First is that there is a sharp decrease in the mapping error once the vehicles switch from explore to exploit, almost a 50% decrease in error. The other apparent feature is that the slope of the KL divergence metric is higher when in explore mode. This implies that starting with the vehicles in lag splay means that the spatial decorrelation scale will be estimated faster than it would be in explore mode.

The extension of the algorithm to nonstationary fields is natural. There are only two changes. First. the recursive Bayesian estimator is now estimating several parameters. In the case of the dip function, this would include the base decorrelation scale, the cartesian location of the dip, the strength of the dip, and its width. The second change is that once the KL divergence metric lowers below the threshold, the vehicles switch to an STC formation using the same formulation as that for a

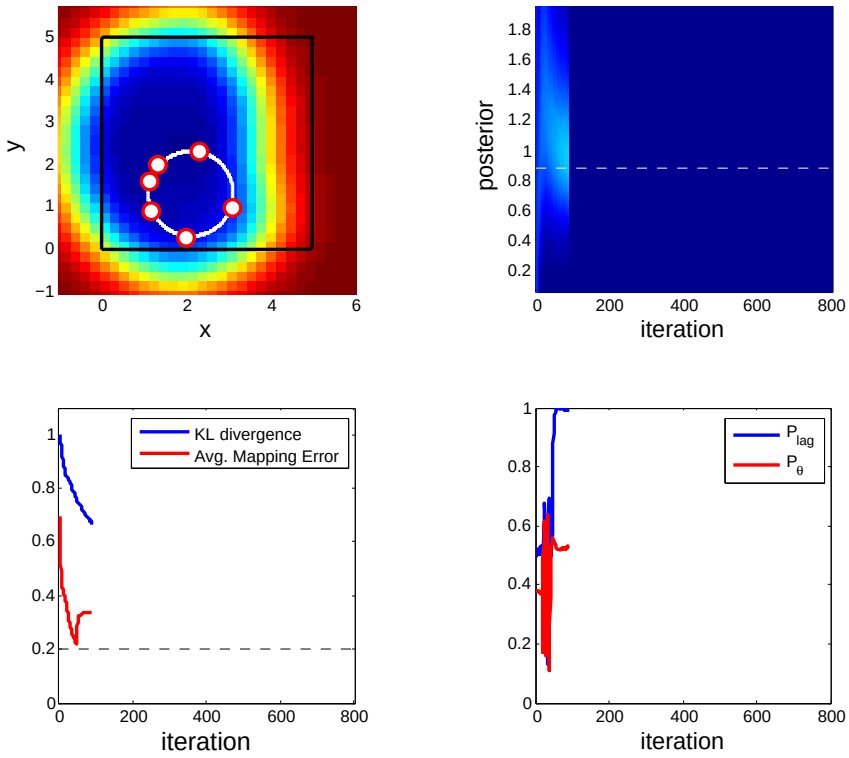


Figure 5.11: Snapshot of explore and exploit simulation for a stationary field at time step 89.

