ABSTRACT

Title of thesis: OPTIC FLOW BASED STATION-KEEPING AND WIND REJECTION FOR SMALL FLYING VEHICLES

Bryan Patrick, Masters of Science, 2010

Thesis directed by: Professor J. Sean Humbert
Department of Aerospace Engineering

Optic flow and Wide Field Integration (WFI) have shown potential for application to autonomous navigation of Unmanned Air Vehicles (UAVs). In this study the application of these same methods to other tasks, namely station-keeping and wind rejection, is examined. Theory surrounding optic flow, WFI and wind gust modeling is examined to provide a theoretical background. A controller based on a H_∞ bounded formulation of the well known Linear Quadratic Regulator in designed to both mitigate wind disturbances and station-keep. The performance of this controller is assessed via simulation to determine both performance and trade-offs in implementation such as the method for optic flow calculation. Furthermore, flight tests are performed to examine the real world effectiveness of the controller. Finally, conclusions about potential improvement to implementation are drawn.

OPTIC FLOW BASED STATION-KEEPING AND WIND REJECTION FOR SMALL FLYING VEHICLES

by

Bryan Patrick

Thesis 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
Masters of Science
2010

Advisory Committee:
Professor J. Sean Humbert, Chair/Advisor
Professor Robert M. Sanner
Professor Inderjit Chopra

Acknowledgments

It is impossible to fit all the people that deserve thanks in a scant page. I hope I can fit most of you.

My family whom have always been supportive deserve many thanks for their love and support. Mom, Dad, I know I have neither called nor came home as much as either of you would have liked but I am grateful to and love you both. Thanks to all my lab-mates in the Autonomous Vehicles Lab. Thanks to David McLaren for helping me make AVLSim, Greg Gremillion for quadrotor piloting, Kedar Dimble and Imraan Farque for letting me bounce ideas off you guys, Andrew Hyslop and Joe Conroy for the help and advice, Renee Campbell and Steve Gerardi for bearing interruptions caused by the flight tests, Jishnu Keshavan for finding the journal paper I needed and Badri Ranganathan for spotting my simple math mistakes. Also to thank are all those friends who have helped in ways that they may not even have realized. Chris Wilson, I remember all those late nights watching you give 110% that finally lead to you getting your degree. Thanks for the example you set. Of course, Dr. Humbert thank you for guiding me through graduate school these past couple years and most of all for convincing me that I could get into the program despite my mediocre GPA. Finally and most importantly, thanks be to God, without whom all is meaningless.

Table of Contents

List of Figures

List of Abbreviations

A	system dynamics/state matrix
a_i	i-th Fourier cosine harmonic coeffient
B	system input/control matix
b_i	i-th Fourier sine harmonic coeffient
β	body fixed elevation angle, positive z-axis(down) is zero
C	system output/observation matrix
$c_u,\ c_v,\ c_r$	proportionality constants for the optic flow velocity measurements
D	system disturbance input matrix
$d(\gamma)$	distance function for vehicle surroundings
$\delta_y aw$	vehicle yaw input
η	white noise input to shaping filter
$\mathbb{F}$	Fourier basis of sines and cosines
$\mathbf{g}$	gust input vector
g	gravity
g_i	gust velocity acting on state i (i = u,v,r)
g_{20}	gust velocity at 20ft altitude
γ	body fixed azimuth angle, positive x-axis (forward) is zero
h	altitude
I	pixel intensity
J	LQR cost function
K	controller gain matrix
K_i	gain for state i
K_p	proportional term gain matirx in MIMO PI controller
k_i	integral term gain in MIMO PI controller
L_2	2nd Lebesgue space, or space of square intergrable functions
L	Ricatti equation iteration matrix
L_T, L_W	characteristic lengths used in the four point model
l	PSD turbulence scale
μ	H_∞ bound inposed in LQR design
Ω	frequency
P	Ricatti equation solution matrix
Φ	basis function set used in WFI
ϕ_k	k-th WFI weighting function
ϕ_{ref}	lateral vehicle reference angle input
ψ	vehicle yaw Euler angle
Q	LQR state weighting matrix
$\dot{Q},\ \dot{Q}_\gamma,\ \dot{Q}_\beta$	optic flow (in the azimuth (γ) or elevation (β) directions
R	LQR input weighting matrix
$R_{BF},\ R_{FB}$	Rotation matrices between the fixed (inertial) and body frames
r	vehicle body yaw rate
ρ	LQR control authority cost constant
S	Von Karman PSD
s	Laplace variable
σ_w	RMS velocity of the wind

t	time
τ	autocorrelation time of Guass-Markov process
θ_{ref}	longitudinal vehicle reference angle input
$\mathbf{u}$	control input vector
u	longitudinal body velocity
$\mathbf{v}$	vector of vehicle velocites u, v and r
v	lateral body velocity
$\mathbf{x}$	vehicle state vector
x	inertial displacement of the vehicle in the x direction
$\mathbf{y}$	vehicle measurement vector
y	inertial displacement of the vehicle in the y direction
z	integral feedback term in PI controller
ζ	input scaling constant for Guass-Markov process
CPDGPS	Carrier Phase Differential Global Positioning System
FoE	Focus of Expansion
GPS	Global Positioning System
IMU	Inetial Measurement Unit
LQR	Linear Quadradic Regulator
MAV	Micro Air Vehicle
MIMO	Multi-Input Multi-Output
PI	Proportional-plus-Integral
PID	Proportional-Integral-Derivative
PSD	Power Spectrum Density
RMS	Root-Mean-Square
UAV	Unmanned Air Vehicle
WFI	Wide Field Intergration

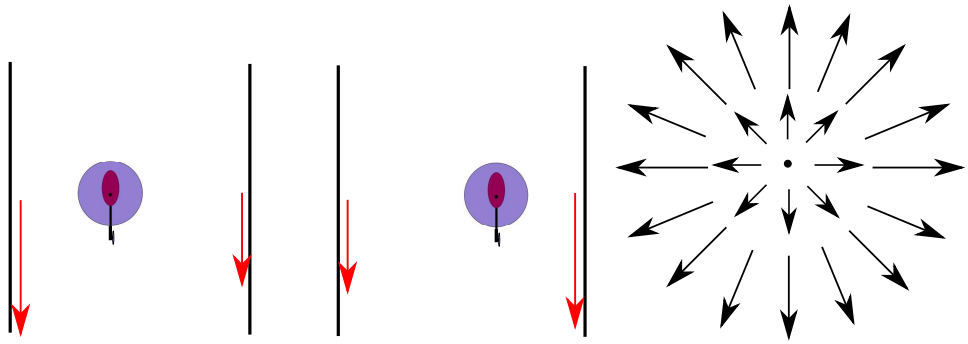
Chapter 1

Introduction

In recent years military operations have seen two new developments. First, is the widespread use of unmanned aerial vehicles (UAVs) for surveillance and in some cases even as strike aircraft. Secondly, in recent conflicts urban warfare has comprised a major portion of military operations. When considered together these two developments make the desire for UAVs capable of urban navigation seem obvious. However, the urban flight arena presents a number of challenges for autonomous operation of UAVs. The most obvious being the close proximity of obstacles many of which, such as power lines, are small and hard to detect. An additional effect of the close proximity of obstacles, is the requirement for high accuracy proximity measurements. GPS is not sufficient under these conditions due to the lack of knowledge of obstacle positions, accuracy issues and update rates. Furthermore, wind becomes a bigger issue due to airflows being forced between buildings and restriction on vehicle size imposed by urban clutter.

1.1 Goals and Motivation

This study was embarked upon with three goals in mind. First, was to develop a control strategy that could station-keep, or hold in place, a small flying vehicle without relying only on off-board sensing equipment. Secondly, was to ensure that



(a) (b)

Figure 1.1: (a) The "balance strategy" [36] tries to center the vehicle by equalizing the optic flow measurements on the right and left. When offset the closer obstacle, here a wall, will produce more optic flow. (b) The Focus of Expansion(FoE) occurs in the direction of travel and can be identified by the divergent node shape

such a controller would be robust under windy conditions that might be encountered in an urban environment. Lastly, accomplish the previous two tasks by sensing apparent motion of stationary objects induced by vehicle motion, called optic flow. Optic flow sensing has shown promise in urban navigation for obstacle detection and avoidance. If this same paradigm could be adapted to provide both station-keeping and wind rejection then the need for different sensors and thus increased payload and power requirements could be eliminated.

1.2 Urban Navigation Using Optic Flow

Optic flow can be defined as the motion of static objects perceived by a moving observer and encodes information concerning both the velocity of the observer as

well as the proximity the objects. In nature, the visual systems of insects rely heavily on optic flow [3, 9]. Whether using the compound insect eye or a man-made camera optic flow is calculated by monitoring changes in luminosity patterns with time. The result of optic flow calculation can be seen as a velocity field that describes the relative motion of surrounding obstacles.

Different studies conducted using optic flow employ algorithms to extract information from optic flow and preform feedback. One of the most common strategies is called "balance strategy" or "centering strategy" [36]. This method attempts to equalize distance from obstacles by equalizing the optic flow measurements on the left and right of the vehicle (see Figure 1.1a). This strategy has been investigated in a number of studies including [18, 36, 44, 48, 7, 39]. One disadvantage of the "balance strategy" is corruption of measurements due to vechicle rotation. Various methods have been employed to counter this including breaking robot motion into translation and rotation phases [36, 39], counter-rotation of the camera [7] and using measurement of the rotation to subtract this component of optic flow [48]. Another method uses a feature of the optic flow field known as the Focus of Expansion(FoE) to determine the direction of travel of the vehicle. The FoE is simply the point of divergence in the optic flow velocity field the corresponds to the pointing direction of the velocity vector as shown in Figure 1.1b. In these strategies the FoE is used to determine the direction of travel and distinguish when the motion of the vehicle is likely to cause a collision using the surrounding optic flow. This method in various incarnations was used in the studies [13, 36, 42, 46]. Both of these methodologies have demonstrated the ability to use optic flow to navigate simple environments

such as straight and bent corridors.

More recent work by the Autonoumous Vehicles Lab at University of Maryland [25, 22, 5, 27] suggested a more formal approach to extraction of data from optic flow. Here a method, called Wide-Field Integration (WFI), involved integration over the optic flow field weighted by patterns designed to extract vehicle data. In addition to providing a more formal framework WFI also drew inspiration from the structure of the insect brain. An alternative interpretation of the WFI process, is a decomposition of the optic flow field into its components along the directions defined by the weighting functions [27]. This decomposition along specific directions is analogous to specialized neurons in insects known as tangential cells. These tangential cells are sensitive to specific patterns in optic flow and will exhibit stronger responses as the observed optic flow becomes closer to the preferred pattern [10, 32, 28, 3]. Work involving simulation and experiments has demonstrated the ability of WFI to perform the tasks previously accomplished using the "balance strategy" or FoE methods [21, 20, 19, 5, 24, 26]. Furthermore, simulation and experimental work by Hyslop [27, 28, 26] suggested the viability of WFI methods for more complex environments including simulated urban scenarios. Furthermore, Hyslop proposes a framework for selection of linearly optimal vehicle state-extraction patterns along with a method of reducing these patterns to a set that maps directly from output to input thus, reducing the number of integrations required for feedback to the number of vehicle inputs.

In addition to the use of optic flow to detect and avoid obstacles, some studies have been performed on odometry, or measurement of distance traveled, using optic

flow. This is of particular interest since position, or distance, measurements are required for station-keeping. Experiments by Weber [48] involving a ground robot using the "balance strategy" in a straight corridor showed integrating optic flow to give distance traveled yielded a standard deviation that was approximately 3.6% of the mean measured distance. This suggests that integrated optic flow could give a somewhat precise measure of distance traversed. A similar study by McCarthy [35] showed for a 2.5 meter corridor showed between 0.03-0.13 meter standard deviation of approximately 1.2-5.2%. A preliminary investigation into underwater optic flow based station-keeping [37] where optic flow calculated position was compared to actual position for a series of discrete displacements of a target object showed a average error of 5.5%.

Another development in optic flow research that is discussed in [14, 13], is the development of dedicated sensors for optic flow acquisition. These sensors have a mass of about 5 grams and operate at speeds up to 1.4 KHz. Furthermore, optic flow can be detected at up to rates of 20 rad/s and in low contrast situations. Additionally, another sensor by Harrison [16], consisting of radially distributed optic flow sensors, was able to detect imminent collisions approximately 0.1-0.4 seconds before contact using only 140 μW of power. Other efforts towards optic flow based sensors are covered in [29, 43].

1.3 Station-Keeping in Robotics

While numerous methods can be imagined for implementation of station-keeping on robotic platforms, here only GPS and computer vision based methods will be discussed. Many studies have been performed for using control based on GPS positioning [40]. However, for urban operations accuracy is important. A comparison of GPS techniques in shows the only GPS technology with sub-meter accuracy is Carrier Phase Differential GPS (CPDGPS) which is accurate within 4-6 centimeters[40, 47]. However, CPDGPS does have a number of disadvantages including the need for a reference GPS station. The need for this station also limits the range of operation to about 10-15 kilometers (6.2-9.3 miles). Additionally, CPDGPS is somewhat expensive costing around $20,000-30,000 and requires complicated calibration procedures [40].

Waslander in his study [47] simulated CPDGPS on a small quadrotor with realistic error models along with a suite of other sensors including a full inertial measurement unit(IMU), magnetometer and sonar height senors. Additionally, wind disturbances were simulated. The resultant simulations had the vehicle remain within 0.4 meters under 2-7 m/s winds using a combination of an Extended Kalman Filter and PID controller. With a wind estimation model the error was reduced to 0.1 meters. However, although this study shows good results it employs computationally expensive methods such as the Extended Kalman Filter and a wind estimation model which may not be practical on a small on-board processing unit.

In addition to GPS based positioning, computer vision is another method

that is commonly used for underwater vehicles. However, these methods are often computationally expensive to implement. Thus with the limited payload of UAVs the feasibility of these methods is questionable. One method designed to eliminate estimate drift [34] required thousands of single board computers to perform the calculation to station-keep an underwater vehicle with a loop closure rate of 30Hz. This however is extreme, another study [30] using image registration techniques required only a two computers to archive similar closure rates at the cost of estimate drift.

1.4 Overview of this Study

Towards the purpose of meeting the goals of section 1.1 design of WFI sensing scheme and control strategy was pursued. WFI basis functions were chosen to extract vehicle states about the operating condition. Using these measurements, a H_∞ bounded Output Linear Quadratic Regulator (LQR) designed specifically for wind-like disturbances was implemented. Additionally, a nonlinear, error signal integration scheme was employed to account for errors otherwise introduced by the vehicle rotational kinematics. Simulation was used to examine controller performance with respect to different optic flow calculation algorithms, disturbances types and evaluate the effectiveness of the nonlinear integration scheme. Furthermore, flight testing of a vehicle was performed under both still-air and gust-like conditions to quantify real-life performance of the control scheme. Metrics such as position estimate error and integrator drift rate are presented to provide a measure

of the controller performance. Finally, results are discussed and conclusions drawn as to the performance of the controller and future work to be performed in this area.

Chapter 2

Optic Flow

In this Chapter the theory associated with optic flow will be examined. Also, the work of Humbert [25] and the framework for control feedback known as Wide-Field Integration (WFI) will be explained. Additionally, explanation of the WFI weighting patterns selected this study will be discussed. Furthermore, optic flow calculation algorithms are discussed. Finally, limitations of optic flow are addressed.

2.1 Background

Optic flow can intuitively be defined as the apparent motion of surroundings as seen by a moving object. More technically, it is the apparent motion of objects in the inertial frame as observed in a moving frame. Mathematically, optic flow can be seen as the projection of the apparent motion of surroundings onto a body-fixed viewing surface (usually a sphere) as shown in Figure 2.1. The general form of optical flow on the sphere was derived in [31]. However, this study is only concerned with optic flow in the yaw, or azimuth, plane a simpler representation can be used. Such a representation was given by [25].

$$\dot{Q} = -\dot{\psi} + \frac{1}{d(\gamma)}(u \sin\gamma - v \cos\gamma) \tag{2.1}$$

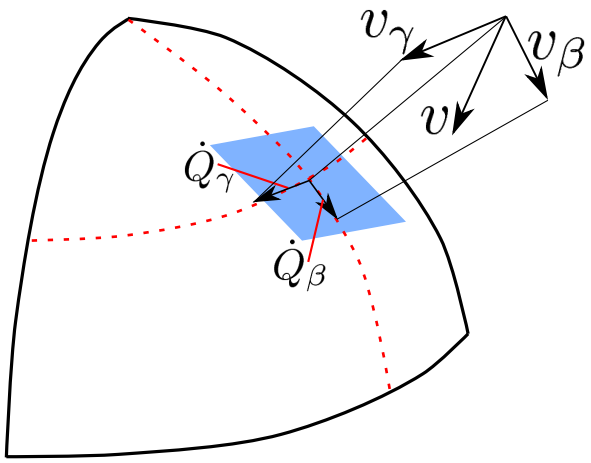


Figure 2.1: Optic flow projects object motion onto the tangent space of the unit sphere

where $\dot{\psi}$ is the vehicle yaw rate, $d(\gamma)$ is the distance to surrounding objects as a function of γ the body frame azimuth angle, and finally u and v are the body frame longitudinal and lateral velocities. Examination, of (2.1) reveals a couple of properties of optic flow. First, that optic flow encodes velocities in a manner that is inversely proportional to distance. This can be seen geometrically in Figure 2.2 where the angle swept out by the farther away green object is smaller that the closer red object. Secondly, it can be noted that in general the different motions $\dot{\psi}$, u and v will create different patterns of optic flow. In the case of rotation ($\dot{\psi}$) the pattern is clearly a constant function. While, in the cases of the translations (u and v) the patterns are much less well defined due to the division by $d(\gamma)$. However, (2.1) does suggest patterns resembling sine and cosine waves. Figure 2.3 shows this more clearly; optic flow will be zero along the axis of translation and maximum on the axis perpendicular to the translation. If this is then parameterized in terms of the body frame azimuth angle, γ, defined earlier in (2.1) then the patterns recovered

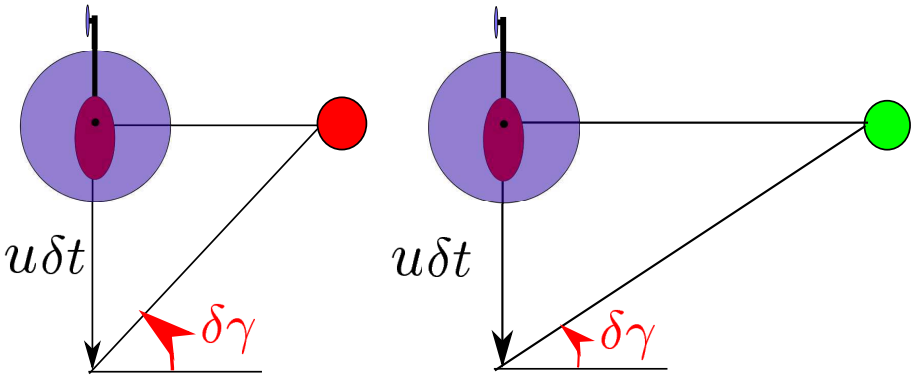


Figure 2.2: Optic flow due to translation is inversely proportional to distance. Note $\delta\gamma$ is larger for the closer object

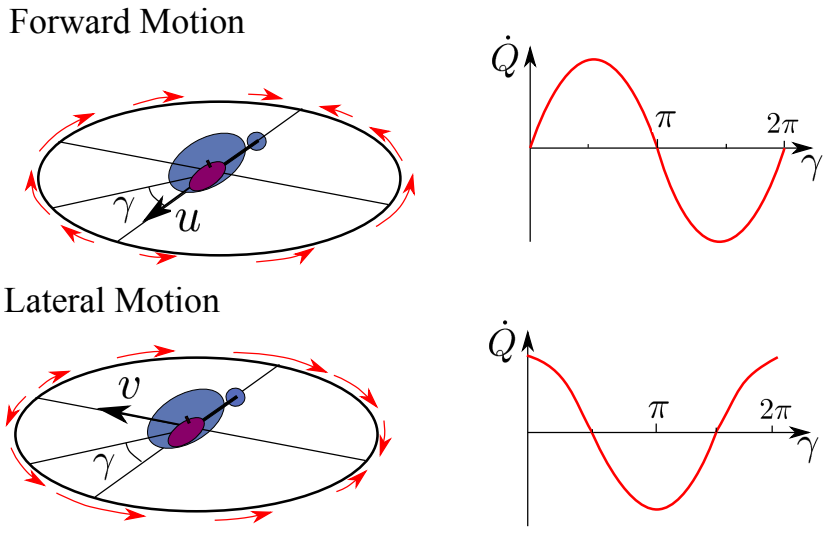


Figure 2.3: Forward and Lateral motion result in Sine and Cosine like shapes

bear resemblance to sine and cosine functions with period 2π for the body velocities u and v respectively.

