

---

Electronic Theses and Dissertations, 2004-2019

---

2016

## Geolocation of Diseased Leaves in Strawberry Orchards for a Custom-Designed Octorotor

Christian Garcia  
*University of Central Florida*



Part of the [Aerospace Engineering Commons](#)

Find similar works at: <https://stars.library.ucf.edu/etd>

University of Central Florida Libraries <http://library.ucf.edu>

This Masters Thesis (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations, 2004-2019 by an authorized administrator of STARS. For more information, please contact [STARS@ucf.edu](mailto:STARS@ucf.edu).

---

### STARS Citation

Garcia, Christian, "Geolocation of Diseased Leaves in Strawberry Orchards for a Custom-Designed Octorotor" (2016). *Electronic Theses and Dissertations, 2004-2019*. 5197.

<https://stars.library.ucf.edu/etd/5197>



University of  
Central  
Florida

STARS  
Showcase of Text, Archives, Research & Scholarship

GEOLOCATION OF DISEASED LEAVES IN STRAWBERRY ORCHARDS  
FOR A CUSTOM-DESIGNED OCTOROTOR

by

CHRISTIAN ARTHUR GARCIA

B.S.A.E. University of Central Florida, 2014

B.S.M.E. University of Central Florida, 2014

A thesis submitted in partial fulfillment of the requirements  
for the degree of Master of Science in Aerospace Engineering  
in the Department of Mechanical and Aerospace Engineering  
in the College of Engineering and Computer Science  
at the University of Central Florida  
Orlando, Florida

Summer Term  
2016

Major Professor:  
Yunjun Xu

© 2016 by CHRISTIAN ARTHUR GARCIA

## **ABSTRACT**

In recent years, technological advances have shown a strive for more automated processes in agriculture, as seen with the use of unmanned aerial vehicles (UAVs) with onboard sensors in many applications, including disease detection and yield prediction. In this thesis, an octocopter UAV is presented that was designed, built, and flight tested, with features that are custom-designed for strawberry orchard disease detection. To further automate the disease scouting operation, geolocation, or the process of determining global position coordinates of identified diseased regions based on images taken, is investigated. A Kalman filter is designed, based on a linear measurement model derived from an orthographic projection method, to estimate the target position. Simulation, as well as an ad-hoc experiment using flight data, is performed to compare this filter to the extended Kalman filter (EKF), which is based on the commonly used perspective projection method. The filter is embedded onto a CPU board for real-time use aboard the octocopter UAV, and the algorithm structure for this process is presented. In the later part of the thesis, a probabilistic data association method is used, jointly with a proposed logic-based measurement-to-target correlation method, to analyze measurements of different target sources and is incorporated into the Kalman filter. A simulation and an ad-hoc experiment, using video and flight data acquired aboard the octocopter UAV with a gimballed camera in hover flight, are performed to demonstrate the effectiveness of the algorithm and UAV platform.

## **ACKNOWLEDGMENTS**

I would like to thank my family for their constant love and support throughout my life. I would like to thank my community for always keeping me grounded, reminding me who I am and where I come from. I also want to thank Nailene for being my motivation during my time in graduate school.

I thank my faculty advisor, Dr. Yunjun Xu for his constant support, advice, and push to strive. I would also like to thank Dr. Jeffrey Kauffman and Dr. Kuo-Chi Lin for taking the time to be part of my faculty committee. Additionally, I thank the U.S. Department of Agriculture for providing the funding to conduct the research on which this thesis is based.

I would like to thank my co-workers at the UCF ARCLab for their time, feedback, and motivation. Also I thank Davis Drake for his time and piloting expertise, without whom the experiments performed for this thesis would not be possible.

# TABLE OF CONTENTS

LIST OF FIGURES .....	viii
LIST OF TABLES .....	xi
CHAPTER 1 INTRODUCTION .....	1
1.1 UAVs in Agriculture.....	1
1.2 Research Mission .....	2
1.3 Geolocation Methods .....	2
1.4 Data Association for Multi-Target Estimation .....	4
1.5 Contribution of Thesis .....	5
1.6 Thesis Outline .....	6
CHAPTER 2 OCTOROTOR UAV PLATFORM.....	8
2.1 Disease Detection Platform Specifications.....	8
CHAPTER 3 GEOLOCATION PROBLEM FORMULATION.....	12
3.1 Geolocation Overall Structure .....	12
3.2 Process Model.....	13
3.3 Orthographic Projection Based Measurement Model.....	13
3.4 Kalman Filter for Geolocation Estimation.....	19
CHAPTER 4 PERSPECTIVE PROJECTION BASED EXTENDED KALMAN FILTER FOR COMPARISON .....	21

4.1 Perspective Projection Based Measurement Model.....	21
4.2 Extended Kalman Filter .....	22
4.3 Comparison Simulation Settings.....	24
4.4 Simulation Results and Discussion.....	25
4.5 Comparison Experiment Settings .....	30
4.6 Experiment Results and Discussion.....	33
<b>CHAPTER 5 REAL-TIME IMPLEMENTATION .....</b>	<b>38</b>
5.1 Real-time Implementation Hardware Setup.....	38
5.2 Geolocation Algorithm for Real-Time Use .....	39
<b>CHAPTER 6 MULTI-TARGET GEOLOCATION WITH DATA ASSOCIATION.....</b>	<b>42</b>
6.1 Multi-Target Geolocation Problem Formulation .....	42
6.2 Multi-Target Geolocation Process Model.....	44
6.3 Multi-Target Geolocation Measurement Model .....	44
6.4 Logic-Based Measurement-to-Target Correlation Approach .....	46
6.5 Probabilistic Data Association for Each Target.....	50
6.6 Data Association in the Kalman Filter.....	52
<b>CHAPTER 7 MULTI-TARGET GEOLOCATION VALIDATION THROUGH SIMULATION</b>	
.....	54
7.1 Simulation Settings .....	54
7.2 Simulation Results and Discussion.....	55

CHAPTER 8 MULTI-TARGET GEOLOCATION EXPERIMENT .....	62
8.1 Experiment Settings .....	62
8.2 Experiment Results and Discussion.....	64
CHAPTER 9 CONCLUSION.....	71
9.1 Concluding Remarks.....	71
9.2 Future Work .....	72
LIST OF REFERENCES .....	73