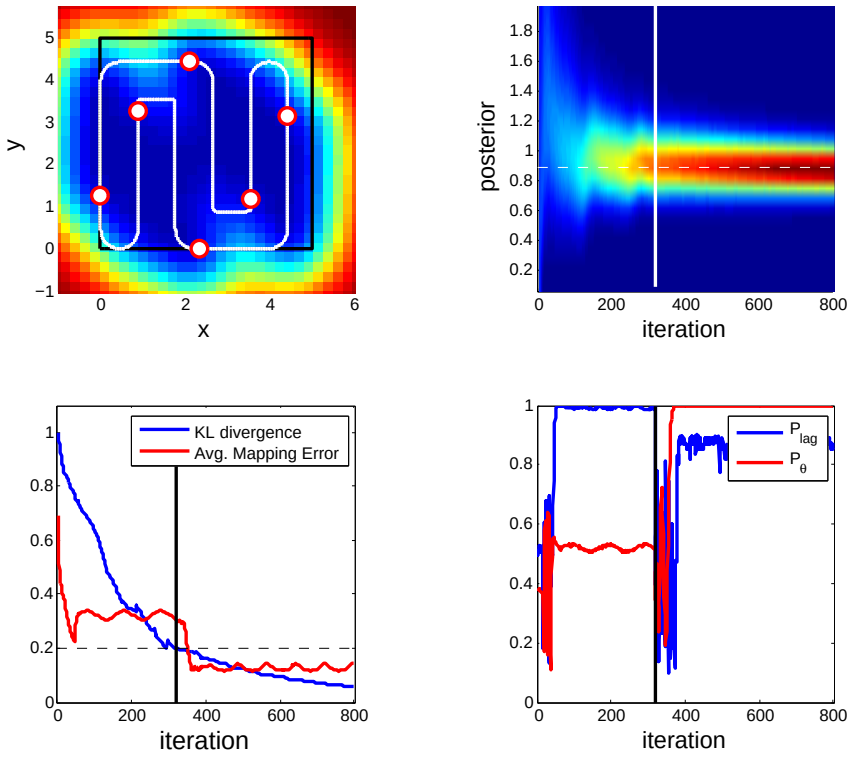


Figure 5.12: Snapshot of explore and exploit simulation for a stationary field at time step 800.

field with known decorrelation scales.

Chapter 6: Conclusion

In this dissertation, I discussed and developed capabilities for mulit-agent teams of autonomous unmanned rotorcraft for sampling spatiotemporal processes. The dissertation treats these rotorcraft as sensor packages capable of hover and fast transit. The purpose of this work is to design robust algorithms for sampling spatiotemporal processes in the presence of strong winds using multi-rotor systems.fdsa

6.1 Summary of Results

I design estimators and controllers to guide a multi-agent aerial team to optimally cover a nonstationary spatiotemporal field, where the decorrelation scales of the field can change in space and time. This section of work uses the concept of exploration versus exploration, where vehicles first explore the region to determine the decorrelation scales, then switch to optimal mapping based on the results of the scale estimation. The exploration strategy moves the vehicles in a rotating spirograph trajectory with the vehicle spacing decided to best sample the field for scale determination. The mapping strategy involves using the estimate of the decorrelation scale to create a persistent path through the domain using a spanning tree coverage algorithm. The developed algorithm works for both stationary and

nonstationary 2D spatiotemporal fields modeled as a Gaussian process.

Analysis of the explore vs. exploit mapping algorithms shows that the team benefits from first determining the decorrelation scales using a spirograph formation before transitioning to the spanning tree based coverage algorithm. This is apparent by examining the average mapping error, which is relatively high while determining the scales, but drops rapidly once the vehicles transition to mapping.

I also created a physics-inspired algorithms for optimal exploration of a 2D domain for the purpose of detecting an unknown number of moving targets using a low-quality camera as the sensor. This work relies on the Likelihood Ratio Tracker, a modified version of a recursive Bayesian estimator, to detect targets based on binary measurements from a camera. Vehicle motion is coordinated to maximize the probability of target detection. The algorithm was tested both in simulation and using an experimental motion capture based testbed.

A closer inspection of the theoretical properties of the target detection strategy coupled with the physics based controller showed the optimality of moving the vehicles along the gradient of the LRT surface. In particular, moving the vehicles along the gradient of the surface maximizes the expected gain in information, i.e. the mutual information, for the next time step.

With both sampling problems discussed in this dissertation, there is still the issue of robust flight. As a first step in developing robust flight controllers, I designed several controllers for flight stability in windy environments. Specifically, I used nonlinear feedback linearization to create a layered flight controller to compensate for external wind fields. The controller, which was tested in simulation, was shown

to reduce the overshoot and settling time of a rotorcraft that is perturbed by an external flowfield. This is true even in the case when the wind speed must be estimated using a recursive Bayesian estimator.