2.2 Wide-Field Integration

WFI in general involves a number of weighted integrations over a large collection of distributed sensors. In the case of camera based optic flow these sensors are single image pixels. These integrations form a decomposition of the optic flow signal into components along a basis of a subspace of L_2, the space of square integrable functions or second Lebesgue space. More mathematically, let $\Phi = \{\phi_k\},\ k = 1, 2...n$ be N elements of the function space basis to be projected onto. Also, assume $\phi_k \in L_2$ and $\dot{Q} \in L_2$. Then the projection of $\dot{Q}$ onto any of these bases can be calculated as a dot product on L_2 [22].

$$\mathbf{proj}_{\phi_k}\dot{Q} = \frac{1}{\pi}\int_0^{2\pi} \dot{Q}\phi_k d\gamma \tag{2.2}$$

In the discrete case with m samples this becomes:

$$\mathbf{proj}_{\phi_k}\dot{Q} = \frac{\delta\gamma}{\pi}\sum_{i=0}^{m} \dot{Q}(\gamma_i)\phi_k(\gamma_i) \tag{2.3}$$

Thus, WFI depends largely on picking a set of basis functions Φ that will extract the states of the system in a meaningful manner. From the observations made about (2.1) above it seems likely that a constant function along with the 2π period sine and cosine will extract the state velocities. This suggests that a subset the Fourier basis of sines and cosines defined as $\mathbb{F} = \{1/2\} \cup \{\cos n\pi\} \cup \{\sin n\pi\}$ where $n = 1, 2...\infty$ would be a good choice of Φ. This choice of Φ has been used in many studies such

as [22, 24, 5]. Here the zero-th and first order harmonics denoted, a_0, a_1 and b_1 where a and b denote sine and cosine respectively, are used.

2.3 Calculating Optic Flow

Numerous methods for calculating optic flow exist. During the course of this study two different methods of calculating optic flow from images were considered. The first is the so called Gradient Method which measures optic flow in a single direction, In the case of the work done here this was in a azimuthal ring around the vehicle. The Gradient Method relies on the assumption that pixel intensity is conserved between image frames. This results in the following constraint equation [45]

$$\frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial t} = 0 \tag{2.4}$$

where I is pixel intensity, x is the pixel position and t is time. Solving (2.4) for the pixel shift dx gives the following equation

$$\frac{\frac{\partial I}{\partial t}}{\frac{\partial I}{\partial x}} = dx. \tag{2.5}$$

Once dx is solved for using (2.5) the camera field of view and direction can be used to formulate the optic flow measurements. The second method was the Lucas-Kanade Method which is a measures optic flow in both the azimuth and elevation directions. This method generally is less noisy than the above Gradient Method [41]. In addition it was found by Barron in his comparison of 2-D optic flow methods, [2], that Lucas-Kanade was one of the most reliable methods tested. Additionally,

a comparative study by Weber [48] showed Lucas-Kanade produced the best results for both corridor centering and odometry. The calculation of optic flow using Lucas-Kanade is complex and will not be given here but is available in [33].

2.4 Limitations of Optic Flow

Optic flow theoretically provides a number useful properties for use in control feedback. However, in practice a couple of issues can arise. The first is the need for image contrast. If the surroundings imaged have poor contrast, such as the case of a room with all white walls, the calculated optic flow will be likewise poor due to the uniformity of the image pixels. Second, is the need for relatively constant brightness between frames. As previously, stated the assumption of intensity conservation is used in optic flow calculation thus this is a direct violation of this assumption. Finally, the image capture rate must be high enough for pixel shifts to be "small" between successive frames.

Chapter 3

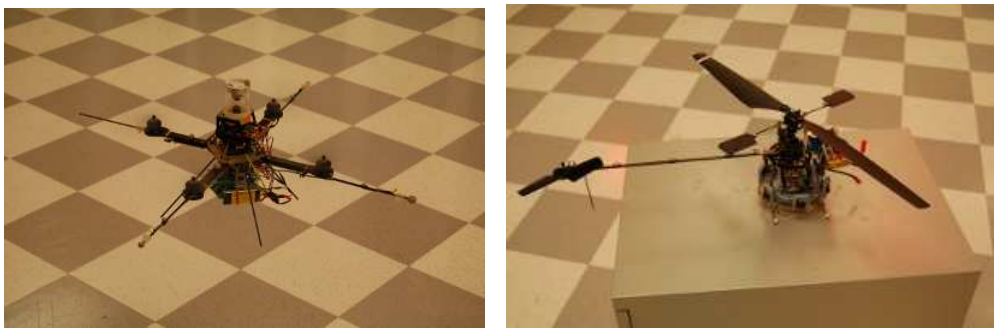
Vehicle Dynamics

Two different flying vehicles where utilized during this study. The first was a E-sky Honeybee hobby helicopter and the second was the X-UFO Quadrotor MAV made by Ascending Technologies GmbH (see Figure 3.1). The linearized dynamics about hover for each vehicle is presented in this section. Additionally, the effects of wind on vehicle dynamics are considered. Finally, the derivation of the sensor output equation is presented.

3.1 Linear Dynamics about Hover

Dynamical models of both vehicles were obtained using linear system identification which are covered in [6] and [15]. Additionally, the full dynamics of each system was reduced via inner loop control of the non-planar system states using onboard avionics along with a Vicon^{TM} visual positioning system [5, 23]. This allowed examination of a smaller set of states and thus a simpler design problem. The reduced dynamics as presented in the papers above are as follows. The helicopter dynamics are

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} X_u & 0 & 0 \\ 0 & Y_v & 0 \\ 0 & 0 & N_r \end{bmatrix} \begin{bmatrix} u \\ v \\ r \end{bmatrix} + \begin{bmatrix} 0 & -g & 0 \\ g & 0 & 0 \\ 0 & 0 & N_{\delta_{yaw}} \end{bmatrix} \begin{bmatrix} \phi_{ref} \\ \theta_{ref} \\ \delta_{yaw} \end{bmatrix}, \tag{3.1}$$



(a) X-UFO Quadrotor MAV (b) E-Sky®Honeybee

Figure 3.1: Images of the two vehicles studied

Table 3.1: Stability Derivative Values

Derivative	Helicopter	Quadrotor
X_u	-.5219	-.2203
Y_v	-.4799	-.2291
N_r	-.8937	0
$N_{\delta_{yaw}}/\Psi_{\delta_{yaw}}$	37.09	4.79

and the quadrotor dynamics are

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} X_u & 0 & 0 \\ 0 & Y_v & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ \psi \end{bmatrix} + \begin{bmatrix} 0 & -g & 0 \\ g & 0 & 0 \\ 0 & 0 & \Psi_{\delta_{yaw}} \end{bmatrix} \begin{bmatrix} \phi_{ref} \\ \theta_{ref} \\ \delta_{yaw} \end{bmatrix}. \tag{3.2}$$

where the states u, v, ψ and r are perturbation states about hover for the body longitudinal and lateral velocities, yaw angle and yaw rate respectively. In addition, the inputs ϕ_{ref}, θ_{ref} and δ_{yaw} are the longitudinal and lateral reference angles for the inner loop control and the vehicle yaw input respectively.

The stability relevant derivatives obtained from the system identifications [15, 6] are shown in Table 3.1. Examination of the two models in equations (3.1) and (3.2) shows the only difference in structure between the two models is the rotational dynamics. This lack of a yaw rate term results form two issues. First, the high resposiveness of the vechicle allows for rotational dynamics to evovle quickly. Second, the size of the flight arena used in the system identification was too small for the more aggresive manuvers require to extract these terms [15].

Since the objective of this study is station-keeping the kinematic equations also need to accounted for. In the planar case this becomes an elementary rotation about the yaw-axis by the yaw angle ψ. Thus defining the state vector

$$\mathbf{x} = [x, y, \psi, u, v, r]^T \tag{3.3}$$

and using the standard form for a linear system

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} \tag{3.4}$$

the A and B matrices can augmented to include the kinematics for the helicopter and quadrotor respectively.

$$A = \begin{bmatrix} 0 & 0 & 0 & \cos\psi & -\sin\psi & 0 \\ 0 & 0 & 0 & \sin\psi & \cos\psi & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & X_u & 0 & 0 \\ 0 & 0 & 0 & 0 & Y_v & 0 \\ 0 & 0 & 0 & 0 & 0 & N_r \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -g & 0 \\ g & 0 & 0 \\ 0 & 0 & N_{\delta_{yaw}} \end{bmatrix} \tag{3.5}$$

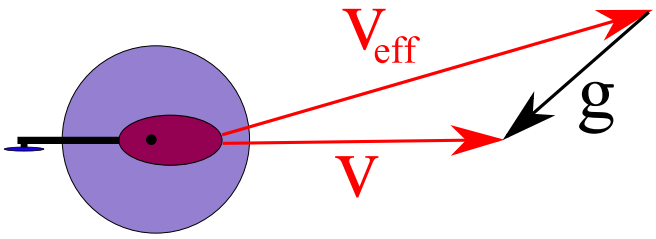


Figure 3.2: Wind acts as a increment to the velocity

$$A = \begin{bmatrix} 0 & 0 & 0 & \cos\psi & -\sin\psi & 0 \\ 0 & 0 & 0 & \sin\psi & \cos\psi & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & X_u & 0 & 0 \\ 0 & 0 & 0 & 0 & Y_v & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \Psi_{\delta_{yaw}} \\ 0 & -g & 0 \\ g & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.6}$$

3.2 Wind Effects

Additionally, it was necessary to include wind disturbances in the dynamical modeling of the vehicles. Dynamically, wind acts as a perturbation of velocity [38]. This can be seen intuitively as in the absence of wind the airflow acts move equal and opposite to the vehicle velocity. Thus as shown in Figure 3.2 the effective velocity can be given by

$$\mathbf{v}_{eff} = \mathbf{v} - \mathbf{g} \tag{3.7}$$

where $\mathbf{v} = [u, v, r]^T$ is the vector of vehicle velocities and $\mathbf{g}$ is the wind, or gust, velocity. Applying this to the linearized equations of motion as in the following example using the longitudinal velocity dynamics from above [11, 38] the modified

equations can be obtained.

$$\dot{u} = X_u(u - g_u) - g\delta_{long} \tag{3.8}$$

This can be shown to be equivalent to extracting the aerodynamic stability derivatives from the state dynamics, or A, matrix and negating them to form a disturbance matrix, here denoted D. The modified state equation is given below.

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} + D\mathbf{g} \tag{3.9}$$

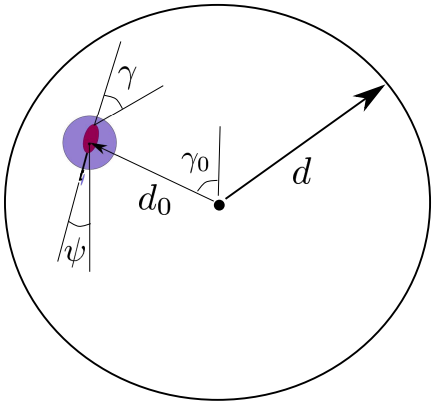
and the D matrix can be given by

$$D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -X_u & 0 & 0 \\ 0 & -Y_v & 0 \\ 0 & 0 & -N_r \end{bmatrix} \tag{3.10}$$

Note here that for the quadrotor the derivative $N_r = 0$ thus this model is insufficient to simulate rotational disturbances on the vehicle. For this reason the helicopter was used for simulation in lieu of the quadrotor vehicle.

3.3 Output Equation

The final part of the dynamics is the output equation. As was noted in Chapter 2 optic flow has a direct dependence on the shape of the surrounding environment, represented as $d(\gamma)$. For the purpose of this study the environment shape was taken



(a)

Figure 3.3: A circular environment can be defined by the radius and the polar position of the vehicle

to be a circle. Earlier, work [5, 24, 25] has used a hallway like shape of two parallel walls. However these were focused on forward-flight rather than hover for which a more room-like shape is more practical. For the case of the circle the environment distance function can be given by

$$d(\gamma) = -d_0 \sin(\gamma + \gamma_0) + \sqrt{d^2 - d_0^2 \cos^2(\gamma + \gamma_0)} \tag{3.11}$$

$$d_0 = \sqrt{x^2 + y^2}, \quad \gamma_0 = \tan^{-1} \frac{x}{y}$$

where x,y and ψ are the Cartesian position and orientation respectively and d is the radius. Using this distance function it is possible to numerically calculate the Fourier (WFI) coefficients that will be used for feedback control. Analytical solutions were not able to be obtained due to the complexity of the distance function. However, numerically calculating the partial derivatives of the WFI outputs can be used to form the output equation. First, the optic flow about some reference condition

denoted $\mathbf{x}^*$, in this case for hover ($\mathbf{x}^* = 0$), is calculated by substituting into (2.1) and the environment distance equation $d(\gamma)$.

$$\begin{aligned} \dot{Q}(\mathbf{x}^*, \gamma) &= -\dot{\psi}^* + \tfrac{1}{d(\gamma, \mathbf{x}^*)}(u^* \sin\gamma - v^* \cos\gamma) \\ \mathbf{x}^* &= [x^*, y^*, \psi^*, u^*, v^*, \dot{\psi}^*]^T \end{aligned} \tag{3.12}$$

Next, each entry in x^* is perturbed individually by a small amount, here denoted δx_i. Then, the difference caused by each perturbation is calculated as follows

$$\delta\dot{Q}_i = \frac{\dot{Q}(x^* + \delta x_i, \gamma) - \dot{Q}(x^*, \gamma)}{\delta x_i}, \tag{3.13}$$

where i is used to denote that the i-th state was perturbed. From here the change to the ouput, here the Fourier coefficients a_0, a_1 and b_1, can be calculated.

$$\begin{aligned} \tfrac{\partial a_0}{\partial x_i} &= \tfrac{1}{2\pi}\textstyle\int_0^{2\pi} \delta\dot{Q}_i d\gamma \\ \tfrac{\partial a_1}{\partial x_i} &= \tfrac{1}{\pi}\textstyle\int_0^{2\pi} \cos(j\gamma)\delta\dot{Q}_i d\gamma \\ \tfrac{\partial b_1}{\partial x_i} &= \tfrac{1}{\pi}\textstyle\int_0^{2\pi} \sin(j\gamma)\delta\dot{Q}_i d\gamma \end{aligned} \tag{3.14}$$

where j denotes the frequency of the Fourier coefficient. From these partial derivatives the observation, or C, matrix can be constructed as follows

$$C = \begin{bmatrix} \frac{\partial b_1}{\partial x_1} & \frac{\partial b_1}{\partial x_2} & \dots & \frac{\partial b_1}{\partial x_n} \\ \frac{\partial a_1}{\partial x_1} & \frac{\partial b_1}{\partial x_2} & \dots & \frac{\partial a_1}{\partial x_n} \\ \frac{\partial a_0}{\partial x_1} & \frac{\partial b_1}{\partial x_2} & \dots & \frac{\partial a_0}{\partial x_n} \end{bmatrix} \tag{3.15}$$

For this environment distance function $d(\gamma)$, given in 3.11, the C matrix takes the form

$$C = \begin{bmatrix} 0 & 0 & 0 & c_u & 0 & 0 \\ 0 & 0 & 0 & 0 & c_v & 0 \\ 0 & 0 & 0 & 0 & 0 & c_r \end{bmatrix} \tag{3.16}$$

where c_u, c_v and c_r are constants of proportionality and the output equation is given as shown.

$$\mathbf{y} = C\mathbf{x}, \quad \mathbf{y} = [b_1, a_1, a_0]^T \tag{3.17}$$

Chapter 4

Control Design

In developing control methodologies for station-keeping with wind disturbances, a number of methods were tried. However, the control strategy that was implemented was a modified version of a multi-variable "proportional plus integral" (PI) controller. This controller was chosen due to the requirement of position feedback in station-keeping. Proportional feedback of velocity would only result in a constant velocity in presence of a constant wind. However, with a integral position feedback term a constant position is maintained. This intuitively makes sense as the position feedback provides a stiffness while the velocity feedback only damping.

This compensation scheme was designed similar to the standard multi-variable PI controller except a non-linear correction was employed to correct for the error introduced by rotation of the vehicle. This error arises because any non-zero vehicle orientation will cause the inertial and body frames to not be aligned and thus optic flow, which is measured in the moving body frame, when integrated will produce an error. As such without correction, the position estimates would be inaccurate in presence of any rotation due to this integration in a moving frame. The following is a brief description of the corrected state integration methodology.

4.1 Integration Scheme

The state integration strategy in general relies on three assumptions in order to work

Assumptions

1. A measurement of the body yaw angle (ψ) is available, possibly from integration of a measurement of $\dot{\psi}$
2. Measurements proportional to the body velocities (u, v) are available
3. The body velocity measurements are proportional to each other by a known constant (ie. c_v/c_u from (3.16) is known).
4. The vehicle can be assumed to act in the azimuthal plane, possibly due to inner loop control.

The first assumption is necessary to correct for the rotation of the body frame. The second is required for the standard PI feedback. The third assumption is required to ensure the velocities measurements have the same scale and thus avoid errors. As can be seen form the output equation presented in Chapter 3 in equation (3.16), these requirements are met since uncoupled measurements of these quantities exist.

With these assumptions the following procedure can be implemented to estimate the position and attitude form the velocity and rate measurements.

Non-Linear Integration Scheme

1. Scale the velocity measurements to the same size using the constant from Assumption 3

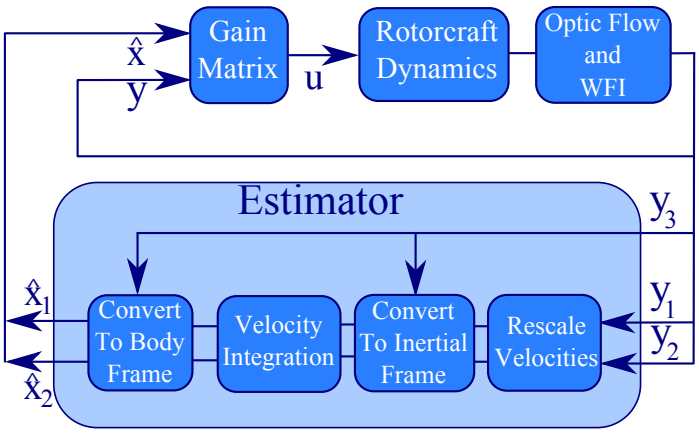


Figure 4.1: Signal Integration: signal integration is carried out by first using the yaw angle measurement (here y_3) to construct matrices to convert between the inertial and body frames. The velocity measurements y_1 and y_2 are then scaled (if needed) and integrated in the inertial frame to get a position which is converted to the body frame for feedback.

2. Convert the body velocity measurements into inertial velocity estimates via rotation matrix constructed from the yaw angle from Assumption 1 (denoted as R_{FB})

3. Integrate the resulting inertial velocity to update the inertial position estimates

4. Convert the inertial position into the body frame using the inverse rotation matrix (denoted R_{BF}) for feedback

The above process, is merely a conversion into the internal frame before the integration so that it is done in a non-moving frame. The conversion back to the body frame at the end is carried out since the input will be entering into the equations for the body velocities. A diagram of this process is provided in Figure 4.1. As a

result of this process the system observation, or C, matrix is effectively augmented as follows

$$C_{eff} = \begin{bmatrix} c_v & 0 & 0 & 0 & 0 & 0 \\ 0 & c_v & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_v & 0 & 0 \\ 0 & 0 & 0 & 0 & c_v & 0 \\ 0 & 0 & 0 & 0 & 0 & c_r \end{bmatrix} \tag{4.1}$$

where c_v and c_r are the nominal velocity measurement proportionality constant and yaw rate proportionality constant. Note that the assumed knowledge (see Assumption 1) of ψ is reflected in the matrix. This C matrix is used to design the feedback gain matrix.

4.2 Gain Matrix Selection

The final step in design of the control strategy was choice of feedback gains. The gain matrix used for feedback of the state estimates and WFI measurements was designed using a modified version of output Linear Quadratic Regulator (LQR) presented in [12] which imposes an H_∞ robustness criterion. The robustness criterion can be seen in the form of the cost function for this formulation. For a system of the following form,

$$\dot{x} = Ax + Bu + Dd, \quad y = Cx \ , \tag{4.2}$$

which is the same form previously given for wind disturbed systems in Chapter 3 (see (3.9)), the cost function is given by

$$J = \frac{\int_0^\infty \|z(t)\|^2 dt}{\int_0^\infty \|d(t)\|^2 dt} = \frac{\int_0^\infty (x^T Q x + u^T R u) dt}{\int_0^\infty (d^T d) dt} \leq \mu. \tag{4.3}$$

here d is the wind disturbance and μ is a positive constant. Looking at (4.3) it can be seen that μ can be taken to be the L_2 gain between the LQR cost and the disturbance. Recalling that the induced norm of a process on L_2 signals is the ∞-norm [49] this can be seen as a bound on the supremum of the transfer function between the LQR cost and the disturbance. Thus, this formulation can be looked at as a H_∞ bounded LQR. The solution to this problem is given by an iteration on the following Algebraic Ricatti Equation(ARE) and matrix equations.

$$P_n A + A^T P_n + Q + \frac{1}{\mu^2} P_n D D^T P n - P n B R^{-1} B^T P_n + L_n^T R^{-1} L_n = 0 \tag{4.4}$$

$$K_{n+1} = R^{-1}(B^T P_n + L_n) C^T (C C^T)^{-1} \tag{4.5}$$

$$L_{n+1} = R K_{n+1} C - B^T P_n, \quad L_0 = 0 \tag{4.6}$$

The terminal condition of the iteration is convergence of the L matrix. It is important to note here that (4.5) is calculated using the right inverse of C which will reconstruct the states from the outputs provided that (A,C) is observable. If the C matrix is invertible as in (4.1) then this iteration becomes deterministic with $L = 0$ being the convergent value of L and only (4.4) and (4.5) have to be solved.