## LIST OF FIGURES

Figure 2.1.1 Octorotor flight avionics layout diagram .....	10
Figure 2.1.2 Octorotor platform.....	11
Figure 3.3.1 Earth-centered-inertial and local inertial Earth model .....	14
Figure 3.3.2 Gimbal model.....	16
Figure 3.3.3 Orthographic projection model.....	17
Figure 4.4.1 Actual simulated target position and position estimate for both KF and EKF in both directions.....	26
Figure 4.4.2 Simulated and predicted measurements for the perspective projection with EKF and orthographic projection with KF, in the $x$ direction.....	28
Figure 4.4.3 Simulated and predicted measurements for the perspective projection with EKF and for the orthographic projection with KF, in the $y$ direction .....	29
Figure 4.4.4 Position estimation covariance trace for the KF and EKF, respectively .....	30
Figure 4.5.1 Screenshot of the video taken from the octorotor UAV of the red box in a grassy field .....	31
Figure 4.6.1 Octorotor position in all three directions, respectively .....	33
Figure 4.6.2 Actual and predicted measurements, for both the geolocation methods, respectively, in the $x$ direction.....	34
Figure 4.6.3 Actual and predicted measurements, for both the geolocation methods, respectively, in the $y$ direction.....	35

Figure 4.6.4 Estimated position for both geolocation methods, along with the actual position and octorotor position, in the $x$ direction .....	36
Figure 4.6.5 Trace of the covariance for both filtering methods, respectively .....	37
Figure 5.1.1 Information flow diagram for the real-time implementation .....	39
Figure 5.2.1 Real-time geolocation algorithm structure components and objects.....	41
Figure 6.1.1 Multi-target geolocation algorithm overall structure .....	43
Figure 6.3.1 Orthographic projection diagram for multiple targets .....	45
Figure 6.4.1 Logic-based approach to measurement-to-target correlation flow chart.....	50
Figure 7.2.1 Simulated octorotor motion along with the moving average filter in each direction	55
Figure 7.2.2 Measurement correlation and association performed in an image frame.....	56
Figure 7.2.3 Actual and estimated output in the $x$ direction for each target .....	57
Figure 7.2.4 Actual and estimated output in the $y$ direction for each target .....	58
Figure 7.2.5 Actual and estimated position in the $x$ direction for each target .....	58
Figure 7.2.6 Actual and estimated position in the $y$ direction for each target .....	59
Figure 7.2.7 Trace of the estimate covariance matrix for each target.....	60
Figure 8.1.1 Video frame of the targets in the field.....	62
Figure 8.2.1 Octorotor position in all three directions, respectively .....	64
Figure 8.2.2 Image frame of the flight video with correlation and data association approaches applied.....	65
Figure 8.2.3 Predicted measurements in the $x$ direction for each target .....	66

Figure 8.2.4 Predicted measurements in the $y$ direction for each target .....	67
Figure 8.2.5 Actual and estimated position in the $x$ direction for each target, respectively .....	68
Figure 8.2.6 Actual and estimated position in the $y$ direction for each target, respectively .....	69
Figure 8.2.7 Estimate covariance trace for each target, respectively.....	69

## LIST OF TABLES

Table 2.1.1 Octorotor overall specifications.....	8
Table 2.1.2 Motor performance specifications .....	9
Table 4.4.1 Simulation results comparing KF and EKF performance.....	27
Table 4.6.1 Experiment results comparing KF to EKF performance .....	36
Table 7.2.1 Simulation results for actual and estimated target position .....	61
Table 8.2.1 Experiment results for actual and estimated target positions .....	70

# CHAPTER 1 INTRODUCTION

## 1.1 UAVs in Agriculture

Autonomous robotics is a growing technological trend that is finding many applications in agriculture [1], where reductions in labor time, human error, and costs, as well as an increase in overall production due to these robots has been seen. As one kind of autonomous robot, unmanned aerial vehicles (UAVs) have been widely used agricultural applications, such as pesticide application [2], crop yield estimation [3], and invasive species identification [4]. One of the newer agricultural applications of UAVs is disease detection through remote sensing [5], where UAVs are carrying sensing equipment in order to obtain higher resolution images when compared to satellite imagery [6][7], as well as for their quick turnaround time, ability to hover, and low cost [8][9]. This process can aid to quantifiably indicate disease in crops [8] and, in some cases, at early stages before symptoms are visible [10]. Once disease has been detected, the images containing suspected regions are mapped to their physical location, through georeferencing methods [3][9][11]. To further advance the disease detection process, the suspected disease position can be localized directly from the images, which cannot currently be done through georeferencing. For this, the geolocation problem is investigated [12]-[16], which is essentially a filtering problem to estimate the position of a target.

## **1.2 Research Mission**

Our research mission, at the UCF ARCLab, is to develop an autonomous robot network, consisting of a UAV and an unmanned ground vehicle (UGV), that can detect diseases in strawberry orchards using spectral technology, without human input [5]. The UAV will rapidly scan the strawberry field autonomously, with a spectral camera on-board, taking continuous spectral images of the trees below. The plant leaves in the images will be analyzed for abnormalities, either in water stress, radiation levels, or other readings, depending on the typically known disease symptoms. The spectral camera associates the intensities being analyzed to the pixelated image of the field where such a suspected region lies, allowing the pixel coordinates of the suspected region to be extracted. These pixel measurements are then used to geolocate the suspected regions in real space, through the work presented in this thesis. The suspected diseased leaf positions are then converted into GPS coordinates and sent through radio telemetry to the UGV. The UGV then traverses the corresponding strawberry row autonomously, based on the given GPS coordinates. The ground robot arrives at each suspected plant, further analyzes the plant to verify the disease type, collects a leaf sample with a manipulator arm (if the disease type is unknown), and returns the sample to the users.

## **1.3 Geolocation Methods**

Geolocation methods generally differ based on the sensors used (e.g. laser range-finders [13], electro-optical infrared sensors [14], or gimballed cameras [15] [16]) and on the model derived. Since image-based sensors are widely used in agricultural disease detection applications, only the

geolocation methods based on this type of sensor are reviewed here. Campbell and Wheeler perform geolocation using a gimbaled camera on an UAV to locate moving vehicles in [15], which accounts for a moving target in its dynamic model, derives a perspective projection based measurement model, and applies a square root implementation of a sigma point filter for a decentralized target localization approach, where the number UAV states being estimated is reduced. Sohn et al [16] uses a single-antenna GPS and gimbaled camera to track a stationary target from a fixed-wing UAV, and a recursive averaging filter is used based on a normalized line of sight (LOS) and triangulation approach. While this approach relies on only two sensors for target localization, it requires vehicle Euler angle estimation without direct angle measurements that causes a bias in the target localization algorithm, which must be compensated for.

Both of these approaches rely on nonlinear projection models which can encounter issues when the target distance from the sensor is not known. Additionally, mostly all nonlinear projection models used rely on nonlinear filtering designs, which are not optimal and generally have high computational cost. Here, the depth issue is mitigated by two assumptions. First, by using a face-down gimbal configuration [12], the flight altitude can be considered as the target distance. Second, small agricultural crops, such as strawberry plants, can be considered shallow objects when seen from aerial images because of the small depth compared to the flight altitude, where estimation in two dimensions is valid [17][18]. With these two assumptions, the orthographic projection model [17] can be used, which is linear in the states. In turn, a linear Kalman filter can be designed to estimate the position of the suspected disease area. The Kalman filter is chosen over other filtering methods because it is optimal in minimizing the covariance trace of the estimate error for linear systems [19].