In addition to the stability controllers, I also designed a artificial potential based controller for the purpose of avoiding nearby multi-rotor systems by detecting their downwash. This controller uses a recursive Bayesian estimator to determine the position of a nearby rotorcraft by measuring its downwash. The downwash avoidance algorithm was tested in a motion capture testbed using pitot tube type pressure sensors to determine the airspeed. Experiments showed that the controller allowed the vehicle to successfully avoid nearby rotorcraft.

6.2 Suggestions for Future Work

While this dissertation has addresses several problems in sampling and estimation using multi-rotor aerial vehicles, it by no means completely solves all related issues. In particular, there is much experimental validation to be performed on the flight stability algorithms, as well as augmenting the algorithms to use other available sensors, such as inertial measurement units or readily available optical flow sensors.

In the area of sampling nonstationary Gaussian processes, there is work to be done in distributing the calculation of the map and the associated mapping error. Additionally, while the STC algorithm has proven useful, there are other coverage strategies, such as optimal foraging or information gathering, that may also lend

themselves easily for this purpose.

For the target detection work, the presented algorithms is distributed with an all-to-all communication architecture. As such, it would be interesting to develop a strategy for distributively merging measurements taken by all the vehicles using a different architecture, such as range limited or nearest neighbor communication.

Bibliography

[1] A. Girard, A. Howell, and J. Hedrick. Border patrol and surveillance mission using multiple unmanned air vehicles. In *Proceedings of the 43rd IEEE Conference on Decision and Control*, pages 620–625, December 2004.

[2] J.T.K. Ping, A. E. Ling, T. J. Quan, and C. Y. Dat. Generic unmanned aerial vehicle (UAV) for civilian application-a feasibility assessment and market survey on civilian application for aerial imaging. In *2012 IEEE Conference on Sustainable Utilization and Development in Engineering and Technology*, pages 289 –294, oct. 2012.

[3] M. Belkheri, A. Rabhi, A. Hajjaji, and C. Pergard. Different linearization control techniques for a quadrotor system. In *2nd International Conference on Communications, Computing and Control Applications,*, pages 1–6, December 2010.

[4] B. Zhu and W. Huo. Trajectory linearization control for a quadrotor helicopter. In *8th IEEE International Conference on Control and Automation*, pages 34 –39, June 2010.

[5] R.C. Leishman, J.C. Macdonald, R.W. Beard, and T.W. McLain. Quadrotors and accelerometers: State estimation with an improved dynamic model. *IEEE Control Systems*, 34(1):28–41, Feb 2014.

[6] D. Mellinger and V. Kumar. Minimum snap trajectory generation and control for quadrotors. In *IEEE International Conference on Robotics and Automation*, pages 2520 –2525, may 2011.

[7] A. Benallegue, A. Mokhtari, and L. Fridman. Feedback linearization and high order sliding mode observer for a quadrotor UAV. In *International Workshop on Variable Structure Systems*, pages 365 –372, June 2006.

[8] E. de Vries and K. Subbarao. Backstepping based nested multi-loop control laws for a quadrotor. In *11th International Conference on Control Automation Robotics Vision*, pages 1911 –1916, Dec. 2010.

[9] G. Hoffmann, H. Huang, S. Waslander, and C. Tomlin. *Quadrotor Helicopter Flight Dynamics and Control: Theory and Experiment*. AIAA, 2007.

[10] C. Powers, D. Mellinger, A. Kushleyev, B. Kothmann, and V. Kumar. Influence of aerodynamics and proximity effects in quadrotor flight. In *Proceedings of the International Symposium on Experimental Robotics*, June 2012.

[11] K. Alexis, G. Nikolakopoulos, and A. Tzes. Experimental model predictive attitude tracking control of a quadrotor helicopter subject to wind-gusts. In *18th Mediterranean Conference on Control Automation, 2010*, pages 1461 – 1466, June 2010.

[12] K. Alexis, G. Nikolakopoulos, and A. Tzes. Constrained-control of a quadrotor helicopter for trajectory tracking under wind-gust disturbances. In *MELECON 2010*, pages 1411 –1416, April 2010.