For design of the R and Q weighting matrices Bryson's Method [4] was used. Thus R and Q can be given by

$$R = \mathbf{diag}\{\frac{\rho}{u_{i,max}^2}\} \tag{4.7}$$

$$Q = \mathbf{diag}\{\frac{1}{x_{i,max}^2}\} \tag{4.8}$$

where ρ is a constant specifying the relative cost of input with respect to the state error, $u_{i,max}$ are the maximum input values and $x_{i,max}$ are the maximum desired excursions of the states. For the design of feedback gain matrices for the two vehicles studied here (see Chapter 3) the gain matrices obtained from this process had the following forms. For the helicopter the matrix was structured as

$$K = \begin{bmatrix} 0 & K_y & 0 & 0 & K_v & 0 \\ K_x & 0 & 0 & K_u & 0 & 0 \\ 0 & 0 & K_\psi & 0 & 0 & K_r \end{bmatrix} \tag{4.9}$$

and for the quadrotor the matrix took the form

$$K = \begin{bmatrix} 0 & K_y & 0 & 0 & K_v & 0 \\ K_x & 0 & 0 & K_u & 0 & 0 \\ 0 & 0 & K_\psi & 0 & 0 & 0 \end{bmatrix} \tag{4.10}$$

which reflects the difference in dynamical structure discussed in Chapter 3. The nominal gain values are given in Table 4.1.

Table 4.1: Nominal Gain Values

(a) Helicopter

Gain	Value
K_x	-5.4017
K_y	5.2639
K_ψ	3.3425
K_u	-3.6954
K_v	3.6143
K_r	0.7517

(b) Quadrotor

Gain	Value
K_x	-14.6473
K_y	16.5788
K_ψ	3.3333
K_u	-7.8352
K_v	8.4936
K_r	0.0000

4.3 Comparison to Multi-variable PI Control

As discussed at the begin of this section, this methodology does bear resemblance to a MIMO PI regulator. The general form of this regulator can be expressed as [1]:

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} Ax + Bu \\ Cx \end{bmatrix} \tag{4.11}$$

$$u = -K_p x - k_i z \tag{4.12}$$

where K_p is the standard gain matrix and k_i is a scalar. Looking at this formulation there are two differences between the LQR methodology implemented here and PI control:

1. the LQR formulation has a non-linear $\dot{z}$ equation for the position terms $\dot{z} = R_{FB}(\psi)Cx$

2. the control equation is non-linear for the positions in the LQR case and uses a matrix in place of k_i ($u = -K_p x - K_i R_{BF}(\psi) z$)

Note here that the R_{BF} and R_{FB} matrices only act on the position states. Looking at items 1 and 2 the LQR method can be seen to be a adaptation on PI control for the system in question. Largely, the differences come from taking into account the difference in the inertial and body frame coordinates. Additionally, the LQR formulation allows for coupling in the integral feedback term which, while not used for this system, allows for better compensation if states are somehow coupled.

Chapter 5

Simulation

As a preliminary examination of controller performance simulation was used. Additionally, simulation allowed investigation of variables such as effect of different wind disturbances and optic flow algorithms that were not easily replicated in experiment. For simulation a custom software package AVLsim was used. In order to extend the capabilities of AVLsim to meet the needs of this study modifications such as implementing wind disturbance models were made. This Chapter discusses the AVLsim capabilities and the specific implementation used in this study. Additionally, formulation of wind models are presented. Furthermore, the structure of simulation trials are discussed. Finally, results are presented and conclusions drawn.

5.1 Simulation Description

AVLsim allows simulation of a single vehicle in a 3-D graphical environment. Additionally, simulation of sensors onboard the vehicle such as including cameras and distance sensors in included. Optic flow simulation is carried out by rendering one or more of these virtual camera images to computer memory and then calculating optic flow using these images. This optic flow is processed with WFI algorithm from Chapter 2 and then the extracted coefficients are used for feedback. For the simulations run in this study, a ring of twelve cameras in the body azimuth plane was

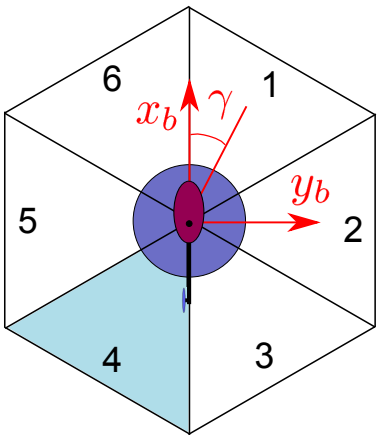


Figure 5.1: Vehicle Camera Configuration: Cameras are configured in the azimuth plane. Here x_b and y_b are the body forward and right axes.

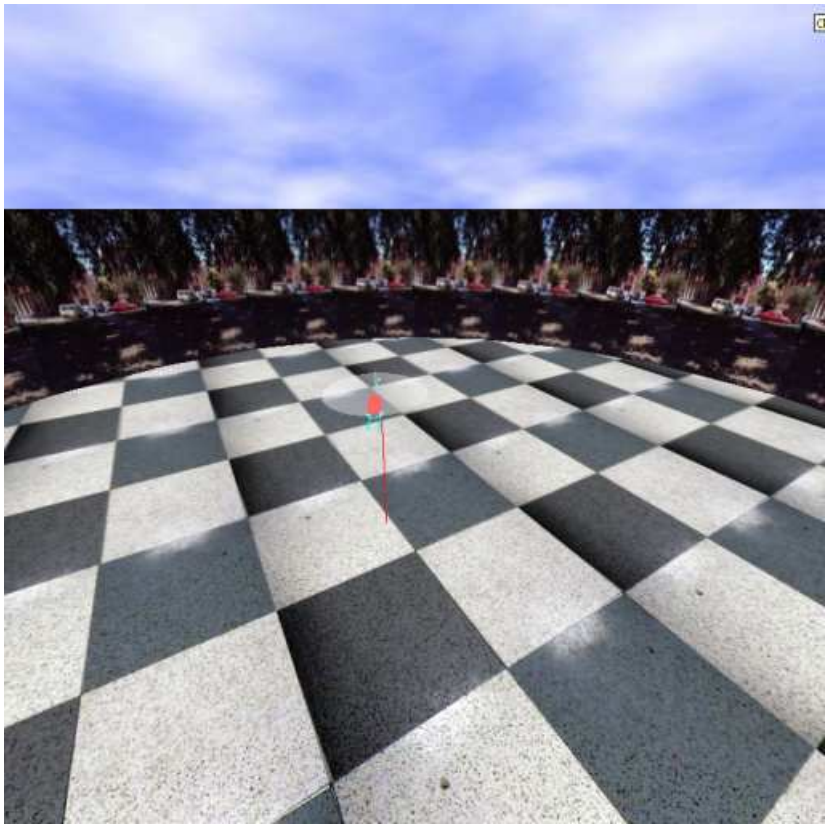


Figure 5.2: Screen Capture from Simulation

used. This was chosen to simulate a panoramic view of the environment surrounding the vehicle given by the parabolic mirror on the quadrotor vehicle used in flight testing [5]. These cameras were constrained to move and rotate with the vehicle. A diagram of this camera configuration, with six cameras instead of twelve, is shown in Figure 5.1. Optic flow was calculated, along with WFI coefficients, at a rate of 55 Hz, which is comparable to the vehicle acquisition rates [5]. Optic flow as noted in Chapter 2 was calculated using two different methods. The first was the so called Gradient Method and the second was the Lucas-Kanade algorithm. The Gradient Method was implement using custom code while the Lucas-Kanade algorithm used is available in the open-source Intel®OpenCV library. Dynamical simulation of vehicle dynamics was carried out using the helicopter model given in Chapter 3. This model was integrated using a fourth order Runge-Kutta scheme with a constant time-step. The dynamics were integrated at 110 Hz, or twice the rate of optic flow capture. This rate was separated by more than an order of magnitude from the nominal closed-loop dynamics and thus was adequate to capture the system dynamics. Due to the noise associated with Gradient Method scaling of the gain matrix was necessary to ensure stability. The constant of this scaling was tuned manually until a near optimum value was obtained. Additionally, a low-pass filter was implemented to lessen the noise on the measurements gotten form WFI. For comparability this filter was left on in the Lucas-Kanade case so that any time delay introduced by the filter would be experienced in both test cases.

5.2 Wind Models

Wind gusts in the simulation were generated using two different methods. First, was the Von Karman power spectrum density(PSD). This was chosen due to its common use in designing aircraft for gust-worthiness [17, 8]. The Von Karman PSD can be given by

$$S(\Omega) = \sigma_w^2 \frac{l}{\Omega} \frac{1 + \frac{8}{3}(1.339 l\Omega^2)}{[1 + (1.339 l\Omega^2)]^{11/6}} \tag{5.1}$$

where S is the PSD and l, σ_w and Ω are the turbulence scale, root-mean square amplitude and frequency respectively. Selection of values for L and σ_w are chosen based on the design altitude of the vehicle h and the wind magnitude at 20ft altitude g_{20} using the following equations [47, 8]

$$\frac{l}{h} = \frac{1}{(0.177 + 0.000823h)^{1.2}} \tag{5.2}$$

$$\frac{\sigma_w}{g_{20}} = \frac{.1}{(0.177 + 0.000823h)^{0.4}} \tag{5.3}$$

Since the Von Karman PSD is not a function of Ω^2, building a linear shaping filter is not possible. However, it can be closely approximated using the following linear shaping filter [17].

$$\frac{g_i}{\eta} = \frac{\sigma_w \sqrt{\nu}}{\pi \sqrt{\Phi_\eta}} \frac{(1 + 2.187\nu s)(1 + 0.1833\nu s)(1 + 0.021\nu s)}{1 + 1.339\nu s)(1 + 1.118\nu s)(1 + 0.1277\nu s)(1 + 0.0146\nu s)} \tag{5.4}$$

where η is a white noise input, g_i is the wind velocity output indirection i, Φ_η if the PSD of the white noise and $\nu = l/V$ where V is the velocity of the turbulence relative to the vehicle. Instead of the usual convention of assuming the turbulence

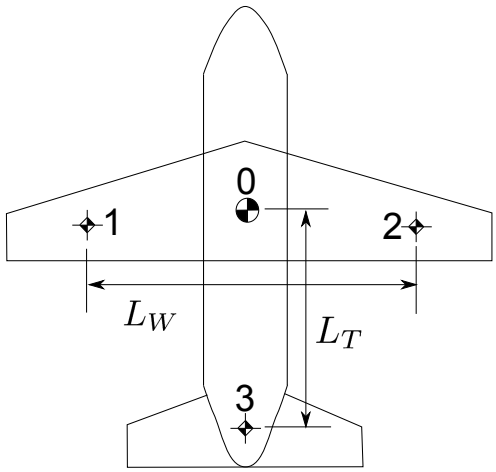


Figure 5.3: The Four Point Model is used to calculate rotational winds on fixed wing aircraft

generated by the Von Karman PSD is stationary with the aircraft moving across it, this study assumed that the turbulence was moving at $V = 1m/s$ relative to the vehicle thus effectively making $g(x)$ in (5.4) into $g(t)$. This was done because this study was concerned with vehicles about hover and thus a stationary wind profile is not realistic. A similar method was applied in [47].

The second wind model used was a first order white noise driven, Gauss-Markov process generated signal as used in [12]. The Gauss-Markov process can be given by

$$\dot{g}_i = \frac{-1}{\tau} g_i + \zeta\omega \tag{5.5}$$

where g_i is the wind gust affecting state i, τ is the autocorrelation time, ζ a constant and ω unit white noise. Values for this process were $\tau = 3.2$ and ζ was scaled to match the mean to that of the Von-Karman PSD. To calculate the rotational wind

Table 5.1: Simulation Test Cases

	Gradient Method		Lucas-Kanade	
	Corrected	Uncorrected	Corrected	Uncorrected
Von-Karman PSD	1	2	5	6
Guass-Markov	3	4	7	8

a modified version of the Four Point Model, which is commonly used for fixed wing aircraft, was used. This model takes the wind at four points on the aircraft and uses these to approximate the rotational wind components. One point is the center of gravity and the other three are placed on each wing and the tail. This is shown in Figure 5.3. This study was only concerned with the yaw direction of wind which is given by [11]

$$g_r = \frac{1}{L_T}(g_{v,0} - g_{v,3}) + \frac{1}{L_W}(g_{u,1} - g_{u,2}) \tag{5.6}$$

where $g_{u,i}, g_{v,i}$ and $g_{r,i}$ are the longitudinal, lateral and yaw rotational gusts at point i respectively. However since this simulation is of a rotorcraft, the second term is dropped due to the lack of wings. As such, the rotational wind is calculated as

$$g_r = \frac{1}{L_T}(g_{v,0} - g_{v,3}) \tag{5.7}$$

Table 5.2: Timing of Events in Simulation Trials

Time (sec.)	Event
0	Simulation start
0.2	Vehicle control turned on
1	Wind disturbance begins
26	Wind disturbance ends
30	Simulation end

5.3 Test Cases

In addition to simulating the vehicle under two different wind disturbances and using two separate optic flow algorithms, in order to show the advantage of using the correction for the body-inertial frame, given in Chapter 4, simulations were performed with this correction included and omitted. The combination of these three pairs of conditions lead to eight separate configurations for simulation. These are shown in Table 5.1 and will be presented in the order given therein.

For each of the eight test cases 100 trials in the simulation were performed to accurately capture the variations caused by the random wind. The environment for the simulation was chosen as a circular room twelve meters across. The walls of the room were textured using realistic outdoor imagery. Each simulation trail was 30 seconds long. The vehicle was started in the center of the room with no initial motion. For the first 0.2 seconds all control was disabled to allow for multiple camera frames to be processed into optic flow and thus avoid any large estimates that might

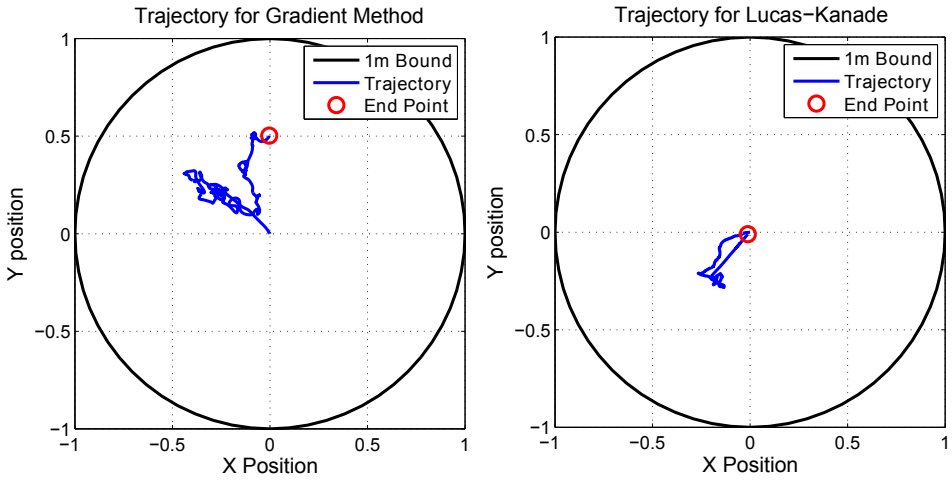


Figure 5.4: Characteristic Simulation Trajectories: 1-meter bound shown for scale and does not reflect environment size

result during the first few frames. Beginning at 1 seconds, the wind disturbance was applied to the vehicle lasting for 25 seconds. Following this the vehicle was given the remaining 4 seconds to attempt to return to the starting position. This run structure is shown in Table 5.2. Trajectories of two trials are shown in Figure 5.4 one for each optic flow algorithm. A one meter bound is plotted to show scale and does not reflect the room size.

5.4 Simulation Results

For each case three separate histograms are presented here. The first shows the error in the final position of the vehicle relative to its starting position. The second shows the maximum distance from the starting position during the trial. Last, is a histogram of the mean wind velocity for each trial. The dashed red line shows the mean of each data set.

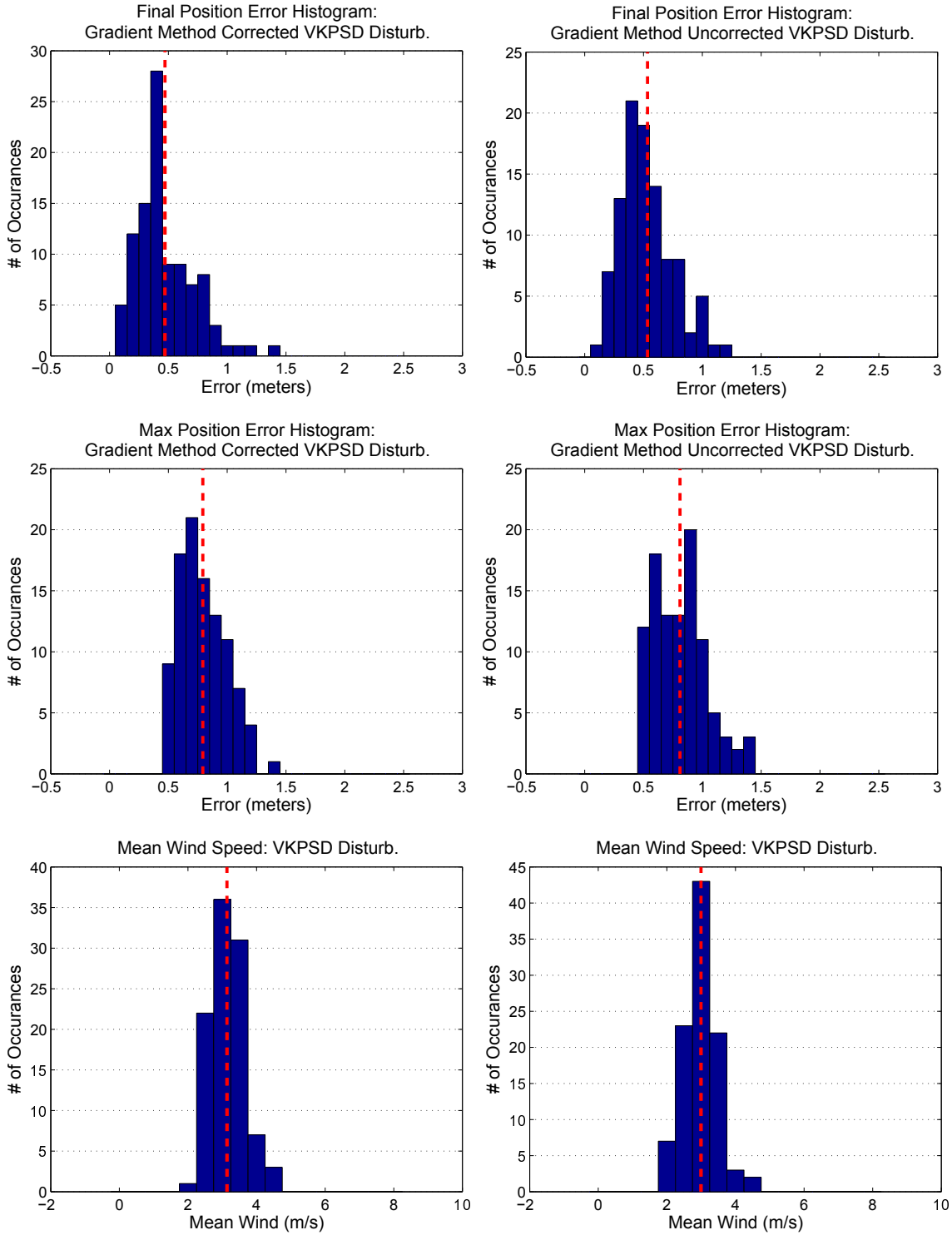


Figure 5.5: Runs 1/2. Gradient Method with/without Correction under Von-Karman PSD Disturbance