## 1.4 Data Association for Multi-Target Estimation

Since aerial images are generally acquired at high altitudes for disease detection, it is likely that multiple diseased regions of plants will appear in each image. Multi-target estimation and tracking has been widely studied through the use of data association techniques in many applications (e.g. naval ship tracking [20], missile, tracking [20] [23], space debris tracking [24]). Generally, there are two data association approaches: Bayesian and non-Bayesian [20]. The Nearest Neighbor approach is a non-Bayesian approach which considers the nearest measurement to the predicted measurement for updating the filter, which is not computationally intensive but doesn't consider whether the nearest measurement is a correct one [21] [22]. Other non-Bayesian approaches, like the Track-Splitting approach and the Maximum Likelihood method, are based on evaluating the likelihood function and also assume that the accepted track is the correct one [21].

Bayesian approaches, like the Probabilistic Data Association Filter (PDAF), associate to each measurement a probability of being correct based on its innovation from the predicted measurement [20][23]. The Optimal Bayesian approach is similar to the PDAF except that it considers all measurements encountered in all previous time for a fully optimal estimation whereas the PDAF only uses the current available measurements [20][21]. While this filter performs better than the PDAF, its associated computational cost grows with time [21]. Li et al uses a PDAF for trajectory tracking of space debris through visual measurements in a cluttered environment [24]. The PDAF uses a validation gate to only consider measurements of statistical importance to the trajectory, which is contradictory in the case of multiple targets where measurements should not be discarded. Hence the Joint Probabilistic Data Association Filter (JPDAF) was proposed in [20], which uses a single validation region for all measurements, determines the measurement-to-target



correlation, and calculates the joint probability for all measurements (in order to maintain a uniform false measurement probability for all targets). This can, however, cause heavy computational burdens [25]. Additionally, the PDAF and JPDAF assume that target trajectories have already been initialized [21]. A logic-based multi-target initiator is presented in [20] for a target-based approach to multi-target estimation, as opposed to the JPDAF which is measurement-based.

## 1.5 Contribution of Thesis

In this thesis, a geolocation algorithm is developed based on pixelated measurements from a gimbaled camera in face-down configuration, and UAV GPS values, to be used to determine the location of multiple suspected disease regions directly from the images. A linear measurement model is derived based on the shallow structures assumption, appropriate for small agricultural crops. The PDAF technique is adopted for associating measurements because of its computational simplicity and better performance when compared to the non-Bayesian methods. A logic based method, considering topics used in the JPDAF of [20], is proposed for trajectory initialization (when a measurement is outside a target validation region), trajectory merger (when a measurement corresponds to two trajectories), and trajectory termination (when no measurements reside in that target validation region). This was chosen in conjunction with the PDAF for multi-target trajectory estimation over the JPDAF or the Optimal Bayesian approach to avoid the computational burdens of considering all measurements for all targets and determining their joint probabilities [20], since computation requirements can be of issue aboard the UAV. The addition of geolocation to the disease detection process allows the suspected regions to be directly localized

autonomously in the initial scouting phase. This crucial step will also bridge the information gap between the tasks of the UAV in the initial scouting phase and the verification and recollection tasks of the UGV, by providing the UGV with position coordinates for its portion of the mission, directly from the scouting results of the UAV. Additionally, the designed Kalman filter geolocation approach is based on conditions pertinent to small crops and is less computationally intensive than the typically used Extended Kalman Filter, based on the perspective projection. To the best of the authors' knowledge, it is the first time a data association technique is used in for locating multiple diseased plants in agriculture, allowing for simultaneous geolocation of multiple diseased regions autonomously from on-board the UAV.

## **1.6 Thesis Outline**

In CHAPTER 2, the custom-designed disease detection octorotor UAV platform is presented, along with its performance specifications and design requirements. The geolocation problem is formulated and the proposed method is derived in CHAPTER 3. CHAPTER 4 consists of validating the proposed geolocation method through simulation and an ad-hoc experiment, by comparing it to a nonlinear, extended Kalman filter approach based on the commonly used perspective projection model. The real-time implementation of the geolocation method on a CPU on-board the octorotor is presented as CHAPTER 5. In CHAPTER 6, the multi-target geolocation problem is defined, a logic-based approach for measurement-to-target correlation is proposed, the probabilistic data association technique used is shown, and both these additions are incorporated into the Kalman filter appropriately. Simulation results of the multi-target geolocation method are shown in CHAPTER 7, and an ad-hoc experiment using real flight data and images is discussed in

CHAPTER 8, both to validate the multi-target geolocation approach. Conclusions are given in CHAPTER 9.

## CHAPTER 2 OCTOROTOR UAV PLATFORM

### 2.1 Disease Detection Platform Specifications

Rapid disease scouting and detection in strawberry orchards is to survey the field with an aerial platform equipped with a disease detection pixelated sensor, acquire images of the field, determine abnormal regions in the images, run the proposed geolocation algorithm to pinpoint coordinates for said regions, and send these coordinates back to the users. Considering a typical strawberry field size, a platform is designed to endure a 20-minute flight time with at least 5 pounds of payload, with the specifications listed in Table 2.1.1 below. In order to acquire a series of images about detected regions of suspected disease, the platform must be able to hover over an area.

Table 2.1.1 Octorotor overall specifications

Full Wingspan, <i>m</i>	1.5
Vehicle weight, <i>kg</i>	8.9
Landing area, <i>m</i> <sup>2</sup>	2.5
Height, <i>m</i>	0.73
Payload weight, <i>kg</i>	3.3
Total weight, <i>kg</i>	12.3 (26.9 <i>lb</i> )
Estimated flight time, <i>min</i>	22.6
Actual flight time, <i>min</i>	> 18
Maximum current, <i>A</i>	154.4

Taking the above two requirements into consideration, an octorotor is designed with a light-weight Tarot™ carbon fiber frame, eight 400 KV rated brushless Navigation Series T-motors™, and 15x5 inch carbon fiber propellers. The vehicle is powered by four 22.2 V Turnigy Nano-tech™ Li-Po batteries, supplying 8,000 mAh each, in parallel for a total current capacity of 32 Ah. The eight-rotor configuration is chosen to increase the individual motor efficiency by reducing the load experienced by each motor, shown in Table 2.1.2. Based on the octorotor weight, hover is performed at about 67%-70% throttle.