[13] D. Alexander and S. Vogel. *Nature's Flyers: Birds, Insects, and the Biomechanics of Flight*. Johns Hopkins University Press, October 2004.

[14] M. Gewecke and Martin Woike. Breast feathers as an air-current sense organ for the control of flight behaviour in a songbird (carduelis spinus). In *Zeitschrift fr Tierpsychologie*, volume 47, pages 293–298, 1978.

[15] R. Brown and M. Fedde. Airflow sensors in the avian wing. In *Journal of Experimental Biology*, volume 179, pages 13–30, 1993.

[16] S. Herwitz, K. Allmendinger, R. Slye, S. Dunagan, B. Lobitz, L. Johnson, and J. Brass. Nighttime UAV vineyard mission: Challenges of see-and-avoid in the NAS. In *Proceedings of the AIAA 3rd Unmanned Unlimited Conference, Workshop and Exhibit*, pages 1–6, September 2004.

[17] R. Beard, D. Kingston, M. Quigley, D. Snyder, R. Christiansen, W. Johnson, T. McLain, and M. Goodrich. Autonomous vehicle technologies for small fixed wing UAVs. In *AIAA Journal of Aerospace Computing, Information, and Communication*, volume 2, page 92, January 2005.

[18] R. Hirokawa, D. Kubo, S. Suzuki, J. Meguro, and T. Suzuki. Small UAV for immediate hazard map generation. In *AIAA Infotech@Aerospace Conf*, May 2007.

[19] F. Hsiao, Y. Ding, C. Chuang, C. Lin, and Y. Huang. The design of a small UAV system as a testbed of formation flight. In *AIAA Infotech@Aerospace Conf*, March 2011.

[20] R. Eubank, E. Atkins, and D. Macy. Autonomous guidance and control of the flying fish ocean surveillance platform. In *AIAA Infotech@Aerospace Conf*, April 2009.

[21] N. Rasmussen, B. Morse, and C. Taylor. Fixed-wing, mini-UAV system for aerial search operations. In *AIAA Guidance Navigation and Control Conference and Exhibit*, August 2007.

[22] P. Xie, A. Flores-Abad, G. Martinez, and O. Ma. Development of a small UAV with autopilot capability. In *Proceedings of the AIAA Atmospheric Flight Mechanics Conference*, August 2011.

[23] M. Patel, Z. Sowle, T. Corke, and C. He. Autonomous sensing and control of wing stall using a smart plasma slat. In *Proceedings of the 44th AIAA Aerospace Sciences Meeting*, January 2006.

[24] P.Bowles and T. Corke. Stall detection on a leading-edge plasma actuated pitching airfoil utiliizing onboard measurement. In *Proceedings of the 47th Aerospace Sciences Meeting*, January 2009.

[25] Y. Xu, F. Jiang, S. Newbern, A. Huand, C. Ho, and Y. Tai. Flexible shear-stress sensor skin and its application to unmanned aerial vehicles. In *Sensors and Actuators A: Physical*, volume 105, pages 321–329, 2003.

[26] C. Gorsjean, G. Lee, W. Hong, Y. Tai, and C. Ho. Micro balloon actuators for aerodynamic control. In *The Eleventh Annual International Workshop on Micro Electro Mechanical Systems*, pages 166–171. IEEE, January 1998.

[27] Air Force Office of Scientific Research (AFOSR). Multidisciplinary university research initiative (muri).

[28] W. Barnwell. Flight Control Using Distributed Actuation and Sensing. Master's thesis, North Carolina State University, USA, 2003.

[29] S. Lion. Control Authorities of a distributed Actuation and Sensing Array on a Blended-Wing-Body Uninhabited Aerial Vehicle. Master's thesis, North Carolina State University, USA, 2007.

[30] W.R. Blanding, P.K. Willett, and Y. Bar-Shalom. Multiple target tracking using maximum likelihood probabilistic data association. In *2007 IEEE Aerospace Conference*, pages 1–12, March 2007.