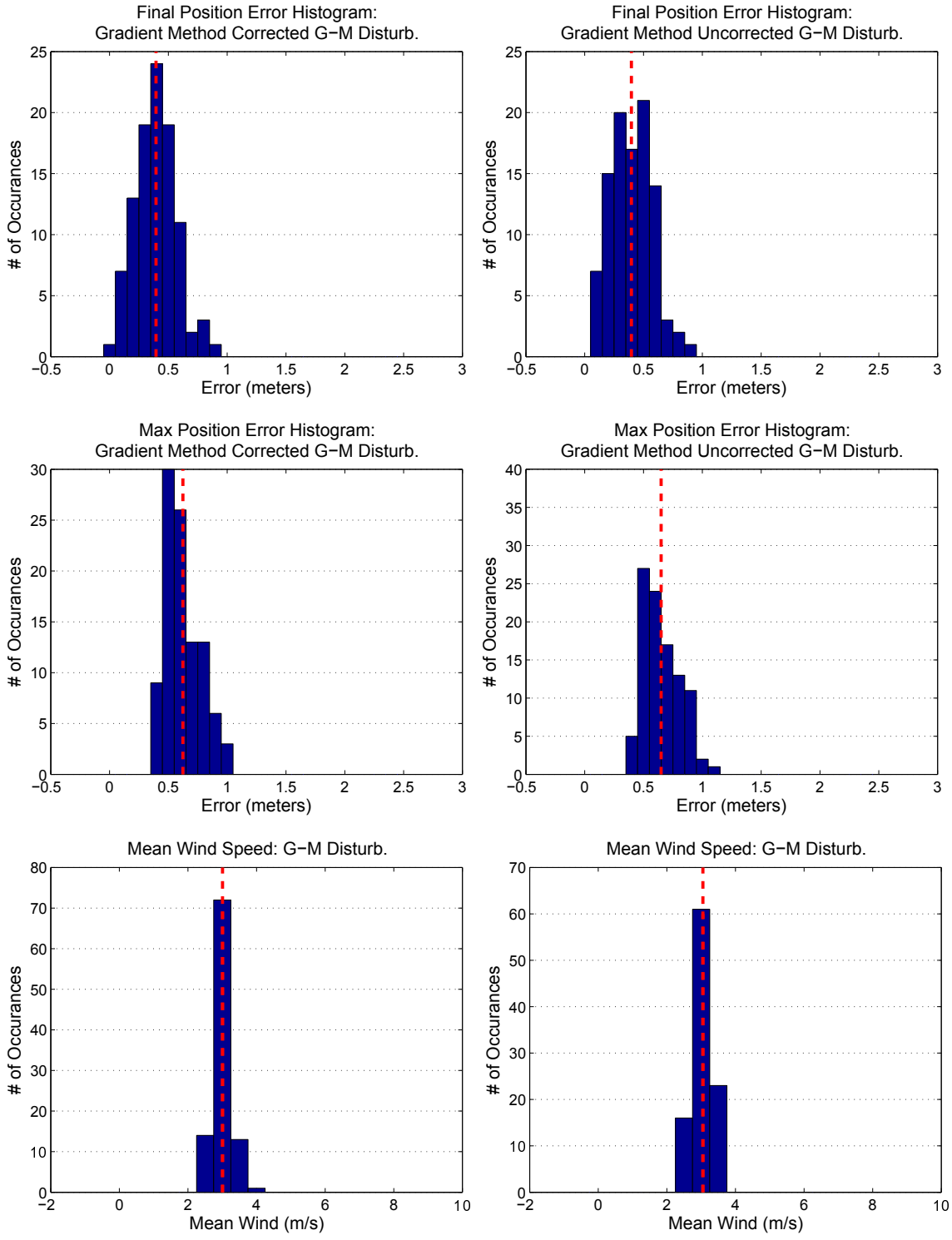


Figure 5.6: Runs 3/4. Gradient Method with/without Correction under Gauss-Markov Disturbance

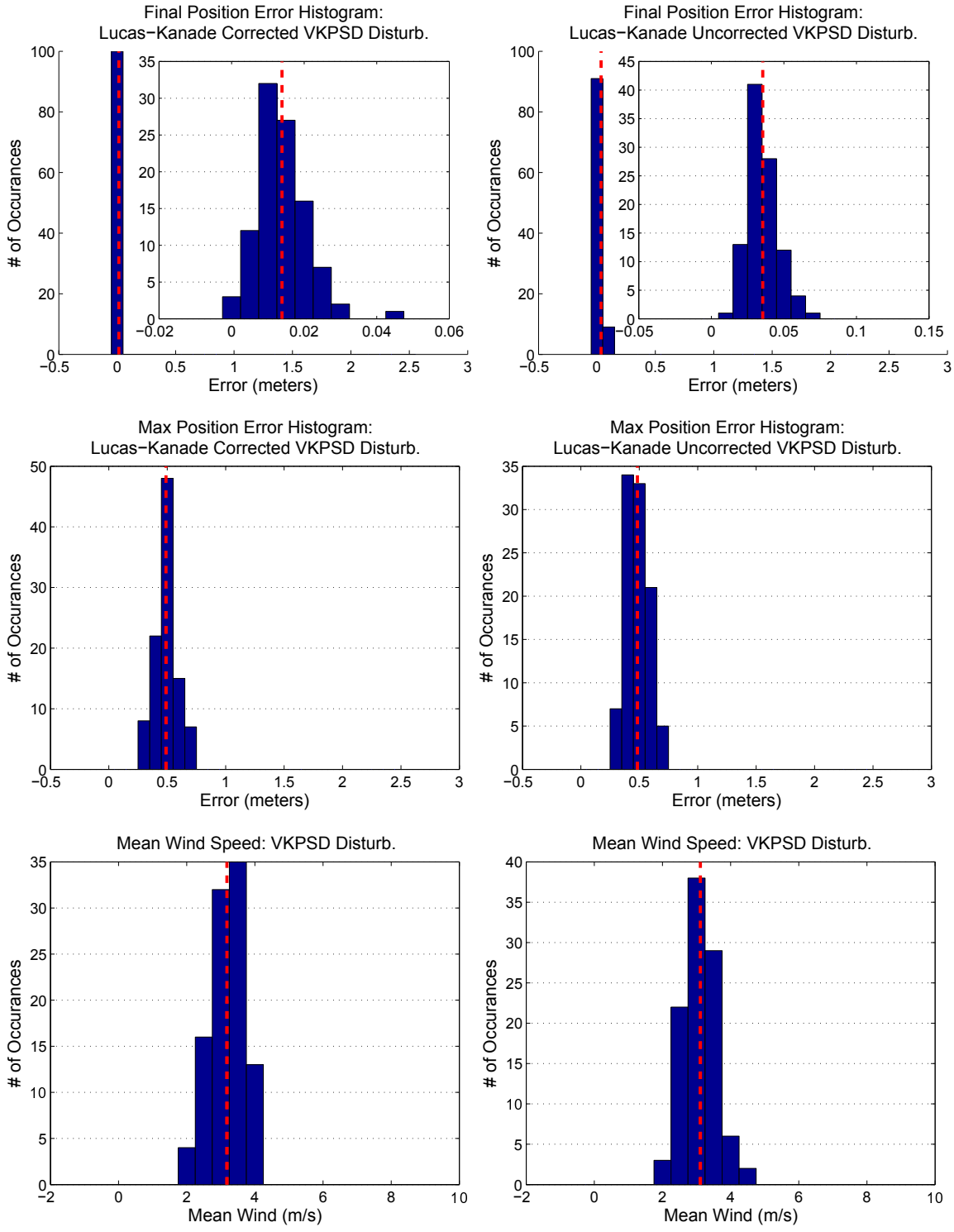


Figure 5.7: Runs 5/6. Lucas-Kanade with/without Correction under Von-Karman PSD Disturbance

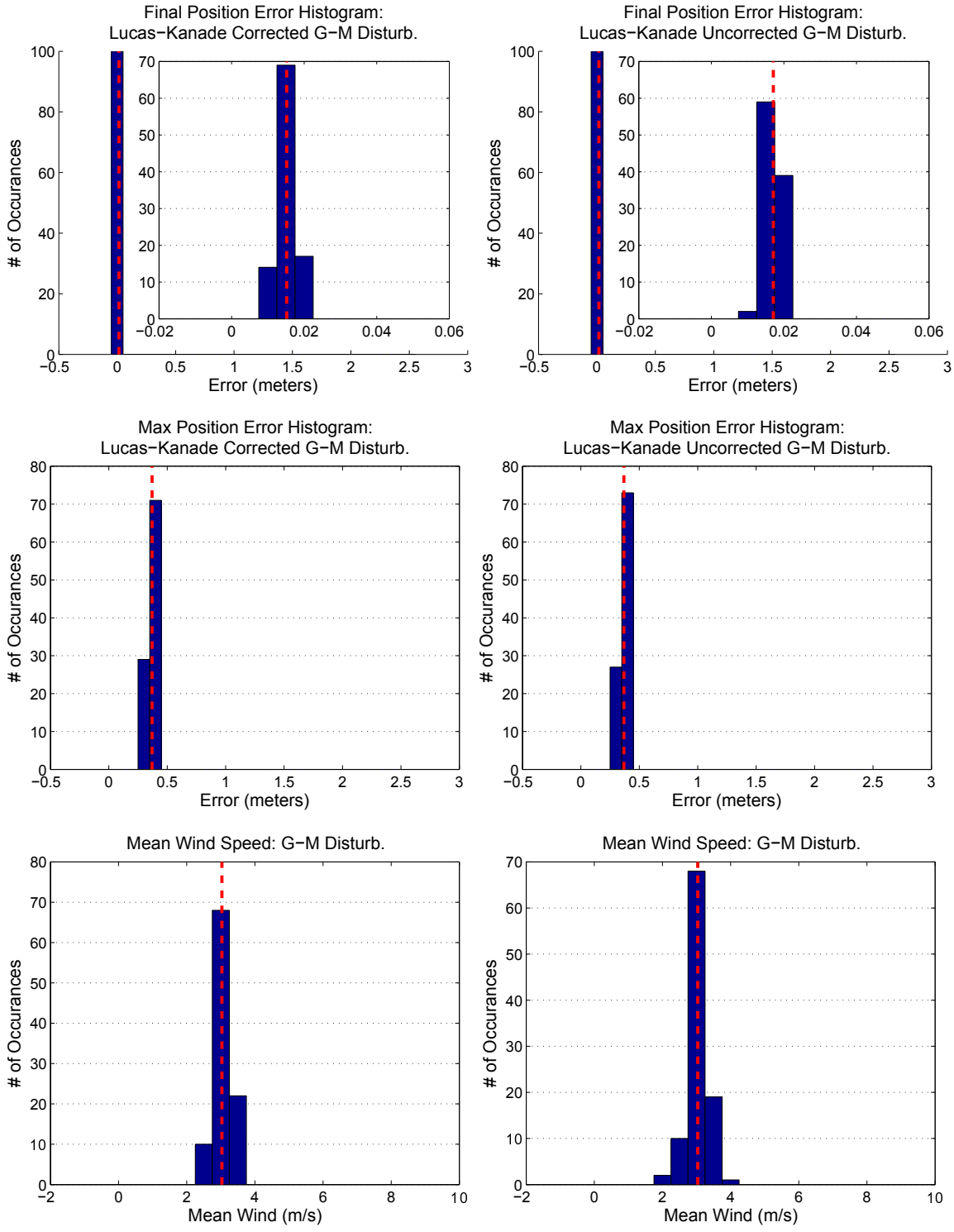


Figure 5.8: Runs 7/8. Lucas-Kanade with/without Correction under Gauss-Markov Disturbance

The histograms of simulation results reveal a number of trends. First, the Gauss-Markov disturbance has a smaller variance than the Von-Karman PSD disturbance. Despite this, the results for both disturbances center around similar values. However, the increased variance of the Von-Karman PSD disturbance does result in higher variance in the error as might be expected. Furthermore, in general the Lucas-Kanade algorithm preforms significantly better than the Gradient Method in terms of final position errors and and to a lesser extent maximum error. Looking at the results for the Gradient Method it can be seen that while the it kept the vehicle in an the general area of the origin this area was significantly bigger than was accomplished by Lucas-Kanade. Furthermore, Gradient Method exhibited less of the returning response that was seen with Lucas-Kanade. Any benefit gained through the body-inertial frame correction is not apparent from the histograms.

Looking at the means presented in Table 5.3 of each run of trials the improvement from the correction can be seen more clearly. Comparison of all four pairs of uncorrected and corrected runs shows that in each case a improvement is seen in the final position error (Table 5.3a). Additionally, the final error is slightly smaller in all cases for the Von-Karman PSD disturbance. The maximum error, in Table 5.3b, shows little improvement and with the corrected case being worse for the case of Lucas-Kanade under the Gauss-Markov disturbance. Finally, Table 5.3c shows the mean of the wind magnitudes across the different runs which are all similar for each disturbance type which suggests comparability of results.

Reduction of this data further by taking the ratio between the uncorrected and corrected means gives an mean percent improvement for the correction. Which as

Table 5.3: Mean Values for Trial Data

(a) Final Position Error Means

	Gradient Method		Lucas-Kanade	
	Corrected	Uncorrected	Corrected	Uncorrected
Von-Karman PSD	0.4710	0.5335	0.01389	0.03548
Gauss-Markov	0.3950	0.3965	0.01519	0.01711

(b) Max Position Error Means

	Gradient Method		Lucas-Kanade	
	Corrected	Uncorrected	Corrected	Uncorrected
Von-Karman PSD	0.7930	0.8096	0.4894	0.4838
Gauss-Markov	0.6236	0.6495	0.3695	0.3679

(c) Mean Wind Means

	Gradient Method		Lucas-Kanade	
	Corrected	Uncorrected	Corrected	Uncorrected
Von-Karman PSD	3.137	2.995	3.181	3.113
Gauss-Markov	3.007	3.052	3.032	3.035

Table 5.4: Percent Improvement Due to Correction

(a) Final Position Error

	Gradient Method	Lucas-Kanade
Von-Karman PSD	13.3%	155%
Gauss-Markov	0.37%	12.6%

(b) Max Position Error

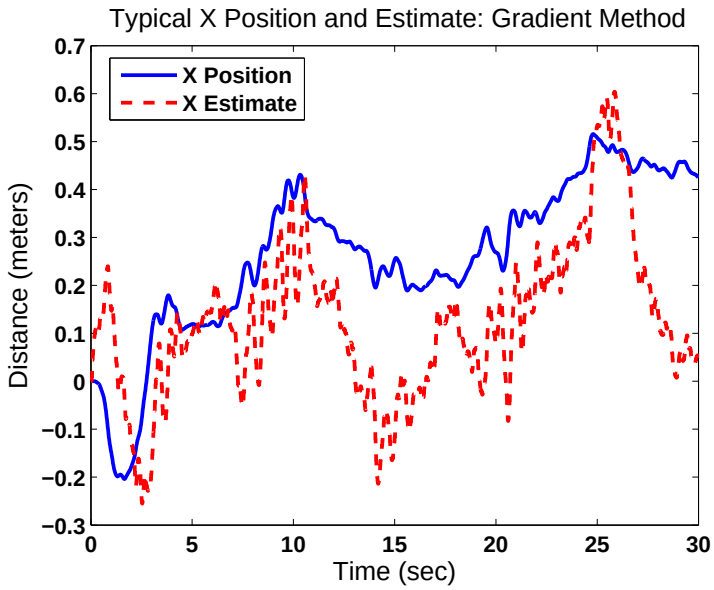
	Gradient Method	Lucas-Kanade
Von-Karman PSD	2.1%	-1.1%
Gauss-Markov	4.2%	0.4%

can be seem in Table 5.4a the correction shows between a 0.37% and 155% improvement over the uncorrected case. On average this results in a 45.3% improvement. However, this quantity is skewed due to the large improvement under Lucas-Kanade and the Von-Karman disturbance. For the maximum error, shown in Table 5.4b, no trend in improvement is seen. This suggests that the correction has little effect on the ultimate bound between the wind and maximum position error but rather provides the controller with a more accurate estimate of the position error thus allowing a better return behavior in the vehicle.

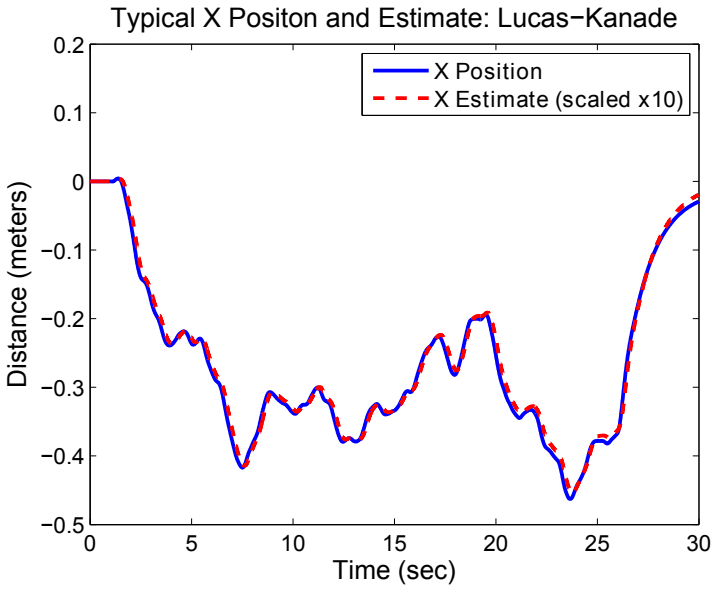
Examination of the state estimate for the two different optic flow algorithms gives insight into the difference in final position errors. Looking at Figure 5.9 it can be seen that the Gradient Method does capture the features of the position to

Figure 5.9: Comparison of Position Estimates Accuracy

(a) Gradient Method



(b) Lucas-Kanade



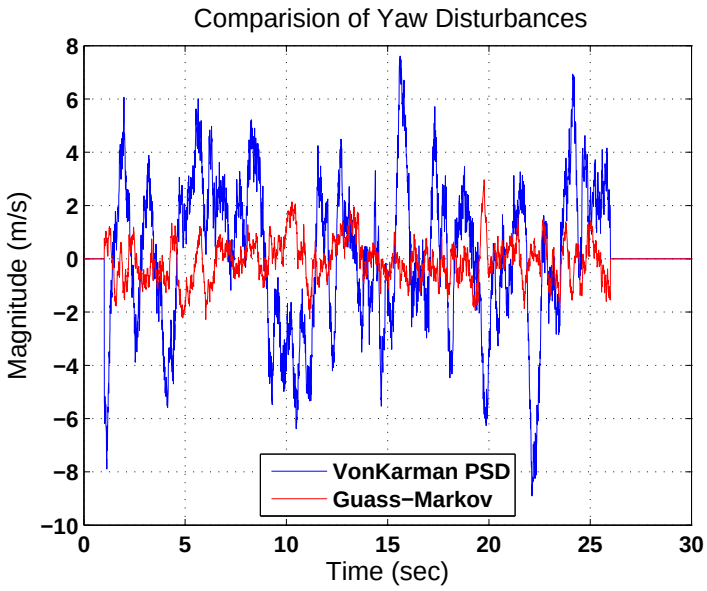


Figure 5.10: Comparision of Yaw Wind Components

some degree but, the accumulation of drift causes inaccuracy over time. Meanwhile, Lucas-Kanade accurately represents the position though the estimate, due to the limitation of optic flow, is only proportional and here has been rescaled to the match the scale of the position. This data reflects the treads presented in the histograms in figures 5.5, 5.6, 5.7 and 5.8. Furthermore, examination of the difference in the rotational component of the wind for the two disturbance types (see Figure 5.10) shows a much larger amplitude in the case of the Von-Karman PSD. This larger wind causes larger rotations of the vehicle thus the correction is accounting for larger discrepances.

5.5 Conclusions from Simulation

A couple of conclusions can be drawn from the simulation data above. The first is that the Gradient Method is not well suited for station-keeping and may be in-

sufficient to accomplish the goals of this study. However, a more accurate algorithm such as Lucas-Kanade appears to be able to met realistic performance requirements in both station-keeping and wind rejection. Second, is that the correction for the inertial and body frame discrepancies does show a significant improvement over the uncorrected case. However, the size of this improvement may be too small to warrant implementation over the simpler uncorrected case in realistic applications.

Chapter 6

Flight Testing

In addition to simulation, flight testing was performed to demonstrate the real-world performance of the control methodology. This testing was performed using the quadrotor vehicle presented in Chapter 3. The Gradient Method discussed in Chapter 2 was used for sensing. While this algorithm showed shortcomings in simulation results (see Chapter 5) performance measures were obtained. Three separate tests were performed for station-keeping, discrete gust disturbances and sustained gust disturbances. In this Chapter the procedure and results of this testing is presented. Additionally, conclusions are drawn as to the performance and overall usability of the proposed control algorithm presented in Chapter 4.

6.1 Sensors and Instrumentation

For the purposes of flight testing the quadrotor vehicle (see Chapter 3) was employed. This vehicle was equipped with a camera to obtain optic flow measurements. In order to get a panoramic view of the surroundings a panoramic mirror was used. Optic flow calculations were performed by sampling rings of pixels out of the resulting image and averaging the resulting optic flow between them (see Figure 6.1). This optic flow was transmitted to a ground station which performed control calculations using LABVEIW^{TM}.