Table 2.1.2 Motor performance specifications

Throttle, %	Current required, A	Power required, W	Thrust, kg	Angular velocity, RPM	Efficiency, g/W
50	4.8	105.6	1.04	4,385	9.8
65	7.4	164.3	1.46	5,190	8.9
75	11.8	262	1.97	6,015	7.5
85	15.2	337.5	2.33	6,545	6.9
100	19.3	428.5	2.71	7,060	6.3

The octorotor flight maneuvers are governed by the Copter APM 2.6™ flight controller, which commands eight Turnigy Plush™ series, 40 A electronic speed controllers (ESCs), that regulate power to the brushless motors. The ESCs, in turn provide appropriate power to the APM 2.6 board, as shown in the avionics diagram of Figure 2.1.1. This flight controller also has autonomous GPS based waypoint navigation functionality. The GPS module aboard the octorotor is a uBlox LEA-6

with 2.5 m-4 m circular error probability. Latitude, longitude, and altitude values from the GPS are logged by the flight controller for later use. An RGB camera is used to acquire the images during experiments.

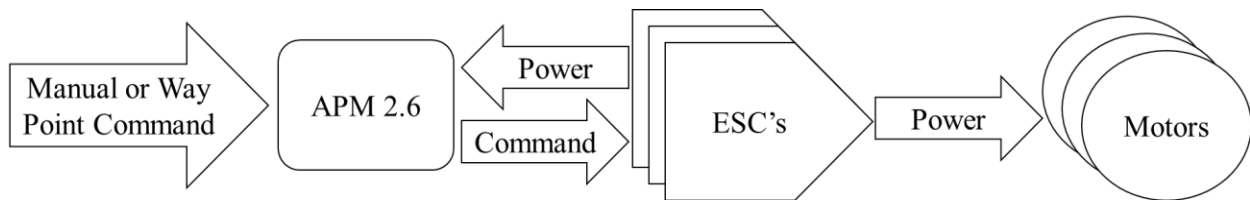


Figure 2.1.1 Octorotor flight avionics layout diagram

A custom casing is designed and built to protect the sensor onboard via a strong fiberglass structure and automated doors, seen in Figure 2.1.2 below. This casing uses a range finder to detect proximity to the ground and actuates the doors to shield the lens from any dust or particles. A robust landing gear is also designed in-house to be low cost and to accommodate for the overall weight of the octorotor and the additional height required by the gimbaled protection casing. A Legacy DSLR 3-axis aerial brushless gimbal™ is installed, which uses two accelerometers to counteract the rotational motion of the octorotor in flight. The gimbal is placed in a face-down configuration, which is necessary for the studied orthographic projection method.



Figure 2.1.2 Octorotor platform

## **CHAPTER 3**

### **GEOLOCATION PROBLEM FORMULATION**

#### **3.1 Geolocation Overall Structure**

The geolocation problem in general is a state estimation problem with pixelated measurements that are modeled as functions of the position vector components of the plant. It is assumed that the pixel measurements of the suspected regions, detected based on the intended identification scheme, are provided already. As an optional step, nearby pixels can be joined into contour regions and the region centroids can be taken as the pixel measurements. Once so, the geolocation problem begins and this proposed method can be applied to estimate the position trajectories for each target.

The target plants are stationary at their roots and move only according to the winds that disturb them, reducing the position estimation to that of a stationary target. Therefore, any large scale, small frequency movement in the predicted measurement trajectories of the filter can be attributed to movement of the UAV in hover. For an orthographic projection, the sight vector of the camera will need to be perpendicular to the ground plane, to minimize any errors that can be introduced by an inclined image plane [12]. This assumption is valid for the hardware setup because of the ability of the gimbal aboard to counteract the octorotor attitude motion while maintaining the camera face-down at all times. Additionally, this allows the altitude of the UAV, which is directly measured, to be used to deal with the depth issue encountered when modeling from a 2-D image to the 3-D real space. Furthermore, it is assumed that the height of each plant is constant and minuscule compared to the flight altitude. This combination allows for the depth of the plant in the image to be ignored and reduces the estimation problem to two dimensions, for the  $x$  and  $y$  components of position. A position error bound of  $0.5m$  or less in either direction is desired, in



order to avoid possibly misleading the estimation onto an incorrect orchard row, which are generally spaced over twice that distance apart.

### 3.2 Process Model

Each target plant is stationary, meaning that the state dynamics can be modeled as zero, plus a process noise term, as shown in

$$\dot{\mathbf{x}}^L = \begin{pmatrix} \dot{x}^L \\ \dot{y}^L \end{pmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \mathbf{w} \quad (3.2.1)$$

This process noise accounts for disturbances (e.g. wind gusts) and is assumed to be zero-mean, Gaussian white noise with a variance matrix  $Q$  as

$$\mathbf{w} \sim N(0, Q) \quad (3.2.2)$$

### 3.3 Orthographic Projection Based Measurement Model

The relation between the measured pixels in the image and the target position being estimated, needs to be derived. The octorotor position, given in latitude, longitude, and altitude, are received from the GPS and barometer onboard. These measurements are used to determine the octorotor position  $\mathbf{x}_o^L$  as

$$\mathbf{x}_o^L = \begin{bmatrix} -S(lon_o) & C(lon_o) & 0 \\ -S(lat_o)C(lon_o) & -S(lat_o)S(lon_o) & C(lat_o) \\ C(lat_o)C(lon_o) & C(lat_o)S(lon_o) & S(lat_o) \end{bmatrix} \left\{ \begin{bmatrix} (alt_o + R_E)C(lat_o)C(lon_o) \\ (alt_o + R_E)C(lat_o)S(lon_o) \\ (alt_o + R_E)S(lat_o) \end{bmatrix} - \begin{bmatrix} (alt_h + R_E)C(lat_h)C(lon_h) \\ (alt_h + R_E)C(lat_h)S(lon_h) \\ (alt_h + R_E)S(lat_h) \end{bmatrix} \right\} \quad (3.3.1)$$

where the superscript  $L$  denotes a local inertial frame, as seen in Figure 3.3.1 below. The octorotor position is determined with respect to the chosen home position, denoted by subscript  $h$ . The terms  $alt$ ,  $lat$ , and  $lon$ , denote the altitude, latitude, and longitude respectively and are taken at both the home position and the octorotor's current position. The term on the right of the expression represents the translation relative to the home position and the term on the left represents the rotation from Earth Centered Inertial frame,  $ECI$ , to the local inertial frame,  $L$ .  $R_E$  is the radius of the Earth at the current latitude and  $C(\cdot)$  and  $S(\cdot)$  denote the cosine and sine functions respectively.