[31] P. Willett, Y Ruan, and R. Streit. Pmht: problems and some solutions. *IEEE Transactions on Aerospace and Electronic Systems*, 38(3):738–754, Jul 2002.

[32] L. D. Stone, C. A. Barlow, and T. L. Corwin. *Bayesian Multiple Target Tracking*. Artech House, 1999.

[33] D.E. Clark and J. Bell. Bayesian multiple target tracking in forward scan sonar images using the phd filter. *IEEE Proceedings of Radar, Sonar and Navigation*, 152(5):327–334, October 2005.

[34] C-C. Huang and S-J Wang. A Bayesian hierarchical framework for multitarget labeling and correspondence with ghost suppression over multicamera surveillance system. *IEEE Transactions on Automation Science and Engineering*, 9(1):16–30, Jan 2012.

[35] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[36] S. Bandyopadhyay and Soon-Jo Chung. Distributed estimation using bayesian consensus filtering. In *American Control Conference*, pages 634–641, June 2014.

[37] M. A. Peot, T. W. Altshuler, A. Breiholz, R. A. Bueker, K. W. Fertig, A. T. Hawkins, and S. Reddy. Planning sensing actions for UAVs in urban domains. volume 5986, pages 59860J–59860J–10, 2005.

[38] F. Rafi, S. Khan, K. Shafiq, and M. Shah. Autonomous target following by unmanned aerial vehicles. volume 6230, jun 2006.

[39] J. Kim and Y. Kim. Moving ground target tracking in dense obstacle areas using uavs. In *Proceedings of the 17th IFAC World Congress*, 2008.

[40] V. Isler and M. Magdon-Ismail. Sensor selection in arbitrary dimensions. *IEEE Transactions on Automation Science and Engineering*, 5(4):651–660, Oct 2008.

[41] G. M. Hoffmann and C.J. Tomlin. Mobile sensor network control using mutual information and particle filters. *IEEE Transactions on Automatic Control*, 55(1):32–47, Jan 2010.

[42] Chris M Kreucher, Keith D Kastella, and Alfred O Hero III. Information-based sensor management for multitarget tracking. In *Optical Science and Technology, SPIE's 48th Annual Meeting*, pages 480–489, 2003.

[43] E. Cuevas, A. Echavarria, and M. A. Ramirez-Ortegon. An optimization algorithm inspired by the states of matter that improves the balance between exploration and exploitation. *Applied Intelligence*, 40(2):256–272, 2014.

[44] T. B. Apker and M. A Potter. Robotic swarms as solids, liquids and gasses. In *AAAI Fall Symposium: Human Control of Bioinspired Swarms*, 2012.

[45] W. M. Spears, D. F. Spears, J. C. Hamann, and R. Heil. Distributed, physics-based control of swarms of vehicles. *Autonomous Robots*, 17(2-3):137–162, 2004.

[46] M. A. Demetriou. Guidance of mobile actuator-plus-sensor networks for improved control and estimation of distributed parameter systems. *IEEE Transactions on Automatic Control*, 55(7):1570–1584, 2010.

[47] Z. Song, Y. Chen, J. Liang, and D. Ucinski. Optimal mobile sensor motion planning under nonholonomic constraints for parameter estimation of distributed systems. In *Proceedings of IEEE International Conference on Intelligent Robots and Sys.*, pages 3163–3168, Edmonton, Alberta, Canada, August 2005.

[48] D. Ucinski and M. Demetriou. An approach to the optimal scanning measurement problem using optimum experimental design. In *Proceedings of the American Control Conference*, volume 2, pages 1616–1621, June 2004.

[49] R. Graham and J. Cortés. Cooperative adaptive sampling of random fields with partially known covariance. *International Journal of Robust and Nonlinear Control*, 5:504–534, 2012.

[50] R. Graham and J. Cortés. Adaptive information collection by robotic sensor networks. *IEEE Transactions on Automatic Control*, 57(6):1404–1419, June 2012.