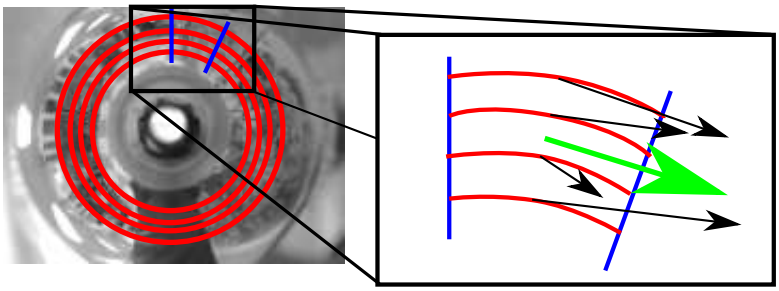
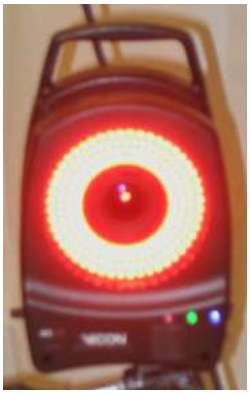
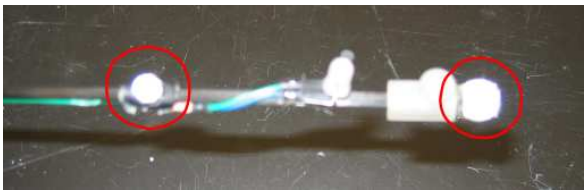


Figure 6.1: Optic flow was calculated using rings of pixels extracted from the image of a parabolic mirror. The results across these rings were averaged to obtain a more accurate measurement.



(a) Vicon Camera



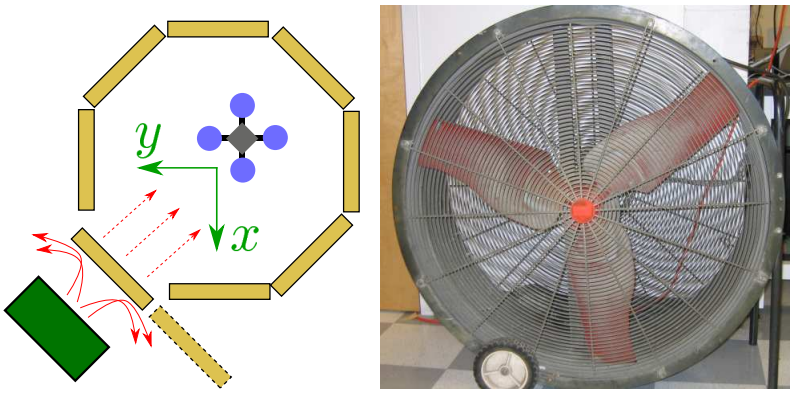
(b) Vicon Markers

Figure 6.2: Vicon Camera and Markers

In addition to optic flow measurements a visual positioning system manufactured by Vicon^{TM} was employed. This system was used both to record vehicle position and orientation as well as providing the heading of the vehicle for feedback control. Vicon's positioning system works by using multiple high speed cameras equipped with strobe lights of a specific wavelength. The cameras monitor the position of retro-reflective markers by monitoring the reflection on the strobe lights. From these reflections the position of markers can be triangulated. The measurements are accurate to less than a millimeter and measurements are taken at 350Hz. The cameras and markers are shown in Figure 6.2.

6.2 Vehicle Control

Vehicle control was handled off-board the vehicle. Measurements from the on-board camera were transmitted to a ground-station via a bluetooth connection. Additionally, this computer would receive data from the Vicon^{TM} positioning system. Using a LABVIEW^{TM} program the ground-station would calculate the control inputs and transmit these via bluetooth back to the vehicle. During takeoff and trial setup the vehicle would be controlled using a PID-controller and data supplied by the positioning system. Additionally, joystick input was used to set the position to be held. Once the trial pressing and holding a button on the joystick would switch to the optic-flow based control. Under this mode the optic flow data would be used for feedback with the heading angle being provided by the positioning system. While this mode was engaged joystick inputs were ignored.



(a) Testing Environment (b) 36-inch drum fan used to introduce gust disturbances

Figure 6.3: Flight Testing Environment

6.3 Environment and Wind Gusts

The vehicle operated in a roughly circular environment approximately two meters in radius will walls textured in a random barcode. This pattern was chosen to give good optic flow. However, any image with good contrast and spatial variation should perform similarly. Wind gusts were supplied using a 36 inch drum fan shown in Figure 6.3. During wind gust testing the section of wall in from of the fan would be removed and replaced to start and stop the gust applied to the vehicle. This section of wall was left untextured to avoid introducing optic flow when it was removed. Additionally, this wall was fitted with Vicon^{TM} position markers so that the timing of the gust would be available in the recorded data. A diagram of the environment layout is given in Figure 6.3.

In order to examine the qualities of the airflow supplied by the fan a LaCross Technology®EU-3010U handheld anemometer (see Figure 6.6) was used to sample

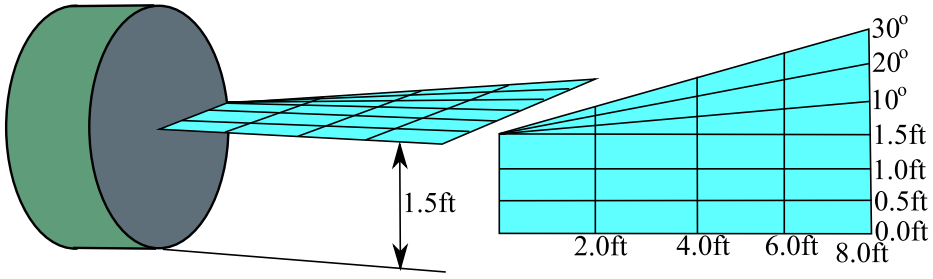
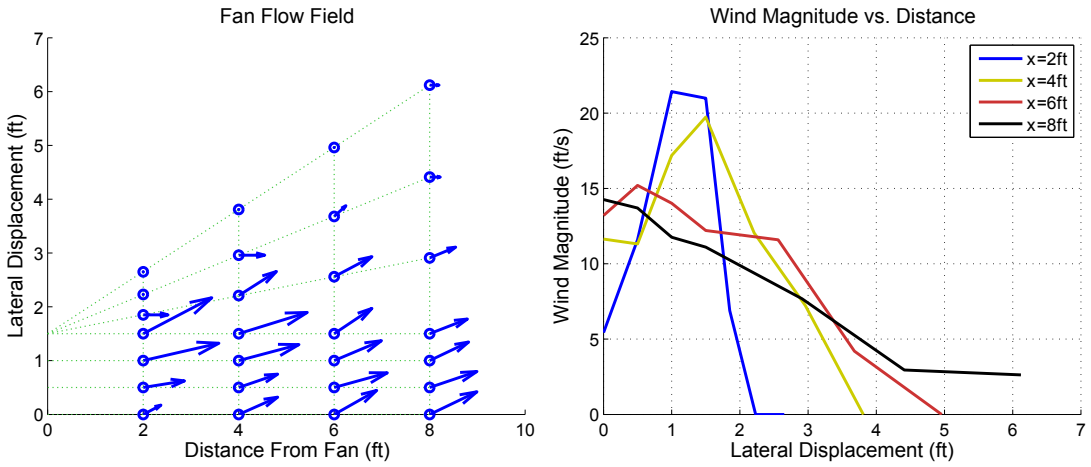


Figure 6.4: Flow Velocity Sampling Grid: Inside the fan radius the grid was Cartesian while outside and angular grid was used.



(a) (b)

Figure 6.5: Gust Fan Flow: (a) Flow field and sampling grid (b) Profiles of wind magnitude at four distances

Figure 6.6: Anemometer used to measure fan flow

flow velocities throughout the flow field. Due to the rotational symmetry of the fan flow the sampling grid used stated at the centerline of the fan and extended laterally to the edge of the fan. From the edge of the fan the grid was then extended in an angular fashion to capture any spreading of the flow with distance. Figure 6.4 shows the design of the sampling grid. At each point on the grid the outward and lateral wind flow was measured to give the direction and magnitude of the wind. In many cases measurements were periodic in time and a central value was taken. Due to the limitations of the anemometer the direction (sign) of the flow is not captured. While this is not problematic for the outward flow the lateral flow direction could likely vary. For the results shown in Figure 6.5 the direction has been assumed to be outward from the fan centerline. Even with this assumption it is possible to see the flow is relatively laminar. Moreover, an angular spread between 10 and 20 degrees can be noted. Additionally, the magnitude profiles plot in Figure 6.5 show that close to the fan the flow has a central low velocity region with very strong lobes towards the edge of the fan. Furthermore, the as distance increases these lobes diminish and the central velocity increases to produce a more Gaussian like profile. Additionally, the flow near the center of the profile is approximately 10-15 ft/s (3-4.6 m/s or 6.8-10.2 mph). While this velocity is faster than the 3 m/s used in simulation (see Chapter 5) it does not vary greatly.

6.4 Station-Keeping Testing

The first type of testing undertaken was under still-air conditions. This had two purposes. First, to establish a baseline behavior for the vehicle under optic flow control. Second, was to characterize the integral term drift present in the algorithm implementation.

In each individual trial the vehicle would initially operate under Vicon^{TM} supplied position control. Once the vehicle was in the air and settled the optic flow control was engaged. Ideally, this control would hold the same position as the Vicon^{TM} control however due to the discontinuity of this control switch in practice the vehicle would attempt to hold another position. Thus, there was a period of time where the vehicle attempts to reach this position. This position was taken to be the first sharp change in direction exhibited by the vehicle. This will be referred to as the settling point. Following this the vehicle was flown under optic flow control until conditions warranted ending the trail (ie. wall collisions, insufficient battery charge, etc.).

Results for ten trails are presented in Figure 6.7. The results are presented in two plots. The first gives the distance from the settling point throughout the trial run. Additionally, a linear regression line is plotted to show the average drift rate of the run. The time of the settling point is also marked. The second plot shows the vehicle trajectory. Additionally, the environment walls (black) and bounds of collision (dashed red) are approximated.

Examining the results presented in Figure 6.7 a couple of qualitative state-

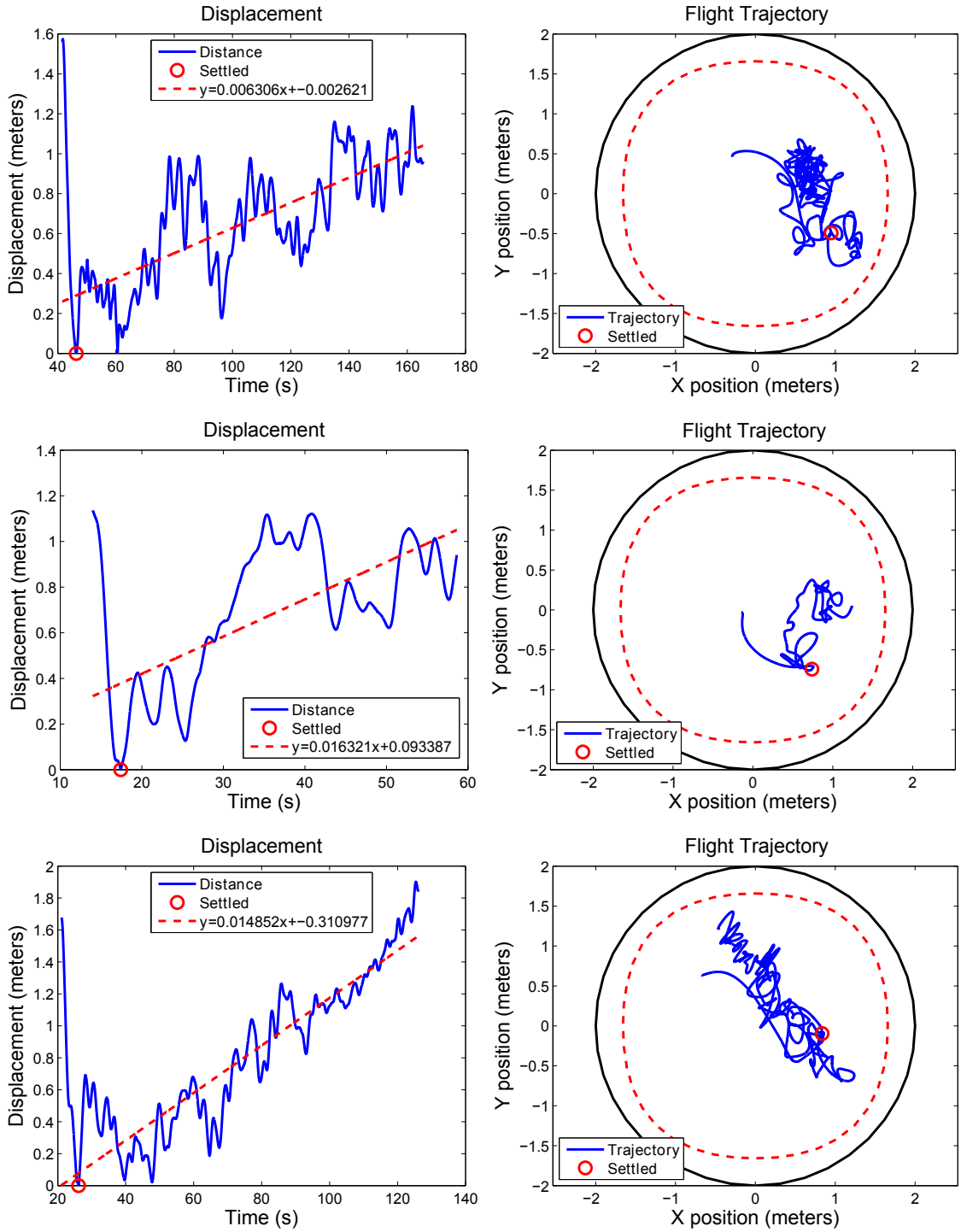


Figure 6.7: Trails 1-3: Station-Keeping Testing

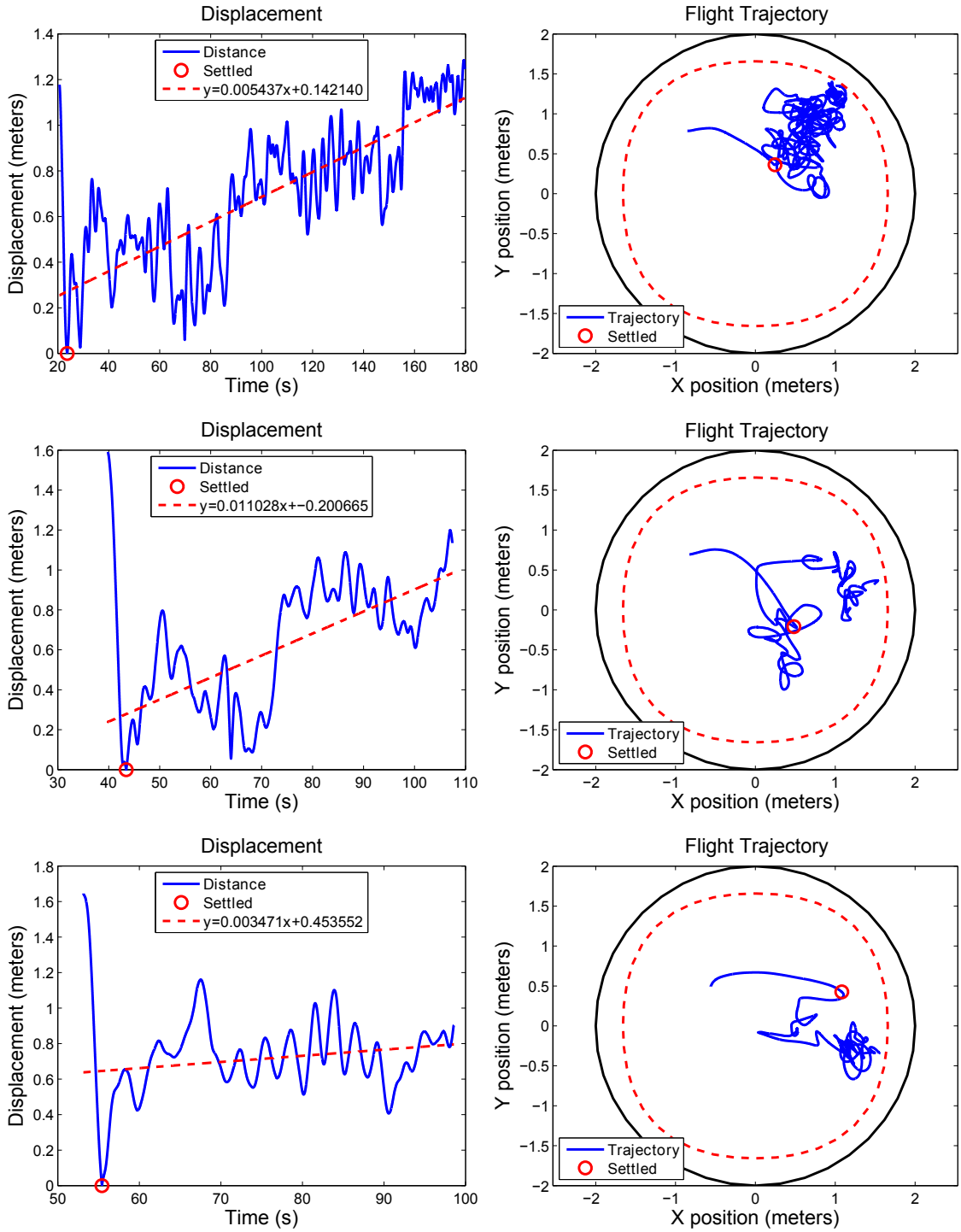


Figure 6.7: Trails 4-6: Station-Keeping Testing

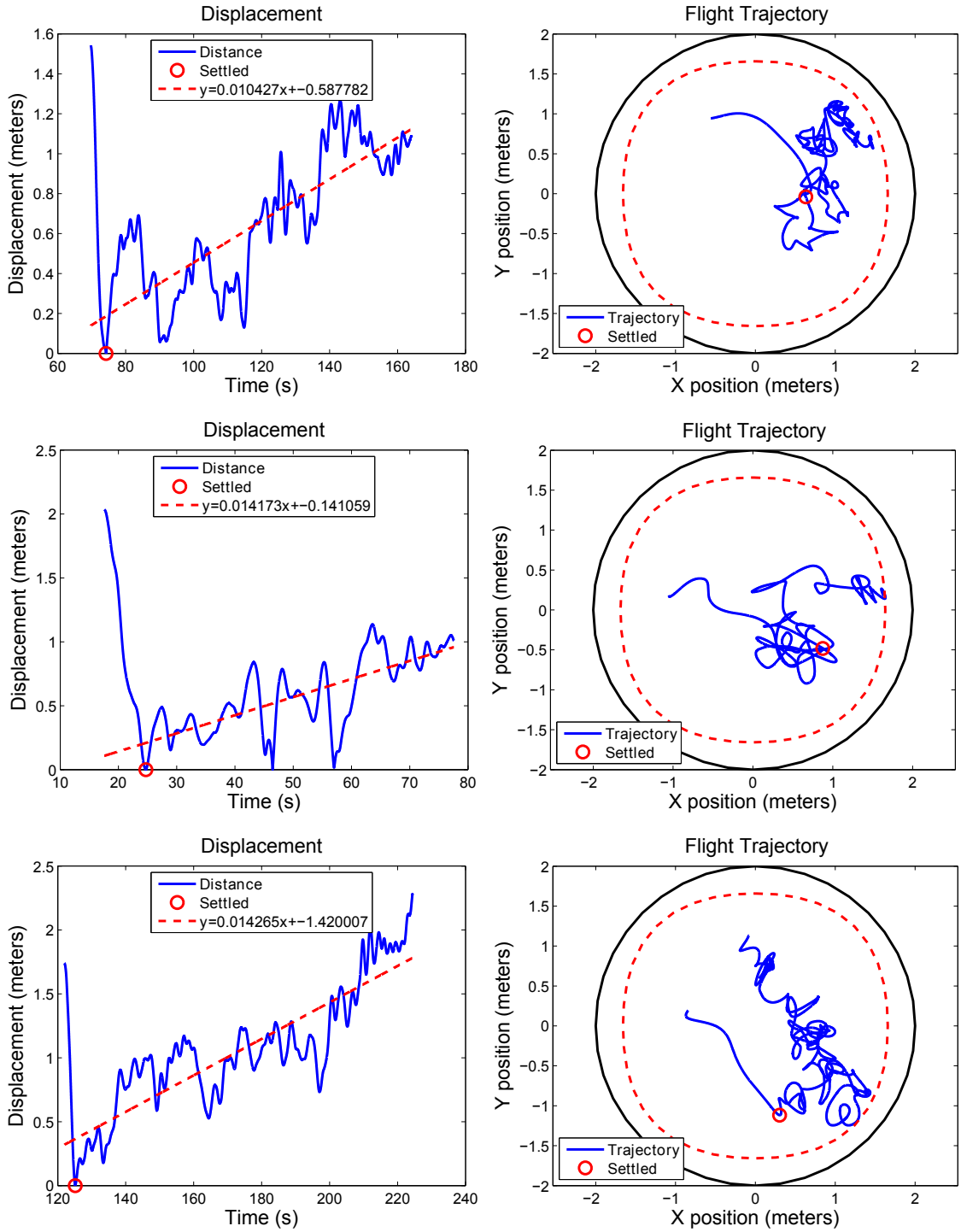


Figure 6.7: Trails 7-9: Station-Keeping Testing

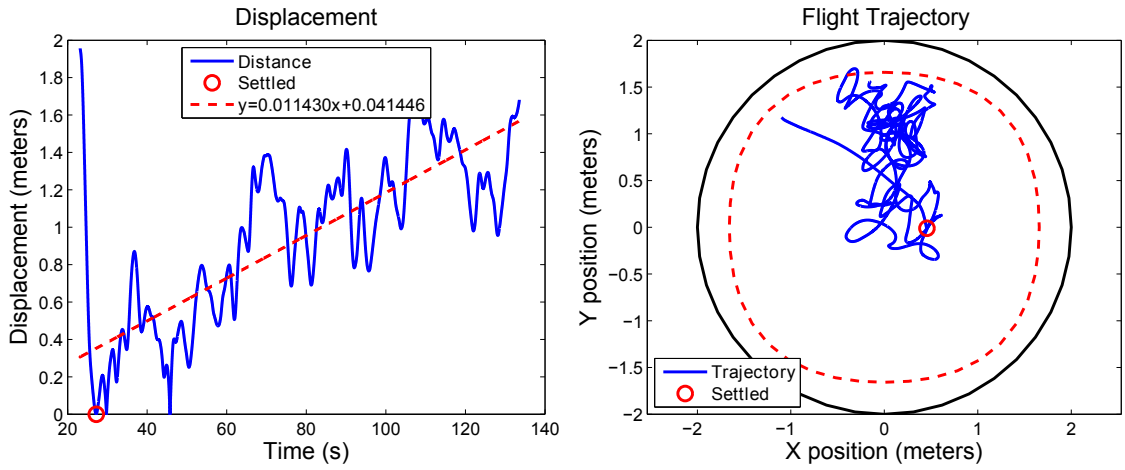


Figure 6.7: Trail 10: Station-Keeping Testing

Trial	Mean Drift	Max Disp.	Trial Length	Max Drift
—	meters/sec	meters	seconds	meters/sec
01	0.00631	1.237	118.95	0.0104
02	0.01632	1.121	41.20	0.0272
03	0.01485	1.902	99.96	0.0190
04	0.00544	1.284	155.99	0.0082
05	0.01103	1.193	63.97	0.0186
06	0.00347	1.161	43.14	0.0269
07	0.01043	1.266	89.87	0.0141
08	0.01417	1.138	52.86	0.0215
09	0.01427	2.281	99.30	0.0230
10	0.01143	1.681	106.40	0.0158
Max	0.01632	2.281	155.99	0.0272
Min	0.00347	1.121	41.20	0.0082
Mean	0.01077	1.426	87.16	0.0184
Median	0.01123	1.252	94.59	0.0188