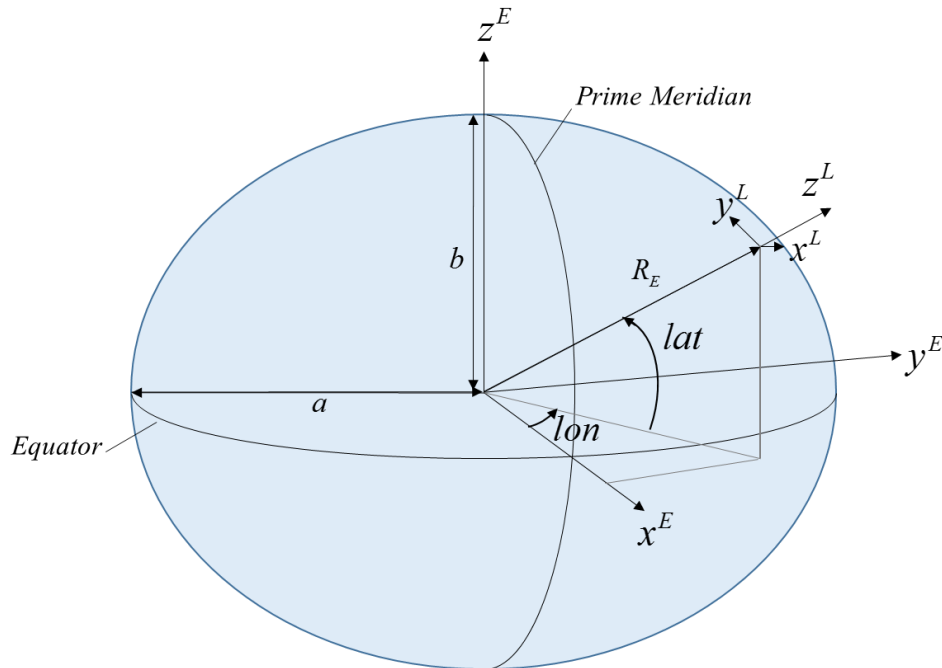


Figure 3.3.1 Earth-centered-inertial and local inertial Earth model

The rotation matrix from the local inertial frame  $L$  to the octorotor body frame  $B$  is given as

$$R_L^B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C(\phi) & S(\phi) \\ 0 & -S(\phi) & C(\phi) \end{bmatrix} \begin{bmatrix} C(\theta) & 0 & -S(\theta) \\ 0 & 1 & 0 \\ S(\theta) & 0 & C(\theta) \end{bmatrix} \begin{bmatrix} C(\psi) & S(\psi) & 0 \\ -S(\psi) & C(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (3.3.2)$$

The last matrix in this rotation, places the vector from an East-North-Up orientation into a North-East-Down (NED) one, commonly used in aircraft modeling [26]. The angles  $\phi$ ,  $\theta$ , and  $\psi$  are the Euler angles for the roll, pitch, and yaw rotations respectively. An additional rotation matrix is required between the octorotor frame and the camera frame, denoted by superscript  $C$ . Based on the gimbal structure, a 3-1-2 rotation sequence was used,

$$R_B^C = R_\beta R_\alpha R_\gamma = \begin{bmatrix} C(\beta) & 0 & S(\beta) \\ 0 & 1 & 0 \\ -S(\beta) & 0 & C(\beta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & C(\alpha) & S(\alpha) \\ 0 & -S(\alpha) & C(\alpha) \end{bmatrix} \begin{bmatrix} C(\gamma) & S(\gamma) & 0 \\ -S(\gamma) & C(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.3.3)$$

The corresponding Euler angles for the gimbal were labeled  $\alpha$ ,  $\beta$ , and  $\gamma$  for its roll, pitch, and yaw respectively, and the gimbal pitch angle  $\beta$  is defined at 90 degrees when facing the ground. The camera position vector in the local frame is expressed as

$$\mathbf{x}_c^L = \mathbf{x}_o^L + (R_L^B)^T \left( R_\gamma^T R_\alpha^T R_\beta^T \mathbf{x}_{3 \rightarrow c}^C + R_\gamma^T R_\alpha^T \mathbf{x}_{2 \rightarrow 3}^{(2)} + R_\gamma^T \mathbf{x}_{1 \rightarrow 2}^{(1)} + \mathbf{x}_{b \rightarrow 1}^B + \mathbf{x}_{o \rightarrow b}^B \right) \quad (3.3.4)$$

where the numbers 1, 2, 3 denote the intermediate frames at each joint of the gimbal in the order they are encountered, shown in Figure 3.3.2 below. The superscript  $T$  describes the transpose of the matrix and lowercase  $b$  is in reference to the base of the gimbal. Note that each relative vector, from one intermediate point to another, must be rotated back to the octorotor body frame  $B$  by a

respective rotation matrix before being superimposed. The final addition results in the position vector camera with respect to the octorotor in the body frame  $B$ , which is then rotated to the inertial frame  $L$  and added to the octorotor position vector, resulting in the camera position vector in the inertial frame.

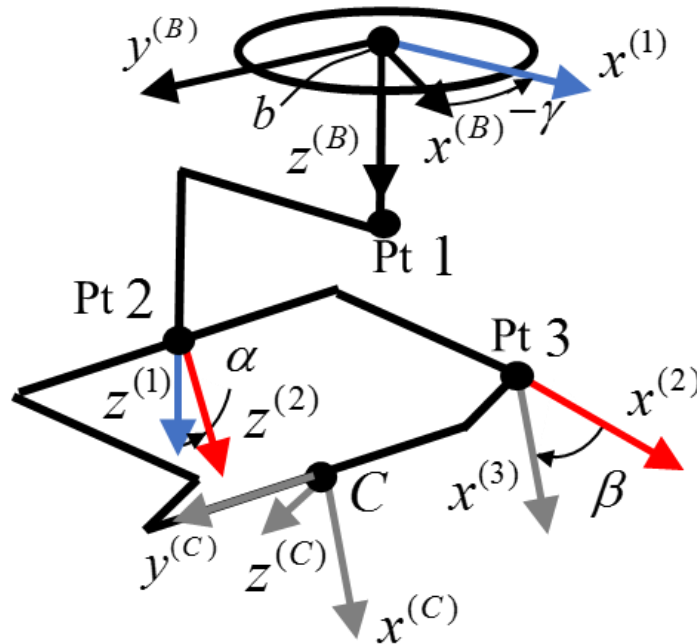


Figure 3.3.2 Gimbal model

For the orthographic projection model, shown in Figure 3.3.3, the vector from the camera origin to each target in the image plane, and the vector from the camera origin to each target in real space must lie along the same line [12][15]. Therefore, the former vector can be regarded as a scaled version of the latter. The latter can be expressed as the relative vector between the targets and the camera, as viewed in the camera frame,

$$\mathbf{x}_{rel}^C = R_L^C (\mathbf{x}^L - \mathbf{x}_c^L), \quad (3.3.5)$$

where  $\mathbf{x}^L$  is the unknown position vector of the target plant and the subscript *rel* denotes that it is the relative vector.