[51] Y. Xu, J. Choi, and S. Oh. Mobile sensor network navigation using Gaussian processes with truncated observations. *IEEE Transactions on Robotics*, 27(6):1118–1131, December 2011.

[52] Y. Xu, J. Choi, S. Dass, and T. Maiti. Sequential Bayesian prediction and adaptive sampling algorithms for mobile sensor networks. *IEEE Transactions on Automatic Control*, 57(8):2078–2084, August 2012.

[53] C. Song, F. Feng, Y. Fan, and Y. Wang. Decentralized adaptive awareness coverage control for multi-agent networks. *Automatica*, 47(12):2749–2756, December 2011.

[54] Y. Wang and I. I. Hussein. Awareness coverage control over large-scale domains with intermittent communications. *IEEE Transactions on Automatic Control*, 55(8):1850–1859, August 2010.

[55] A. Kwok and S. Martínez. A distributed Deterministic Annealing algorithm for limited-range sensor coverage. *IEEE Transactions on Control Systems Technology*, August 2010. Electronically available. DOI: 10.1109/TCST.2010.2053036.

[56] M. Zhu and S. Martínez. Distributed coverage games for mobile visual sensor networks. *SIAM Journal on Control and Optimization*, 51:1–27, 2013.

[57] G. Mathew and I. Mezic. Spectral multiscale coverage: A uniform coverage algorithm for mobile sensor networks. In *Proceedings of the 48th IEEE Conference on Decision and Control*, pages 7872–7877, 2009.

[58] N. Sydney and D. A. Paley. Multi-vehicle control and optimization for spatiotemporal sampling. *Proceedings of IEEE Conference on Decision and Control*, pages 5607–5612, December 2011.

[59] A. Alvarez and B Mourre. Optimum sampling designs for a glider-mooring observing network. *Journal of Atmospheric and Ocean Technology*, 29, April 2012.

[60] D. A. Paley. *Cooperative Control of Collective Motion for Ocean Sampling with Autonomous Vehicles*. PhD thesis, Princeton University, September 2007.

[61] Sergey Frolov, Bartolame Garau, and James Bellingham. Can we do better than the grid survey: Optimal synoptic surveys in presence of variable uncertainty and decorrelation scales. *Journal of Geophysical Research: Oceans*, 119(8):5071–5090, 2014.

[62] D. Higdon, J. Swall, and J. Kern. Non-stationary spatial modeling. In *Bayesian Statistics 6*, pages 761–768. Oxford UP, 1999.

[63] J. Cortés, S. Martinez, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, 2004.

[64] N. E. Leonard and A. Olshevskyi. Nonuniform coverage control on the line. In *Proceedings of IEEE Conference on Decision and Control*, pages 3163–3168, Orlando, Florida, December 2011.

[65] P. D. Sampson and P. Guttorp. Nonparametric estimation of nonstationary spatial covariance structure. *Journal of the American Statistics Association*, 87:108–1195, March 1992.

[66] A. M. Schmidt and A. O'Hagan. Bayesian inference for non-stationary spatial covariance structure via spatial deformations. *Journal of the Royal Statistical Society: Series B*, 65:743–758, August 2003.

[67] P. R. Liebelt. *An Introduction to Optimal Estimation*. Addison-Wesley, 1976.

[68] A. Eliassen and J. S. Sawuer. Upper air network requirements for numerical weather prediction. *Technical Notes of th World Meteorological Organization*, 29, 1954.

[69] L. S. Gandin. *Gidrometerologicheskoe Izdatelstvo - Objective Analysis of Meteorological Fields*. Israel Program for Scientific Translations, 1963.

[70] A. Bennett. *Inverse Modeling of the Ocean and Atmosphere*. Cambridge University Press, 2005.

[71] F. P. Bretherton, R. E. Davis, and C. B. Faundry. A technique for objective analysis and design of oceanographic experiments applied to MODE-73. *Deep-Sea Research*, 23(7):559–582, 1976.