Table 6.1: Statistics For Station-Keeping Trials

ments can be made. First, the trajectory plots show clearly that the vehicle exhibits station-keeping behavior. Secondly from the displacement plots, the integration drift can be observed. A more quantitative analysis is presented in Table 6.1. Here the mean drift for each trial, as quantified by slope of the trend lines in Figure 6.7, is presented along with the maximum displacement from the settling point, trail length and maximum drift calculated by dividing the maximum displacement by trail length. Additionally, the maximum, minimum, mean and median across all ten trails are shown. The results presented in Table 6.1 shown the mean drift across all trials to be approximates 1.1 cm/sec while the mean maximum drift is approximately 1.8 cm/sec. Drift at these rates results in the vehicle moving 1 meter in 90.9 and 55.6 seconds respectively or roughly a minute and a half and a minute. Looking at the worst case shown in maximum of the maximum drift results in 36.8 seconds to drift a meter.

6.5 Discrete Gust Testing

The first type of wind testing performed involved introduction of short duration, or discrete, wind disturbances. A discrete gust, though not completely realistic, allows for a more simplistic examination of the gust response of the vehicle [17]. The main goal of these tests was to characterize the ability of the vehicle to detect and respond to gust disturbances.

Each individual trail of testing consisted of a number of stages. First, the vehicle would takeoff using position control provided by the Vicon^{TM} system. Once

the vehicle was in position the optic flow based control would be engaged. As discussed in Section 6.4 after switching of controllers the vehicle would move to a new position. After the vehicle is allowed to settle in this position the untextured section of wall covering the fan was removed starting the discrete gust. This gust would last approximately 2-3 seconds before the wall was replaced. Then the vehicle would exhibit a returning behavior. Once the vehicle had returned, its motion was then characterized by a turn in the direction of the drift. The position of this turn was considered to be the point of return. In Figure 6.8 the results of fifteen trials are presented. In the trajectory plots the black and red circles represent the ideal environment walls and collision bounds for the testing however these are approximate. The green lines and arrow show the idealized wind bounds and direction. The trajectory (blue) is plotted between the time of the optic flow control being engaged and disengaged. Additionally, the wind start, wind end and return points are marked. In the displacement plots the displacement from the wind start position in the direction of the wind disturbance is shown. Only the wind direction is shown to attempt to exclude any effects of drift. Finally, the wind start, wind stop and return times are plotted.

Results presented in the plots in Figure 6.8 reveal a couple trends. First, that the vehicle shows a returning type behavior in all trials. Additionally, in about a third trails the point of return in very close to the starting position. A quantitative analysis of each trial along with statistics across all fifteen trails are presented in Table 6.2. The first two columns show the vehicle return error and maximum error in meters from the "wind on" point. The third column shows the return error as