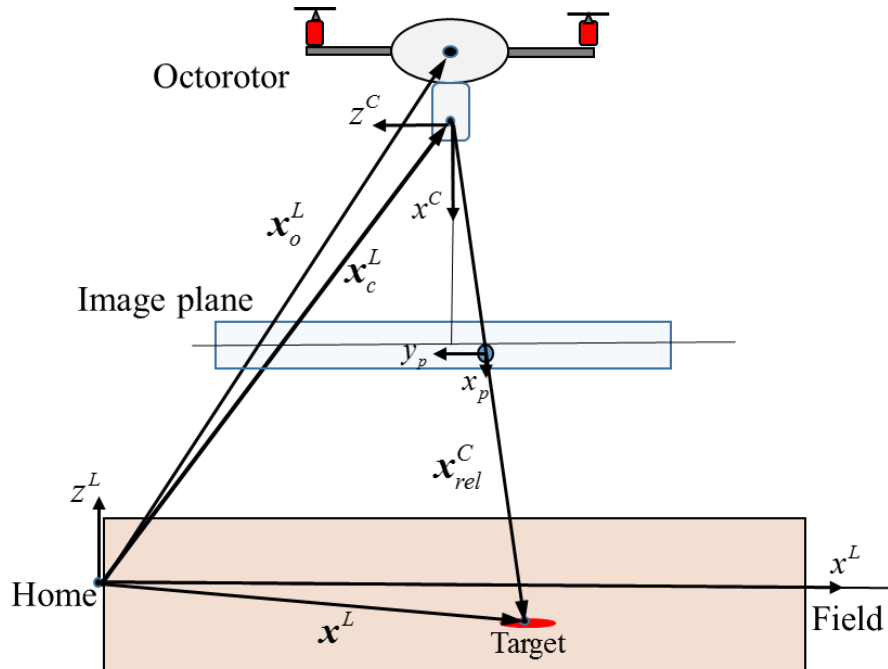


Figure 3.3.3 Orthographic projection model

By applying the correct scaling to convert the remaining 2-D vector from meters into pixels, will result in the orthographic measurement model

$$\mathbf{y} = \begin{pmatrix} x_p \\ y_p \end{pmatrix} = \begin{bmatrix} 0 & l_x & 0 \\ 0 & 0 & l_y \end{bmatrix} \mathbf{x}_{rel}^C + \mathbf{v} \quad (3.3.6)$$

where  $\mathbf{y}$  is the measurement. Note that the depth component of the camera vector, once rotated to the camera frame, makes no contribution to an orthographic projection [12][17]. The scaling variables  $l_x$  and  $l_y$  were determined experimentally by capturing an image of an object of known

dimensions aligned perpendicularly and at a known distance. The pixel span of the object in the image is then measured and related to the physical dimension of the object as

$$l_x = \frac{z_{ref}}{z_c^L} \left( \frac{x_p}{x_{ref}} \right), \quad l_y = \frac{z_{ref}}{z_c^L} \left( \frac{y_p}{y_{ref}} \right) \quad (3.3.7)$$

where  $z_{ref}$ ,  $x_{ref}$ , and  $x_p$  are the experimental distance, object dimension, and pixel span respectively and  $z_c^L$  is the camera altitude. These scaling variables are camera specific and in units of pixels per meter. The sensor noise term  $\mathbf{v}_i$  is zero-mean Gaussian white noise with a constant covariance matrix  $R_v$  as

$$\mathbf{v} \sim N(0, R_v) \quad (3.3.8)$$

The relation of position to pixels resulted in a linear expression and is a combination of an unknown term and a known term plus noise, explicitly separated below in an expansion of Equation (3.3.6),

$$\mathbf{y} = \begin{bmatrix} l_x(R_L^C)_{2,1} & l_x(R_L^C)_{2,2} \\ l_y(R_L^C)_{3,1} & l_y(R_L^C)_{3,2} \end{bmatrix} \begin{pmatrix} x^L \\ y^L \end{pmatrix} + \begin{pmatrix} -l_x(R_L^C)_{2,1} x_c^L - l_x(R_L^C)_{2,2} y_c^L - l_x(R_L^C)_{2,3} (z_c^L - z^L) \\ -l_y(R_L^C)_{3,1} x_c^L - l_y(R_L^C)_{3,2} y_c^L - l_y(R_L^C)_{3,3} (z_c^L - z^L) \end{pmatrix} + \mathbf{v} \quad (3.3.9)$$

where the subscripts of the rotation matrix denote its individual components by row and column and  $z^L$  is the assumed height of each plant. The known term of the output from Equation (3.3.9) can be subtracted directly from the measured pixels, as

$$\tilde{\mathbf{y}} = \mathbf{y} - D = C\mathbf{x}^L + \mathbf{v} \quad (3.3.10)$$

where

$$C = \begin{bmatrix} l_x(R_L^C)_{2,1} & l_x(R_L^C)_{2,2} \\ l_y(R_L^C)_{3,1} & l_y(R_L^C)_{3,2} \end{bmatrix} \quad (3.3.11)$$

and

$$D = \begin{pmatrix} -l_x(R_L^C)_{2,1} x_c^L - l_x(R_L^C)_{2,2} y_c^L - l_x(R_L^C)_{2,3} (z_c^L - z^L) \\ -l_y(R_L^C)_{3,1} x_c^L - l_y(R_L^C)_{3,2} y_c^L - l_y(R_L^C)_{3,3} (z_c^L - z^L) \end{pmatrix} \quad (3.3.12)$$

Note that the measurement matrix  $C$  is constant for each target and the vector  $D$  is composed of all known terms.

### 3.4 Kalman Filter for Geolocation Estimation

A continuous time Kalman filter [19] is derived to execute the position estimation. In the form for continuous systems, the Kalman filter is formulated by combining the prediction and correction steps into a single differential equation for the state estimate and one for its covariance. Since the system for this problem is modeled with no dynamics, the state estimate is solely dependent on the measurement update. Using the derived measurement model, the state estimate vector is denoted as

$$\hat{\mathbf{x}} = \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} \quad (3.4.1)$$

and the predicted measurement as

$$\hat{\mathbf{y}} = C\hat{\mathbf{x}} \quad (3.4.2)$$

The Kalman gain is calculated using the estimation covariance matrix  $P$ , as

$$K = PC^T R_v^{-1} \quad (3.4.3)$$

This Kalman gain is the optimal gain matrix that minimizes the trace of the error covariance [19]. Since the target is modeled as stationary, its estimate depends only on the pixel measurements. Hence, the target position estimate and its covariance are updated, respectively, as

$$\dot{\hat{x}} = K(\tilde{y} - C\hat{x}), \quad \hat{x}(0) = \hat{x}_0 \quad (3.4.4)$$

and

$$\dot{P} = Q - PC^T R_v^{-1} CP, \quad P(0) = P_0 \quad (3.4.5)$$

where  $\hat{x}_0$  and  $P_0$  are the initial conditions for the estimate and its covariance, respectively.