[72] N. E. Leonard, D. A. Paley, R. Sepulchre, D. M. Frantantoni, and R. E. Davis. Collective motion, sensor networks and ocean sampling. *Proceedings of the IEEE*, 95(1):48–74, 2007.

[73] H. K. Khalil. *Nonlinear Systems*. Prentice Hall, 2002.

[74] A. Verhoff. Two-dimensional potential flow solutions with separation. *Journal of Fluid Mechanics*, 657:238–264, 2010.

[75] J. Seddon and S. Newman. *Basic Helicopter Aerodynamics*. AIAA, 3rd edition, 2011.

[76] J. G. Leishman. *Principles of Helicopter Aerodynamics*. Cambridge University Press, New York, 2000.

[77] F. Hoblit. *Gust Loads on Aircraft: Concepts and Applications*. AIAA Education Series, 1998.

[78] H. Huang, G. M. Hoffmann, S. L. Waslander., and C. J. Tomlin. Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering. In *IEEE International Conference on Robotics and Automation, 2009. ICRA '09.*, pages 3277 –3282, May 2009.

[79] J. Kasdin and D. Paley. *Engineering Dynamics: A Comprehensive Introduction*. Princeton University Press, 2011.

[80] Heemin Shin, Dongil You, and D.H. Shim. An autonomous shipboard landing algorithm for unmanned helicopters. In *International Conference on Unmanned Aircraft Systems*, pages 769–778, May 2013.

[81] N. Sydney, B. Smyth, and D. A. Paley. Dynamic control of autonomous quadrotor flight in an estimated wind field. In *IEEE 52nd Annual Conference on Decision and Control*, pages 3609–3616, Dec 2013.

[82] F. White. *Viscous Fluid Flow*. McGraw-Hill Mechanical Engineering, 3 edition, January 2005.

[83] R. Jensen M. Albertson, Y. Dai and H. Rouse. Diffusion of submerged jets. In *Transactions*, volume 115, pages 639–697. American Society of Civil Engineers, 1950.

[84] W. Khan, R. Caverly, and M. Nahon. Propeller slipstream model for small unmanned aerial vehicles. In *AIAA Modelling and Simulation Conference*. AIAA, August 2013.

[85] N. A. Macmillan and C. Douglas Creelman. *Detection Theory: A User's Guide*. Psychology Press, 2nd edition, 2008.

[86] M. A. Richards. *Fundamentals of Radar Signal Processing*. McGraw-Hill, 2005.

[87] Cristóbal Carnero Li nán. cvBlob. http://cvblob.googlecode.com.

[88] N. Moin. *Fundamentals of Engineering Numerical Analysis*. Cambridge, 2nd edition, 2010.

[89] J.E. Lennard-Jones. On the determination of molecular fields. *Proceedings of the Royal Society of London*, 106(738):463–477, 1924.

[90] M. Quigley, K. Conley, B. Gerkey, J. F., T. Foote, J. Leibs, R. Wheeler, and A.Y. Ng. ROS: an open-source robot operating system. *ICRA Workshop on Open Source Software*, 3(3.2), 2009.

[91] R. Sepulchre, D. A. Paley, and N. E. Leonard. Stabilization of planar collective motion: All-to-all communication. *IEEE Transactions on Automatic Control*, 52:811–824, 2007.

[92] Y. Gabriely and E. Rimon. Spanning-tree based coverage of continuous areas by a mobile robot. *Annals of Mathematics and Artificial Intelligence*, 31, 2001.

[93] E. M. Arkin, S. P. Fekete, and J.S. Mitchell. Approximation algorithms for lawn mowing and milling. *Computational Geometry*, 17:25–50, October 2000.

[94] E. Justh and P. Krishnaprasad. Equilibria and steering laws for planar formations. *System Control Letters*, 52(1):25–38, 2004.

[95] N.J. Gordon, D. J. Salmond, and A.F. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *Radar and Signal Processing*, volume 140, pages 107–113, 1993.

[96] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.