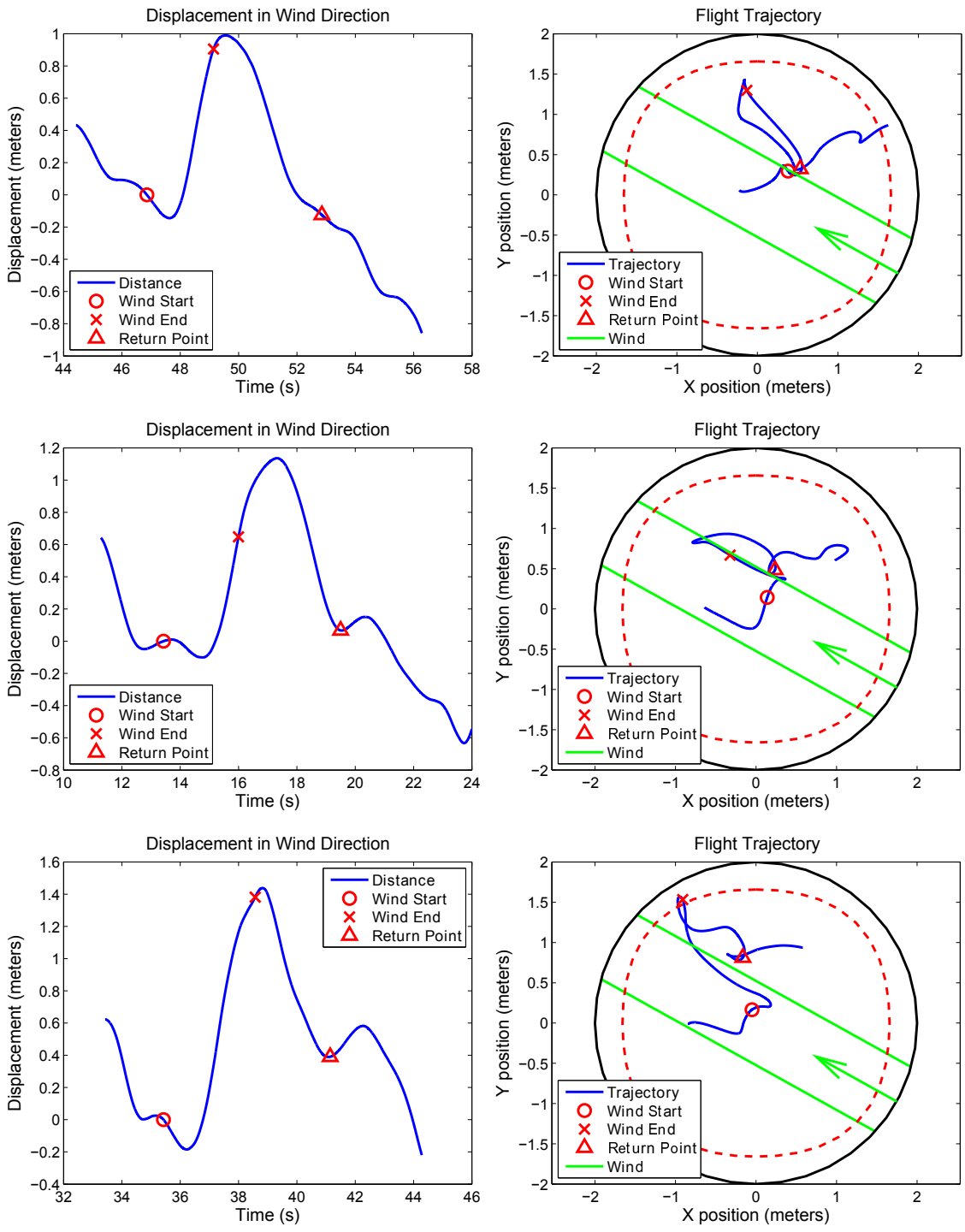


Figure 6.8: Trials 1-3: Discrete Gust Testing

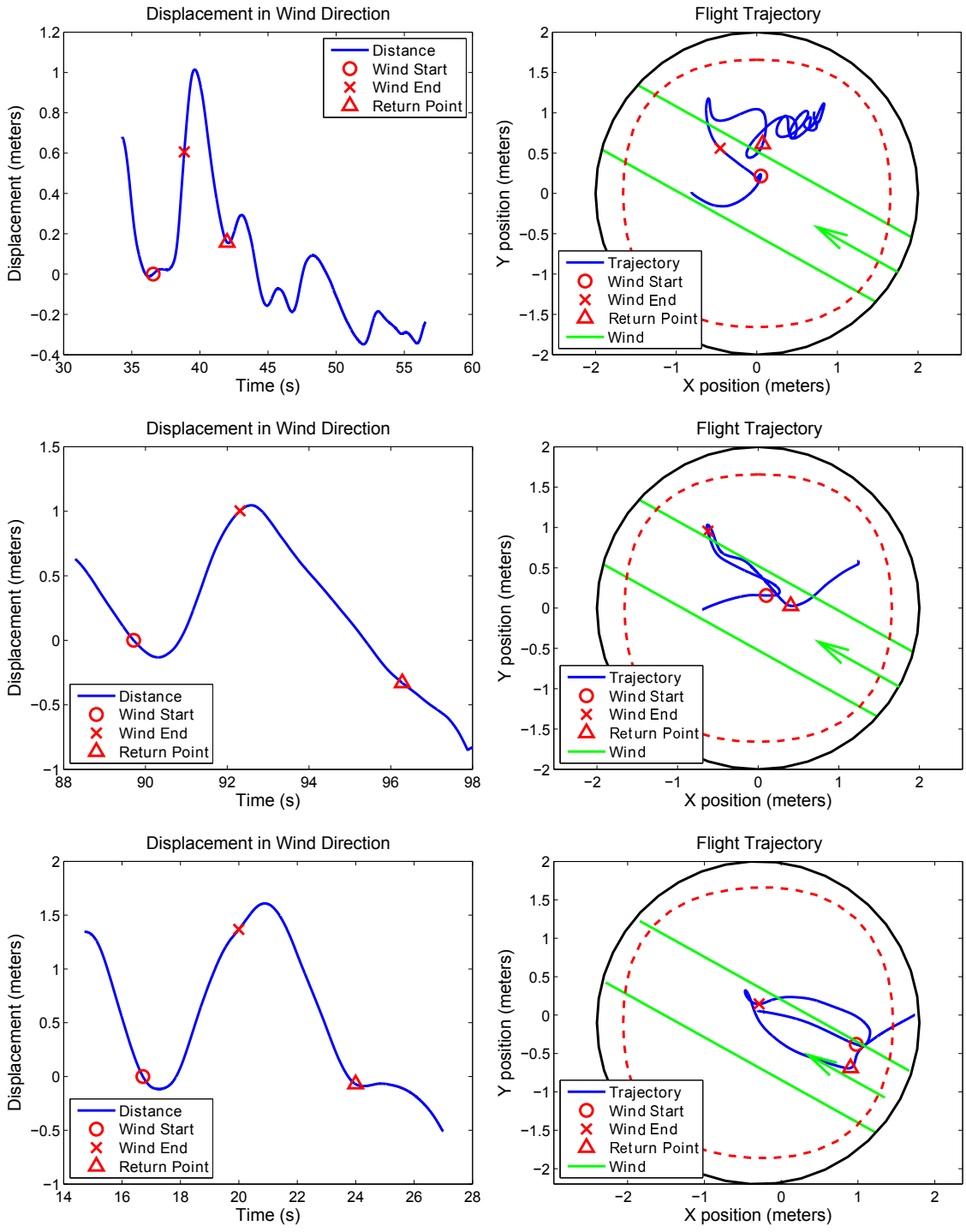


Figure 6.8: Trails 4-6: Discrete Gust Testing

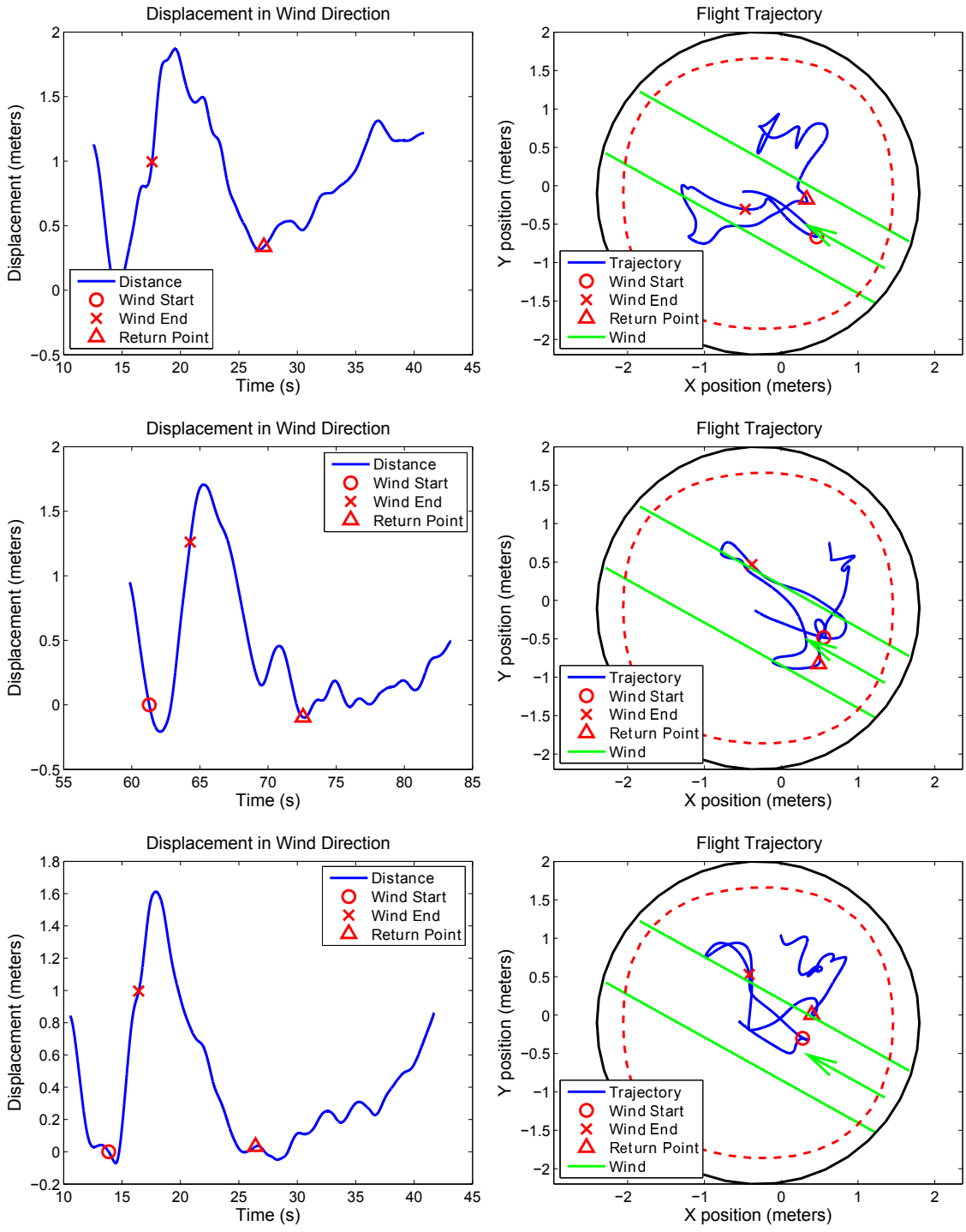


Figure 6.8: Trails 7-9: Discrete Gust Testing

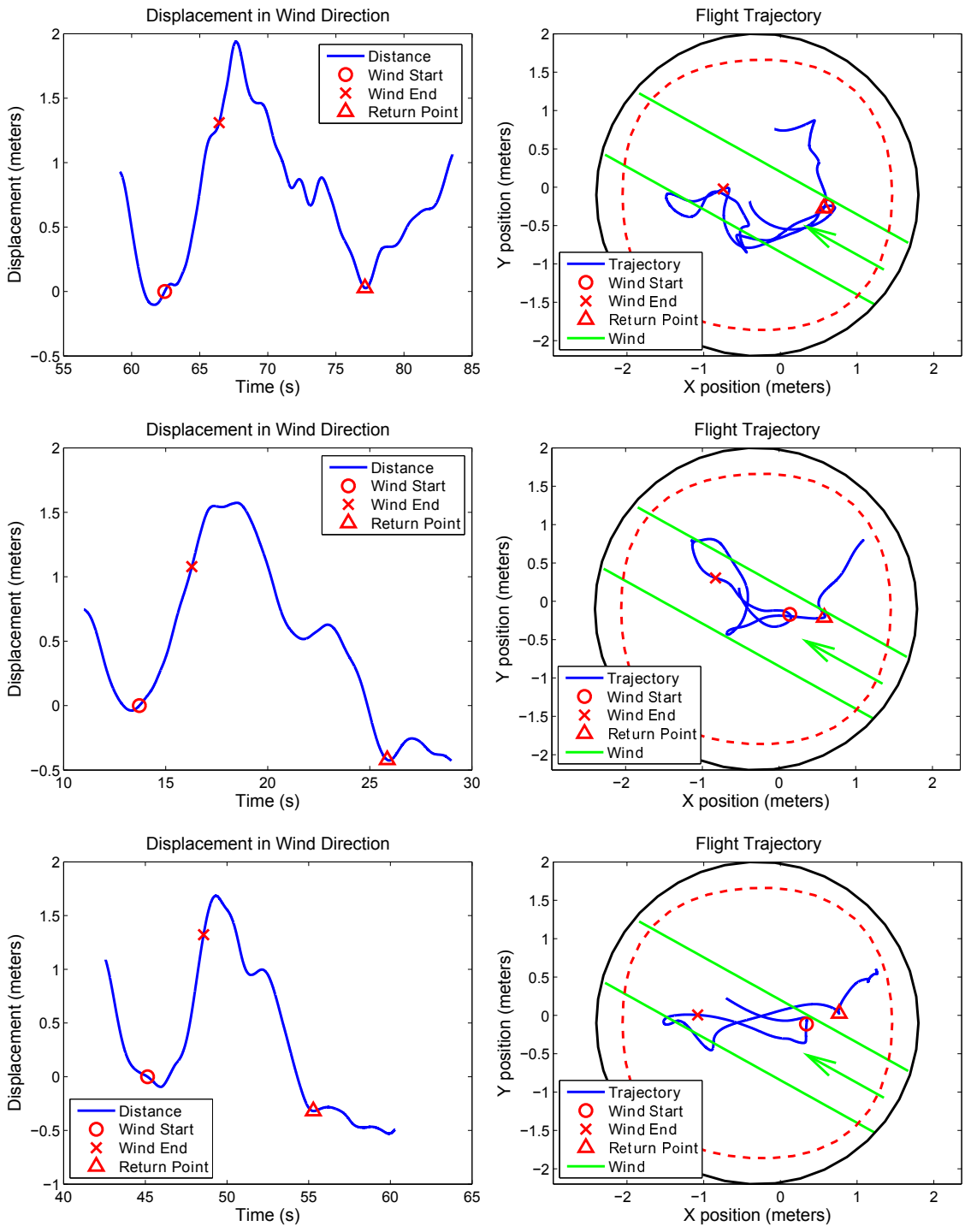


Figure 6.8: Trails 10-12: Discrete Gust Testing

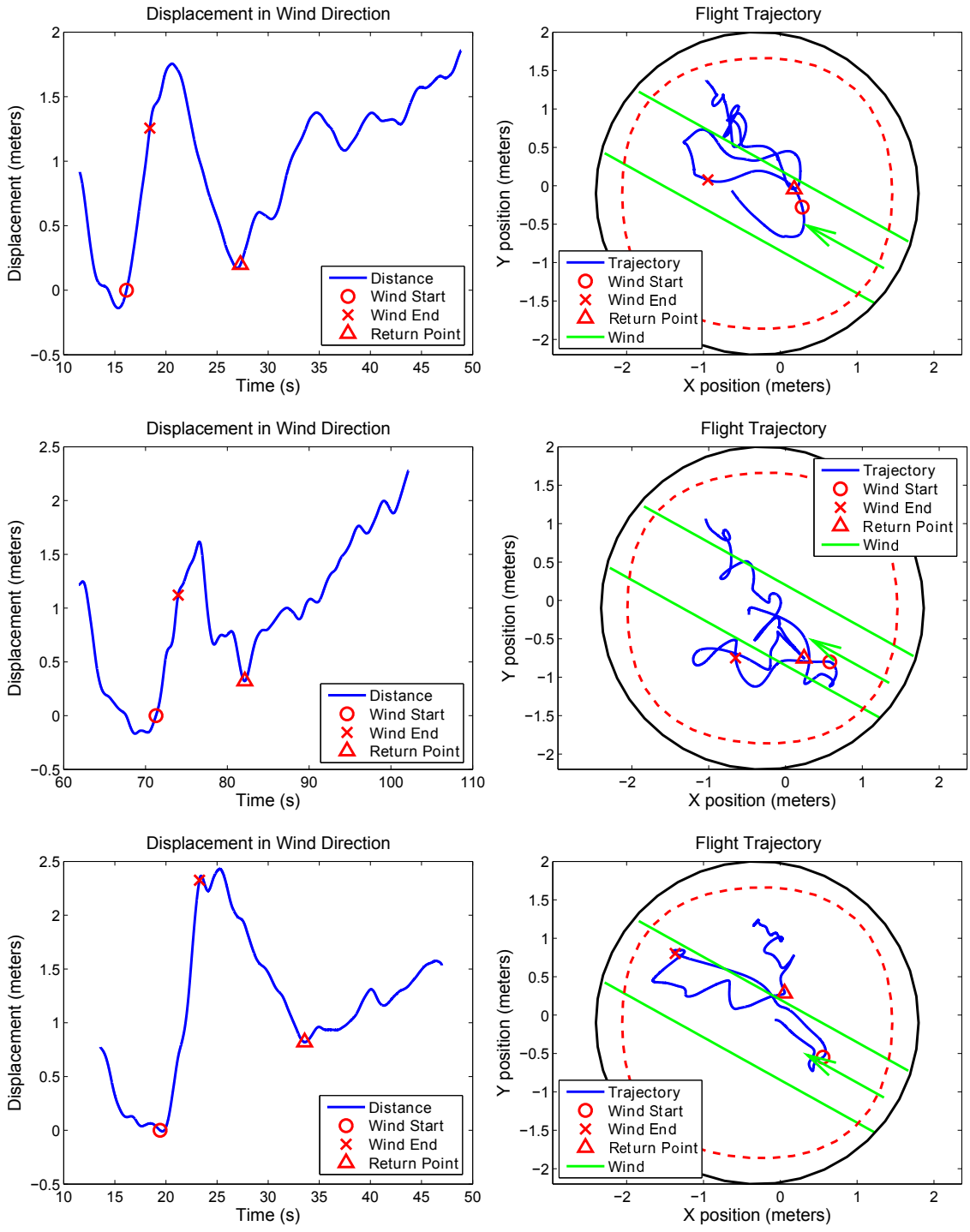


Figure 6.8: Trails 13-15: Discrete Gust Testing

Trial	Return Error	Max Error	% Error
—	meters	meters	percent
01	-0.1248	0.9887	12.6
02	0.0667	1.1360	5.8
03	0.3905	1.4380	27.7
04	0.1572	1.0140	15.5
05	-0.3298	1.0460	31.5
06	-0.0706	1.6090	4.4
07	0.3365	1.8690	18.0
08	-0.0961	1.7060	5.6
09	0.0321	1.6120	2.0
10	0.0272	1.9380	1.4
11	-0.4207	1.5740	26.7
12	-0.3197	1.6840	19.0
13	0.1981	1.7550	11.3
14	0.3211	1.6110	19.9
15	0.8200	2.4310	33.7
Max	0.8200	2.4310	33.7
Min	0.0272	0.9887	1.4
Mean	0.2474	1.5608	15.7
Median	0.1981	1.6110	15.5

Table 6.2: Statistics For Discrete Gust Trials

a percentage of the maximum error. Additionally, the minimum, maximum, mean and median are given for each quantity. From the mean values presented in Table 6.2 show the vehicle being displaced approximately 1.56 meters and returning within about a quarter meter or about 15.7% of the maximum displacement.

6.6 Sustained Gust Testing

In addition to discrete gust disturbances longer disturbances were examined. This allowed quantification of the boundedness of the vehicle position to the wind disturbance. These trials are largely identical to the discrete gusts trial with the exception that the wind was allow to remain on until the vehicle settled at a constant displacement. This resulted in gust disturbances that lasted approximately 5-10 seconds. The results for the trials in Figure 6.9 are presented in the same fashion as the discrete gust trials.

The plots shown in Figure 6.9 reveal behavior similar to the discrete gust trials. However, a few additional observations can be made. First, in the presence of constant disturbance the vehicle settles at a more or less constant displacement from the starting position, averaging about 2 meters. Furthermore, the return error con be seen to be somewhat larger that in the discrete trials. The statistics on the return and maximum errors is presented in Table 6.3. Observation of the mean values here shows both a larger return and maximum displacement than in the discrete case with the return error more than doubling while the maximum error increases by about a third. This results in a percent error of approximately 30% or

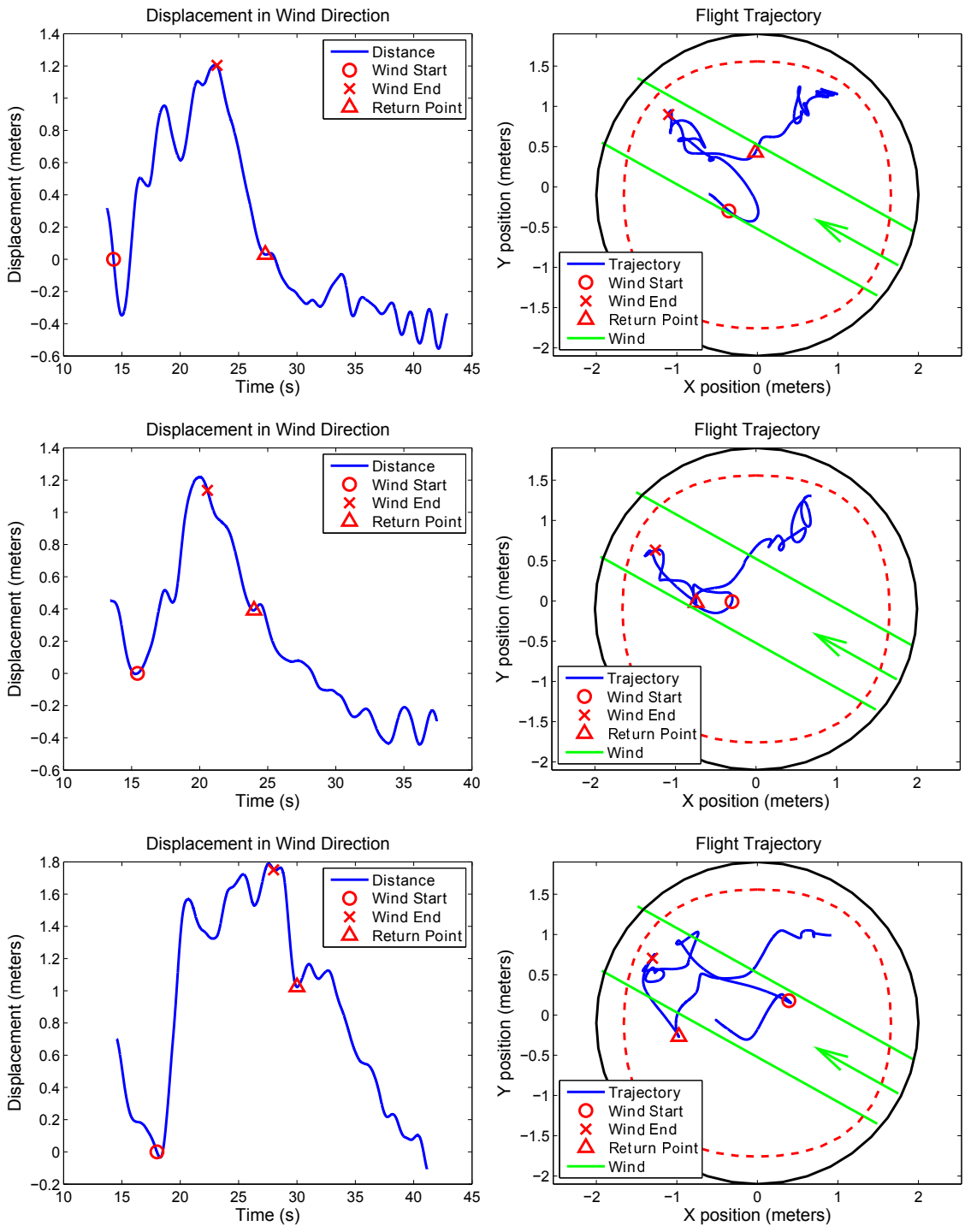


Figure 6.9: Trials 1-3: Sustained Gust Testing

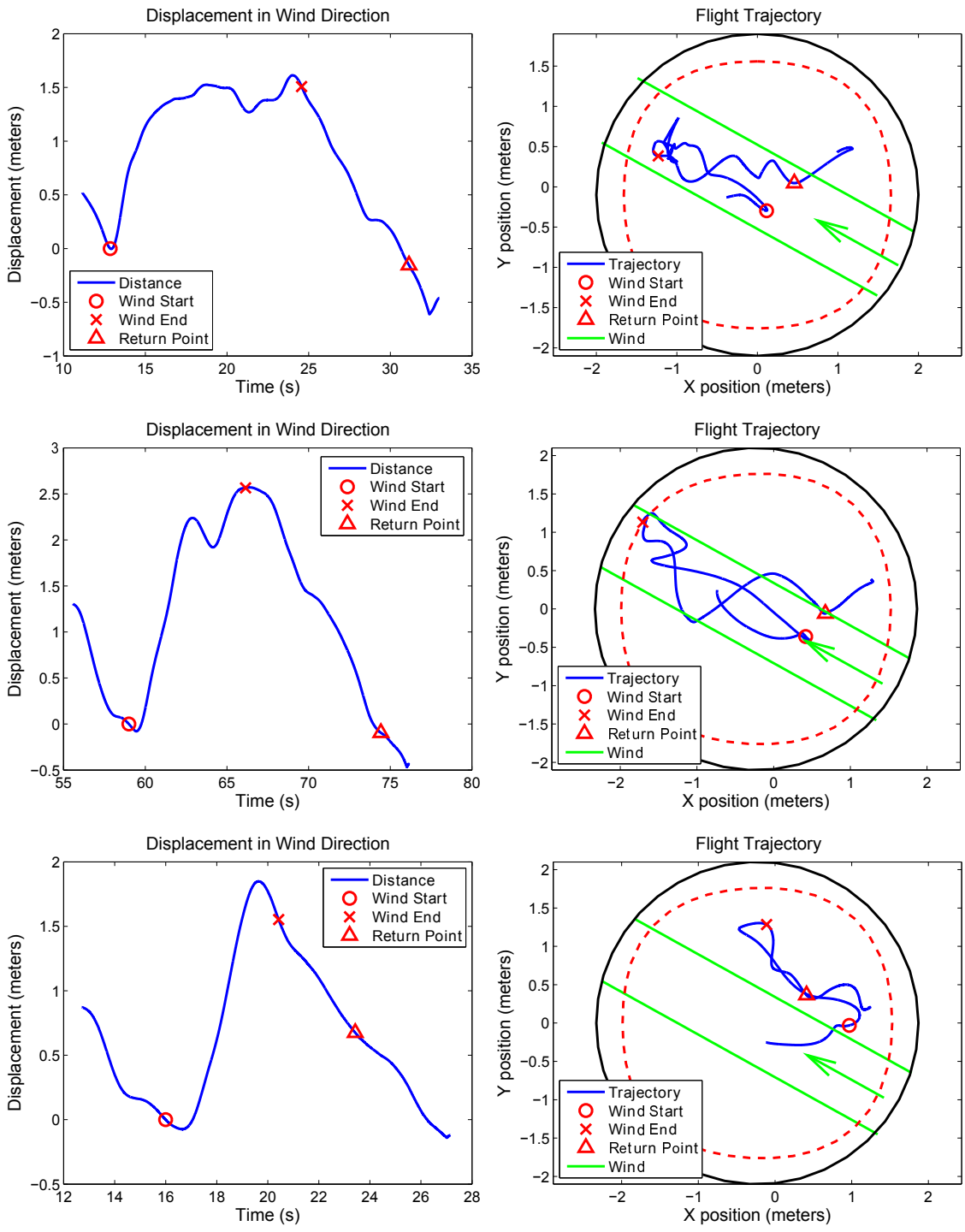


Figure 6.9: Trails 4-6: Sustained Gust Testing

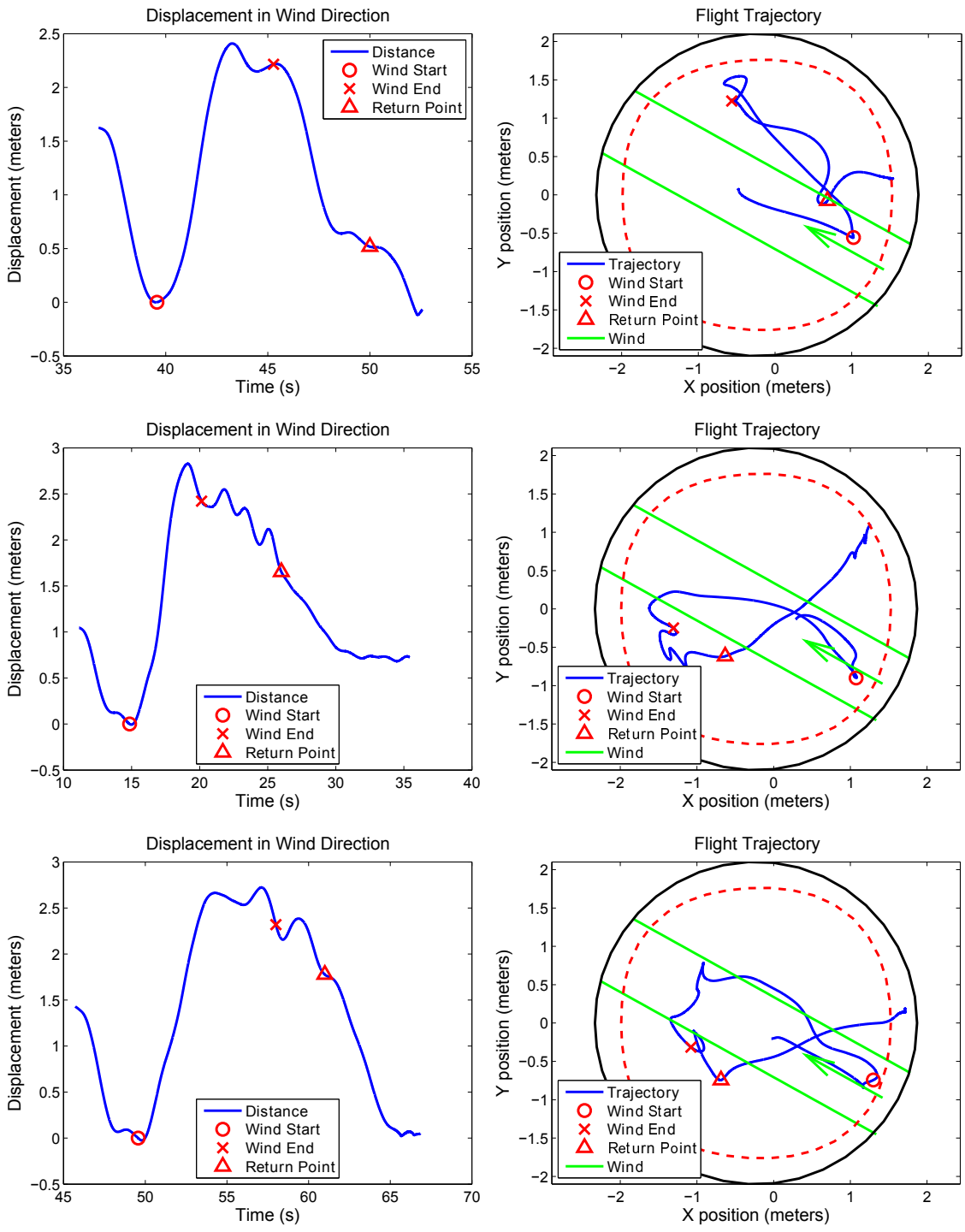


Figure 6.9: Trails 7-9: Sustained Gust Testing

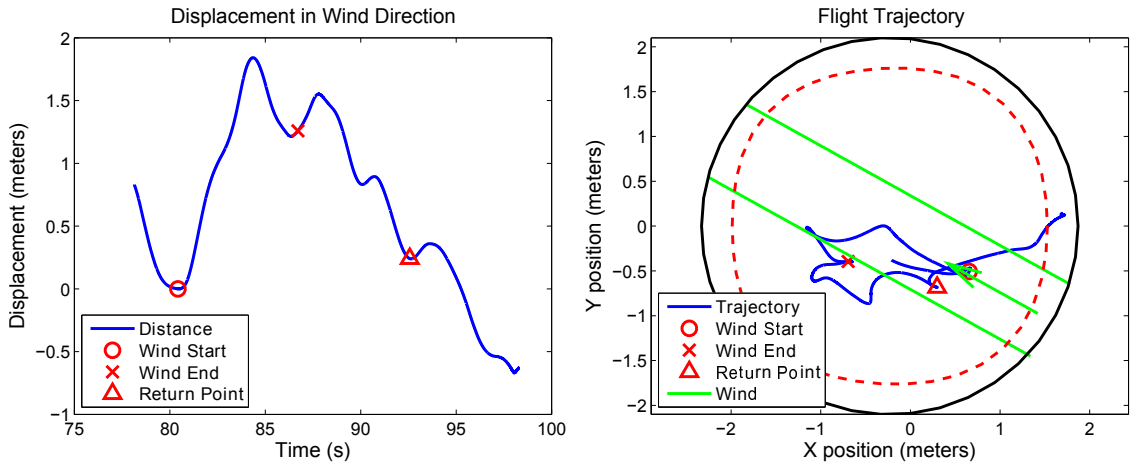


Figure 6.9: Trail 10: Sustained Gust Testing

almost twice that of the discrete case.

6.7 Comparison to Simulation

As a final step, simulations matching the conditions of the station-keeping trials were performed to understand the causes of drift. In this testing the quadrotor was started at various positions in a circular room of the same size as used in flight tests. This positions included the center of the room and two positions at 50% of the radius. Additionally, gains were matched as closely as possible to the flight testing gains. Exact matching was not possible due to the two implementations of the Gradient Method yielded optic flow outputs of differing sizes. Whether this difference is caused by differences in implementation or extraneous factors, such as image quality, was not determined. Results were collected for two gain matrix scaling factors. The first was 1/1000 which was comparable to the actual scaling between the two Gradient Method implementations. The second was 1/200 which

Trial	Return Error	Max Error	% Error
—	meters	meters	percent
01	0.0295	1.211	2.4
02	0.3917	1.218	32.2
03	1.0230	1.789	57.5
04	-0.1543	1.609	9.5
05	-0.0947	2.571	3.7
06	0.6747	1.848	36.5
07	0.5176	2.409	21.5
08	1.6540	2.827	58.5
09	1.7770	2.721	65.3
10	0.2405	1.842	13.1
Max	1.7770	2.827	65.3
Min	0.0295	1.211	2.4
Mean	0.6557	2.005	30.0
Median	0.4546	1.845	26.9

Table 6.3: Statistics For Sustained Gust Trials

was chosen to allow the noise on the optic flow signal, which was less in simulation versus flight testing, to have more effect on the vehicle. These results are presented in figures 6.10 and 6.11. Each figure shows three trials at different positions in the environment. Each trial is presented in two plots. The first shows distance from the starting position along with a linear regression to show the drift rate, as in Figure 6.7. The second the trajectory of the vehicle. Additionally, the second plot shows a inset with the position in the room displayed.