These filter equations were updated using a Euler integration scheme.



# CHAPTER 4

## PERSPECTIVE PROJECTION BASED EXTENDED KALMAN FILTER FOR COMPARISON

### 4.1 Perspective Projection Based Measurement Model

In order to demonstrate the effectiveness of the orthographic projection based geolocation estimator, a comparison between the commonly used perspective projection based method will be performed. The perspective projection model is derived from the orthographic projection model by relaxing the assumption that the image plane remains parallel to the strawberry field. However, the condition that the target, its projection in the image, and the camera point of origin lie in-line still holds. The relative vector of the target as seen from the camera in the camera frame is still expressed as in Equation (3.3.5) above. In the perspective projection, it is assumed that the target relative vector and the target projection in the image plane can be related by a scalar variable. The measurement equation can be expressed, in three dimensions, as

$$\mathbf{y}_{per} = \begin{pmatrix} f \\ x_p \\ y_p \end{pmatrix} = S_p \mathbf{x}_{rel}^C = S_p R_L^C (\mathbf{x}^L - \mathbf{x}_c^L), \quad S_p \in R \quad (4.1.1)$$

where  $S_p$  is the scalar variable and the distance from the camera to the image plane is known as the focal length,  $f$ . Observing the first component of the previous expression, the new scalar variable can be determined as

$$f = S_p R_L^C (1) (\mathbf{x}^L - \mathbf{x}_c^L) \quad (4.1.2)$$

where  $R_L^C(1)$  is the first row of the rotation matrix from frame “L” to frame “C”. Rearranging the expression, the scalar can be solved for, as

$$S_p = \frac{f}{R_L^C(1)(\mathbf{x}^L - \mathbf{x}_c^L)} \quad (4.1.3)$$

The remaining two components of the measurement equation are expressed as

$$\mathbf{y}_{per} = \begin{pmatrix} x_p \\ y_p \end{pmatrix} = \begin{pmatrix} \frac{l_{x,per} f R_L^C(2)(\mathbf{x}^L - \mathbf{x}_c^L)}{R_L^C(1)(\mathbf{x}^L - \mathbf{x}_c^L)} \\ \frac{l_{y,per} f R_L^C(3)(\mathbf{x}^L - \mathbf{x}_c^L)}{R_L^C(1)(\mathbf{x}^L - \mathbf{x}_c^L)} \end{pmatrix} \quad (4.1.4)$$

where the new scale factors are used to convert the two components from meters to pixels respectively. Note that the perspective projection measurement equation contains the target position in the numerator and denominator of each term, which makes it nonlinear.

## 4.2 Extended Kalman Filter

Since the measurement model for the perspective projection based method is nonlinear, an extended Kalman filter is used, which deals with nonlinearities by linearizing the model at the current estimate of each time step. In order to update the Kalman gain and covariance expressions effectively, the measurement nonlinearity is addressed by constructing a Jacobian matrix through linearization. Since the linearization creates an approximation that leaves out higher order terms, the extended Kalman filter is considered a sub-optimal estimation method [19]. First, the measurement equation will be rewritten in terms of four parameters as

$$\mathbf{y}_{per} = \begin{pmatrix} x_p \\ y_p \end{pmatrix} = \begin{pmatrix} i_x/j \\ i_y/j \end{pmatrix} \quad (4.2.1)$$

where

$$i_x = l_{x,per} f R_L^C(2) (\mathbf{x}^L - \mathbf{x}_c^L) \quad (4.2.2)$$

$$i_y = l_{y,per} f R_L^C(3) (\mathbf{x}^L - \mathbf{x}_c^L) \quad (4.2.3)$$

and

$$j = R_L^C(1) (\mathbf{x}^L - \mathbf{x}_c^L) \quad (4.2.4)$$

The Jacobian, evaluated at the estimate, can then be expressed as

$$H = \left. \begin{pmatrix} \frac{\partial x_p}{\partial x^L} & \frac{\partial x_p}{\partial y^L} \\ \frac{\partial y_p}{\partial x^L} & \frac{\partial y_p}{\partial y^L} \end{pmatrix} \right|_{\hat{\mathbf{x}}} = \begin{pmatrix} \frac{j \left( \frac{\partial i_x}{\partial \hat{x}} \right) - i_x \left( \frac{\partial j}{\partial \hat{x}} \right)}{j^2} & \frac{j \left( \frac{\partial i_x}{\partial \hat{y}} \right) - i_x \left( \frac{\partial j}{\partial \hat{y}} \right)}{j^2} \\ \frac{j \left( \frac{\partial i_y}{\partial \hat{x}} \right) - i_y \left( \frac{\partial j}{\partial \hat{x}} \right)}{j^2} & \frac{j \left( \frac{\partial i_y}{\partial \hat{y}} \right) - i_y \left( \frac{\partial j}{\partial \hat{y}} \right)}{j^2} \end{pmatrix} \quad (4.2.5)$$

where

$$\frac{\partial i_x}{\partial \hat{x}} = l_{x,per} f (R_L^C)_{2,1}, \quad \frac{\partial i_x}{\partial \hat{y}} = l_{x,per} f (R_L^C)_{2,2} \quad (4.2.6)$$

$$\frac{\partial i_y}{\partial \hat{x}} = l_{y,per} f (R_L^C)_{3,1}, \quad \frac{\partial i_y}{\partial \hat{y}} = l_{y,per} f (R_L^C)_{3,2} \quad (4.2.7)$$

and

$$\frac{\partial j}{\partial \hat{x}} = (R_L^C)_{1,1}, \quad \frac{\partial j}{\partial \hat{y}} = (R_L^C)_{1,2} \quad (4.2.8)$$

The Kalman gain can now be expressed, using the Jacobian matrix in place of the previous measurement matrix, as

$$K = PH^T R_v^{-1} \quad (4.2.9)$$

This Kalman gain is used to update the position estimate and its covariance, respectively, as

$$\dot{\hat{x}} = K(y - \hat{y}_{per}), \quad \hat{x}(0) = \hat{x}_0 \quad (4.2.10)$$

and

$$\dot{P} = Q - PH^T R_v^{-1} HP, \quad P(0) = P_0 \quad (4.2.11)$$

where the predicted measurement,  $\hat{y}_{per}$ , is found by evaluating the perspective projection measurement equation at the position estimate,  $\hat{x}$ .