The trails shown in Figure 6.10 show both a tendency for sudden "jumps" in motion of the vehicle. This is alos seen in some of the flight tests (see Figure 6.7). Also, a tendency for an initial jump, as seen in flight tests, appears in the trails though it is small in all cases. However, with the exception on the first trial, aside from the initial jump, little drift is seen. However, in the higher gain trails shown in

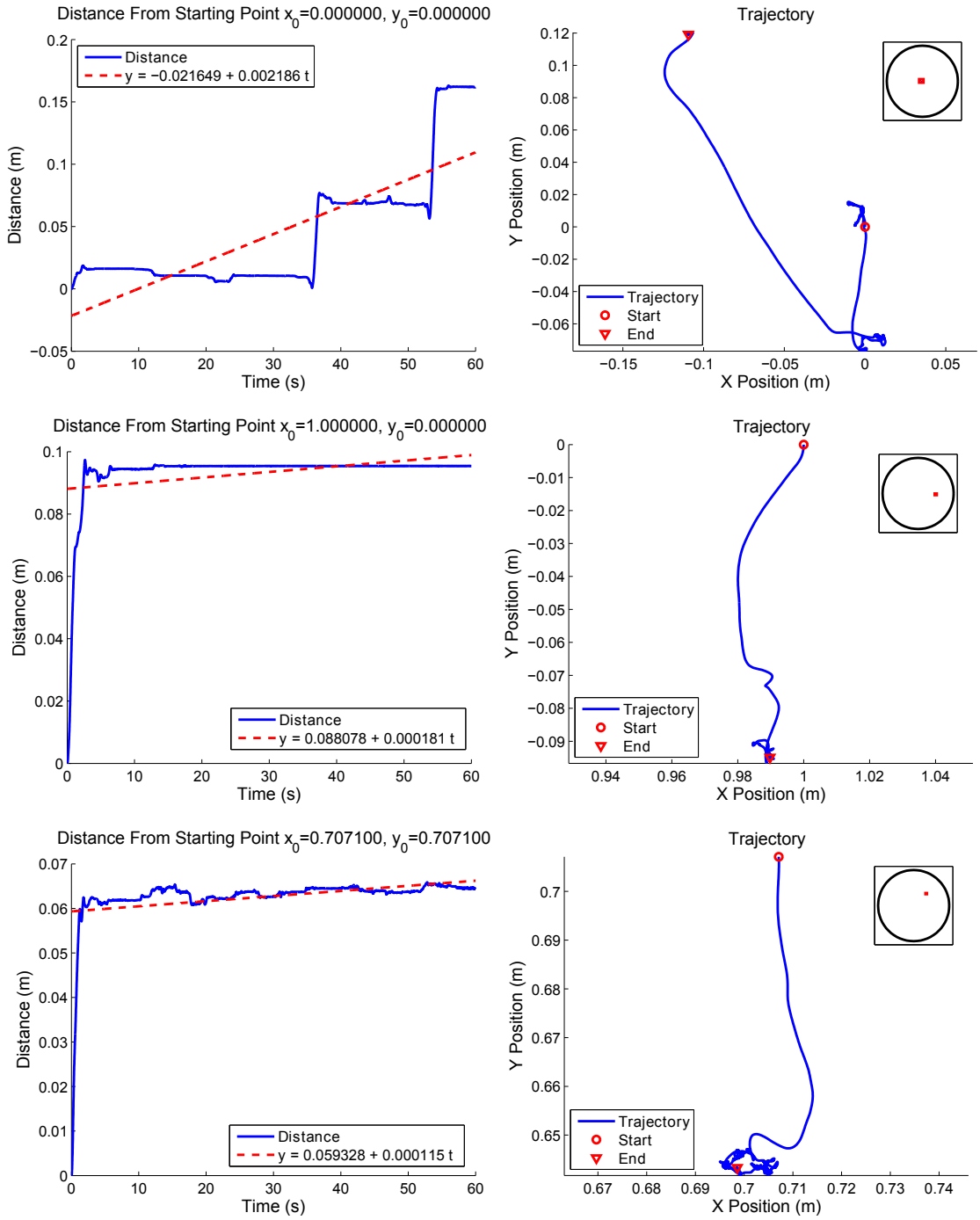


Figure 6.10: Trials for Gain Scale Factor 1/1000

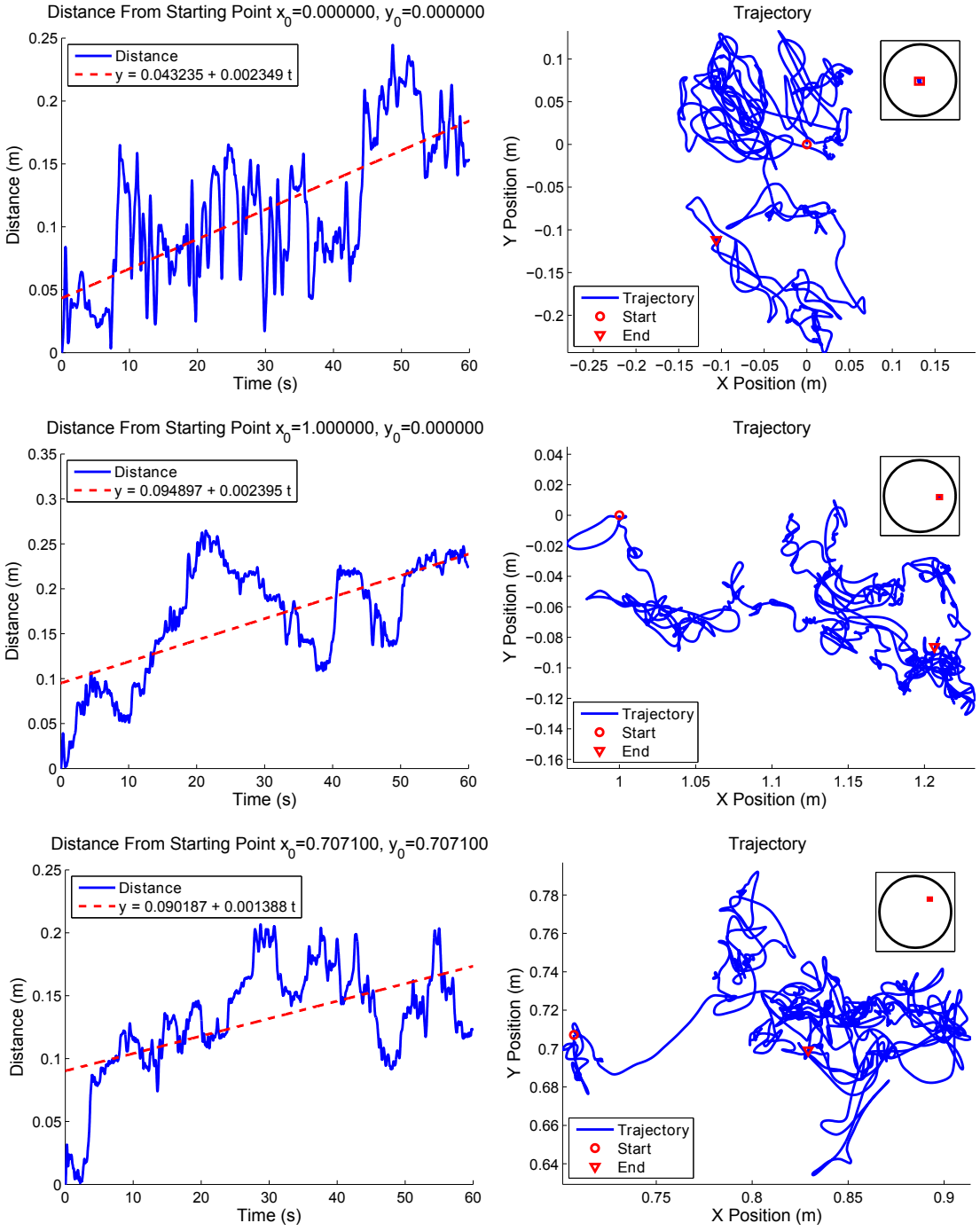


Figure 6.11: Trials for Gain Matrix Scale Factor 1/200

Figure 6.11, drift can be seen both in the form of discrete jumps as seem in the prior trials and in a more continuous fashion. These results suggest that the drift and presence of discrete jumps in motion result from the Gradient Method generated optic flow. Additionally, the initial change in position may be at least partially due to this algorithm. It should be noted, when Lucas-Kanade was simulated under these conditions no vehicle motion resulted.

6.8 Conclusions

Based on the simulation results presented in Chapter 5 it was concluded that the use of optic flow calculated using Gradient Method was likely insufficient for station-keeping purposes. Additionally, it was seen that Gradient method gave much poorer results than optic flow provided by the Lucas-Kanade algorithm. Here using the Gradient Method algorithm, the vehicle still showed a returning response when subjected to wind disturbances with a 15.7% and 30% error in the discrete and sustained gust trials. Furthermore, station-keeping trial showed and average drift rate of about 1.1 cm/sec or 90.9 seconds per meter. While, both the return error and drift results are large compared to what might be practically usable they are relatively close with only a separation of an order of magnitude or less. This fact along with the large separation in the return error showed between the two optic flow algorithms in the simulation results suggests that with a better optic flow algorithm, such as Lucas-Kanade, resonable proformance is likely achievable.

Chapter 7

Conclusions and Future Work Recommendations

As stated in Section 1.1 the goal of this study was to examine the possibility of a optic flow based controller for station-keeping and wind rejection. Towards this end theory surrounding both optic flow and wind simulation was examined. Additionally, a modified robust LQR with integral feedback was designed. Finally, the performance of this controller was testing using both simulation and flight testing.

7.1 Conclusions

Examination of the results from the testing performed leads to a number of conclusions both about the performance of this specific controller and the use of optic flow for the purposes of station-keeping and wind rejection. First, high noise optic flow algorithms such as Gradient Method are not sufficient to achieve the goals of this study. In both simulation and flight testing the Gradient Method, while sufficient to show both station-keeping and wind response behaviors, resulted in too much error to be practically usable. Second, more robust optic flow algorithms such as Lucas-Kanade are likely to show significantly better performance than simplistic methods. Based on the large differences in the simulation results between Gradient Method and Lucas-Kanade it is expected that a similar trend of improvement would be seen in hardware implementation. Third, the effect of rotational kinematics on

integration error is significant especially if large rotational wind components are present. Based on the simulation results, the larger rotational wind of the Von-Karman PSD caused a large improvement when using the correction for rotation kinematics suggesting the error introduced due to this was large. Fourth, the controller proposed in Chapter 4 is likely sufficient to achieve the goals of this study given accurate measurements. The large difference between Lucas-Kanade and Gradient Method performance suggests that noise in measurement is the limiting factor in controller performance thus with low-noise measurements the controller should exhibit adequate performance. Fifth, drift is a significant problem but will likely improve with better optic flow algorithms. Flight testing showed drift to be significant, while simulation showed significant drift only for Gradient Method thus a significant improvement is expected with better algorithms.

7.2 Limitations of the Proposed Controller

While these proposed method does offer a station-keeping methodology that is cheap relative to GPS methods and computationally inexpensive when compared to computer vision (see Chapter 1), it is not without limitations. One of the foremost is imposed by integrator drift. Since drift is present this effectively limits the duration for which the station-keeping is viable. Thus, a pilot is required to periodically reset the hold position. Furthermore, optic flow and WFI only provide relative measures of velocity and thus position, when integrated. While it is sufficient for feedback that these measures are merely proportional to actual values, actual distances are

unavailable. Additionally, the body-inertial frame correction requires an accurate measure of heading. Here perfect knowledge was assume and significant improvement was shown. However, the only way to capture heading from optic flow while in hover is integration. If the drift of this integration is large then another sensor, such as a magnetometer may be required for implementation. Finally, in the current implementation only the planar states are controlled. Thus either expansion of the controller to the full state case or having an inner-loop controller to control these states is necessary.

7.3 Recommendations for Future Work

Based on the conclusions of this research three areas are seen in which future work should proceed. First, as can be noted in many of the conclusions better optic flow algorithms are expected to show significant improvement, thus better algorithms should be examined. Results of such testing should provide insight into the quality of optic flow, and thus computation cost, required for adequate performance. Second, testing should be performed under more realistic wind conditions. As shown in simulation in chapter 5 the type of wind disturbance had a sizable effect on performance thus for thoroughness realistic wind disturbances should be examined. Third, examination of methods to reduce integration drift should be examined. This could potentially allow implementation with simpler optic flow algorithms thus reducing computation requirements.

Bibliography

[1] Karl J. Astrom and Richard M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton Univeristy Press, 2008.

[2] J.L Barron, D.J. Fleet, and S.S. Beauchemin. Performance of optic flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.

[3] A. Borst and J. Haag. Neural networks in the cockpit of the fly. *Journal of Comparative Physiology A*, 188:419–437, 2002.

[4] Arthur Earl Bryson and Yu-Chi Ho. *Applied optimal control: optimization, estimation, and control*. Hemisphere Publishing Coorperation, revised printing edition, 1975.

[5] Joseph Conroy, Gregory Gremillion, Badri Ranganathan, and J. Sean Humbert. Implementation of wide-field intergration of optic flow for autonomous quadrotor navigation. *Autonomous Robots*, 27(3):189 – 198, 2009.

[6] Joseph K. Conroy, J. Sean Humbert, and Darryl J. Pines. System identification of a rotary wing micro air vehicle. *AIAA Journal of Aircraft*, 46(4):1260–1269, 2009.

[7] D. Coombs, M. Herman, T. H. Hong, and M. Nashman. Real-time obstacle avoidance using central flow divergence, and peripheral flow. *IEEE Transactions on Robotics and Automation*, 14:49–59, 1998.

[8] Department of Defense. *Flying Qualities of Piloted Planes*, mil-hdbk-1797 edition, 1997.

[9] Martin Egelhaaf and Roland Kern. Vision in flying insects. *Current Opinion in Neurobiology*, 12:699706, 2002.

[10] Martin Egelhaaf, Roland Kern, Holger G. Krapp, Jutta Kretzberg, Rafael Kurtz, and Anne-Kathrin Warzecha. Neural encoding of behaviourally relevant visual-motion information in the fly. *TRENDS in Neurosciences*, 25(2):96–102, February 2002.

[11] J. Etele. Overview of wind gust modeling with application to autonomous low-level uav control. Technical report, Defence R& D Canada - Ottawa, 2006.

[12] Jyotirmay Gadewadikar, Frank L. Lewis, Kamesh Subbarao, Kemao Peng, and Ben M. Chen. H-infinity static output feedback control for rotorcraft. *Journal of Intelligent Robotic Systems*, 54:629–646, 2009.

[13] William E. Green, Paul Y. Oh, and Geoffrey Barrows. Flying insect inspired-vision for autonomous aerial robot maneuvers in near-earth environments. In *International Conference on Robotics and Automation*, pages 2347–2352, New Orleans, LA., April 2004. IEEE.

[14] William E. Green, Paul Y. Oh, Keith Sevcik, and Geoffrey Barrows. Autonomous landing for indoor flying robots using optic flow. In *International Mechanical Enginerring Congress*, Washington D.C., November 2003. ASME.

[15] Gregory Gremillion and J. Sean Humbert. System identification of a quadrotor micro air vehicle. In *American Control Conference*, Baltimore, MD, 2010. under review.

[16] R. Harrison. A biologically inspired analog ic for visual collision detection. *IEEE Transactions on Circuits and Systems*, 52(11):2308–2318, November 2005.

[17] Frederic M. Hoblit. *Gust Loads on Aircraft: Concepts and Applications*. AIAA Education Series. American Institute of Astronautics and Aeronautics, 370 L'Enfant Promenade, S.W., Washington D.C. 20024, 1988.

[18] Stefan Hrabar, Gaurav S. Sukhatme, Peter Corke, Kane Usher, and Jonathan Roberts. Combined optic-flow and stereo-based navigation of urban canyons for a uav. In *International Conference on Intelligent Robots and Systems*. IEEE, 2005.

[19] J. S. Humbert, A. Hyslop, and M. Chinn. Experimental validation of wide-field integration methods for autonomous navigation. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*. San Diego, CA, 2007.

[20] J. S. Humbert, R. M. Murray, and M. H. Dickinson. Pitch-altitude control and terrain following based on bio-inspired visuomotor convergence. In *Proceedings of the AIAA Guidance, Navigation and Control Conference*. San Francisco, CA, 2005.

[21] J. S. Humbert, R. M. Murray, and M. H. Dickinson. Sensorimotor convergence in visual navigation and flight control systems. In *Proceedings of the 16th IFAC World Congress*. Praha, Czech Republic, 2005.

[22] J. Sean Humbert. *Bio-inspired visuomotor convergence in navigation and flight control systems*. PhD thesis, California Institute of Technology, 2006.

[23] J. Sean Humbert and Joseph Conroy. *Flying Insects and Robots*, chapter Wide-Field Integration Methods for Visuomotor Control. Springer Verlag, 2009.

[24] J. Sean Humbert and Andrew M. Hyslop. Bio-inspired visuomotor convergence. *IEEE Transactions on Robotics*, 26(1):121–130, 2010.

[25] J. Sean Humbert, Richard M. Murray, and Michael H. Dickinson. A control-oriented analysis of bio-inspired visuomotor convergence. In *44th IEEE Conference on Decision and Control*, Seville, Spain, 2005.

[26] Andrew Hyslop. *Bio-Inspired Information Extraction In 3-D Environments Using Wide-Field Integration Of Optic Flow*. PhD thesis, University of Maryland, College Park, 2010.

[27] Andrew Hyslop and J. Sean Humbert. Autonomous navigation in 3-d urban environments using wide-field integration of optic flow. *AIAA*, 2010.

[28] Andrew Hyslop, Holger G. Krapp, and J. Sean Humbert. Control theoretic interpretation of directional motion preferences in optic flow proccessing interneurons. *Biological Cybernetics*, 2010. accepted.

[29] G. Indiveri, J. Kramer, and C. Koch. Parrellel analog vlsi architectures for computation of heading direction and time-to-contact. volume 8, Cambridge,MA, 1996. MIT Press.

[30] L. Jin, X. Xu, S. Negahdaripour, C. Tsultamoto, and J. Yuh. A real-time vision-based stationkeeping system for underwater robotics applications. In *OCEANS '96. MTS/IEEE. 'Prospects for the 21st Century'. Conference Proceedings*. IEEE, 1996.

[31] J. J. Koenderink and A. J. van Doorn. Facts on optic flow. *Biological Cybernetics*, 56:247–254, 1997.

[32] H.G. Krapp, B. Hengstenberg, and R. Hengstenberg. Dendritic structure and receptive-field organization of optic flow processing interneurons in the fly. *J. Neurophysiol.*, 79:1902–1917, 1998.

[33] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. pages 674–679, 1981.

[34] Richard L. Marks, Howard H. Wang, Michael J. Lee, and Stephen M. Rock. Automatic visual station keeping of an underwater robot. In *OCEANS '94. 'Oceans Engineering for Today's Technology and Tomorrow's Preservation.' Proceedings*, pages 137–142. IEEE, 1994.

[35] Chris McCarthy and Nick Barnes. Performance of optical flow techniques for indoor navigation with a mobile robot. In *IEEE International Conference on Robotics & Automation*, pages 5093–5098, New Orleans, LA, April 2004. IEEE.

[36] Laurent Muratet, Stephane Doncieux, Yves Briere, and Jean-Arcady Meyer. A contribution to vision-based autonomous helicopter flight in urban environments. *Elsevier Science*, 2004.

[37] S. Negahdaripour and J. Fox. Improved methods for undersea optical station-keeping. In *IEEE International Conference on Robotics and Automation*, pages 2752–2758. IEEE, 1991.

[38] Robert C. Nelson. *Flight Stability and Automatic Control*. McGraw-Hill, New York, 2nd edition, 1998.

[39] T. Netter and N. Franceschini. A robotic aircraft that follows terrain using a neuromorphic eye. In *Proceedings of the IEEE/RSJ IROS Conference on Robots and Systems*. Lausanne, Switzerland, October 2002.

[40] Michael Lee OConnor. *Carrier-Phase Differential GPS For Automatic Control Of Land Vehicles A Dissertation*. PhD thesis, Stanford University, December 1997.

[41] Scott Owen. Stocastic properties of wide field integrated optic flow measurements. Master's thesis, Univeristy of Maryland, College Park, May 2009.

[42] M. Sarcinelli-Filho, H. A. Schneebeli, E. M. O. Caldeira, and C. M. Soria. Optical flow based obstacle detection and avoidance in mobile robot navigation. In *Mediterranean Conference on Control and Automation*, Lisbon, Portugal, July 2002.

[43] R. Sarpeshkar, G. Indiveri, J. Kramer, and C. Koch. Analog vlsi architectures for motion processing: from fundamental limits to system applications. In *Proceedings of the IEEE*, volume 84, pages 969–987. IEEE, July 1996.

[44] Julien Serres, Franck Ruffier, and Nicolas Franceschini. Two optic flow regulators for speed control and obstacle avoidance. In *International Conference on Biomedical Robotics and Biomechatronics*, 2006.

[45] Eero Simoncelli, Edward Adelson, and David Heeger. Probability distributions of optic flow. In *IEEE Conference on Computer Vision and Pattern Recognition*, Mauii, HI, June 1991.

[46] Kahlouche Souhila and Achour Karim. Optical flow based robot obstacle avoidance. *International Journal of Advanced Robotic Systems*, 4(1):13–16, 2007.

[47] Steven L. Waslander and Carlos Wang. Wind disturbance estimation and rejection for quadrotor position control. In *AIAA Infotech@Aerospace Conference and AIAA Unmanned...Unlimited Conference, Seattle, Washington, Apr. 6-9, 2009*.

[48] K. Weber, S. Venkatesh, and M.V. Srinivasan. Insect inspired behaviors for the autonomous control of robots. In *Proceedings of the ICPR '96*. IEEE.

[49] Kemin Zhou. *Robust and Optimal Control*. Prentice hall, Upper Saddle River, New Jersey, 07458, 1996.