### 4.3 Comparison Simulation Settings

A simulation was conducted to compare the two projection methods side by side. The octorotor position was simulated at  $(28.599887^\circ, -81.196495^\circ, 20m)$  with the home position set at  $(28.599859^\circ, -81.196549^\circ, 0m)$ , latitude, longitude, and altitude, respectively. Expressed in the local frame, the octorotor position vector is  $(5.274, 3.115, 20)m$ . The camera position vector, calculated from the octorotor position vector and the gimbal dimensions, is  $(5.233, 3.095, 19.74)m$ . The simulated Euler angles for the UAV and gimbal, respectively, are  $(0^\circ, 0^\circ, 0^\circ)$  and  $(0^\circ, 90^\circ, 0^\circ)$ . Note that the gimbal pitch angle reflects the face-down configuration. A simulated target was

placed at  $(28.599891^\circ, -81.196499^\circ, 0.04m)$  or  $(4.883, 3.560, 0.04)m$ , which is about  $0.35m$  and  $0.45m$  away from the octorotor in the  $x$  and  $y$  directions, respectively. Measurements are simulated using both projection methods respectively, with an added Gaussian measurement noise with covariance of  $16\text{pixel}^2$  in each direction. The scale factors for the orthographic projection based model are chosen as  $33.33\text{pixel}/m$  in both directions. The measurement matrix  $C$  is found as a diagonal matrix with its diagonals as  $33.33\text{pixel}/m$  each and the known parameter  $D$  is  $(-174.3, 103.2)\text{pixel}$ . For the nonlinear perspective projection model, the scale factors are chosen as  $19,048\text{pixel}/m$  in each direction, in order to reproduce measurements of similar magnitudes, for comparison purposes. The focal length is  $0.035m$  and the Jacobian Matrix  $H$  is calculated as a diagonal matrix with a value of  $33.841\text{pixel}/m$  in each diagonal. Note the closeness in magnitude of the measurement matrices for the linear and nonlinear projection methods, which comes as a result of choosing the perspective scale factors to reproduce similar measurements as in the linear fashion. The filter matrices,  $Q$ ,  $R_v$ , and  $P_0$ , are chosen as diagonal matrices with values of  $0.0025m^2/s$ ,  $16s\cdot\text{pixel}^2$ , and  $0.27m^2$  for each diagonal in the respective matrix, and the estimate is initialized below the octorotor position at  $\hat{\mathbf{x}}_0 = (5.274 \quad 3.115)^T m$  (corresponding to the center of the image). These values are used for both filtering methods. The simulation is run for a duration of  $15s$  with a time step of  $0.05s$ .

#### 4.4 Simulation Results and Discussion

The position estimation results for both filters in both directions are plotted in Figure 4.4.1 below, as well as the actual target position. It is seen that both filters perform very similarly, reaching their target with a settling time of  $1.6s$  and  $1.2s$  for the Kalman filter and EKF,

respectively and with a maximum encountered position error of  $0.11m$  for the Kalman filter and  $0.072m$  for the EKF in either direction, as shown in Table 4.4.1 below. It should be noted that the clearest distinction between the performance of the two filters is the computational time, where the EKF is more intensive in that it must reevaluate the Jacobian matrix at each time step.

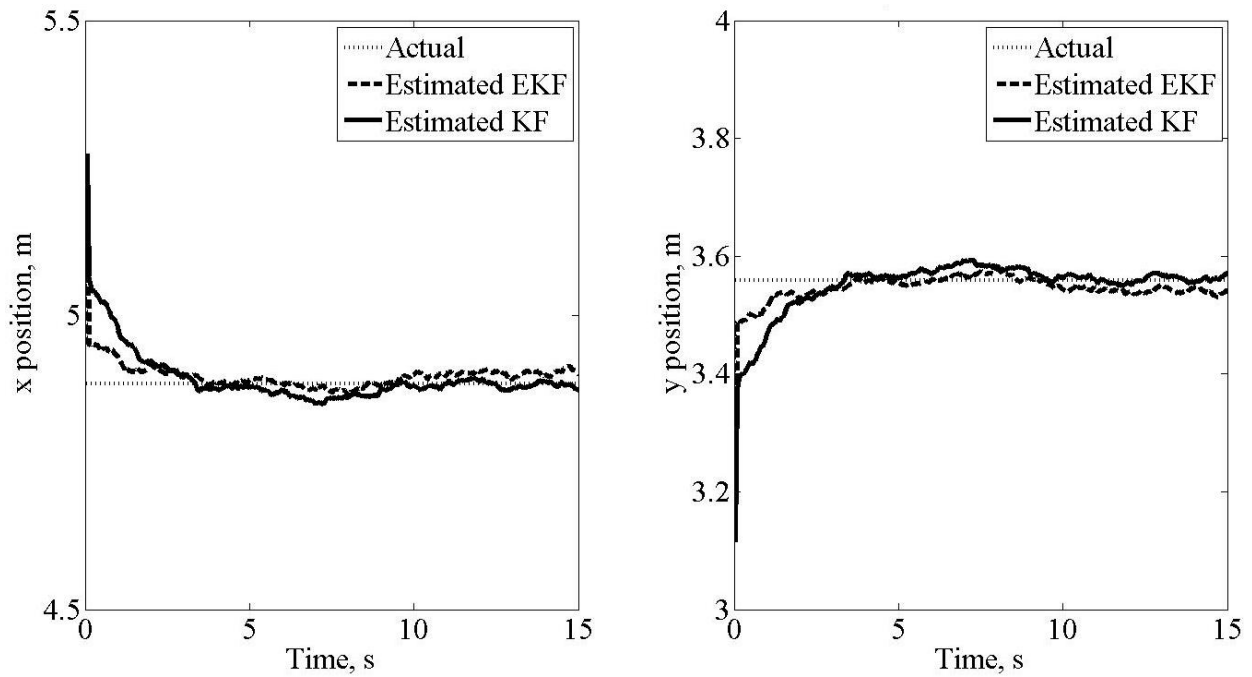


Figure 4.4.1 Actual simulated target position and position estimate for both KF and EKF in both directions

Table 4.4.1 Simulation results comparing KF and EKF performance

	Orthographic Projection Geolocation		Perspective Projection Geolocation	
	$x$	$y$	$x$	$y$
Actual position, $m$	4.883	3.560	4.883	3.560
Estimated position, $m$	4.882	3.559	4.891	3.551
Final position error, $m$	0.001	0.001	-0.008	0.009
Maximum position error, $m$	0.108	-0.110	0.070	-0.072
Actual output, $pixel$	308.3	224.5	308.1	224.3
Predicted measurement, $pixel$	308.2	224.4	308.4	224.5
Minimum covariance trace, $m^2$	0.012		0.0118	
Settling time, $s$	1.60		1.20	
Computational time, $s$	0.222		0.658	

For better comparison, the simulated measurements for both projection methods, along with the predicted measurements of the corresponding filter, were plotted separately per direction as Figure 4.4.2 for the  $x$  direction and Figure 4.4.3 for the  $y$  direction, respectively. It is shown that the filters were both successful in removing the Gaussian measurement noise, as designed.

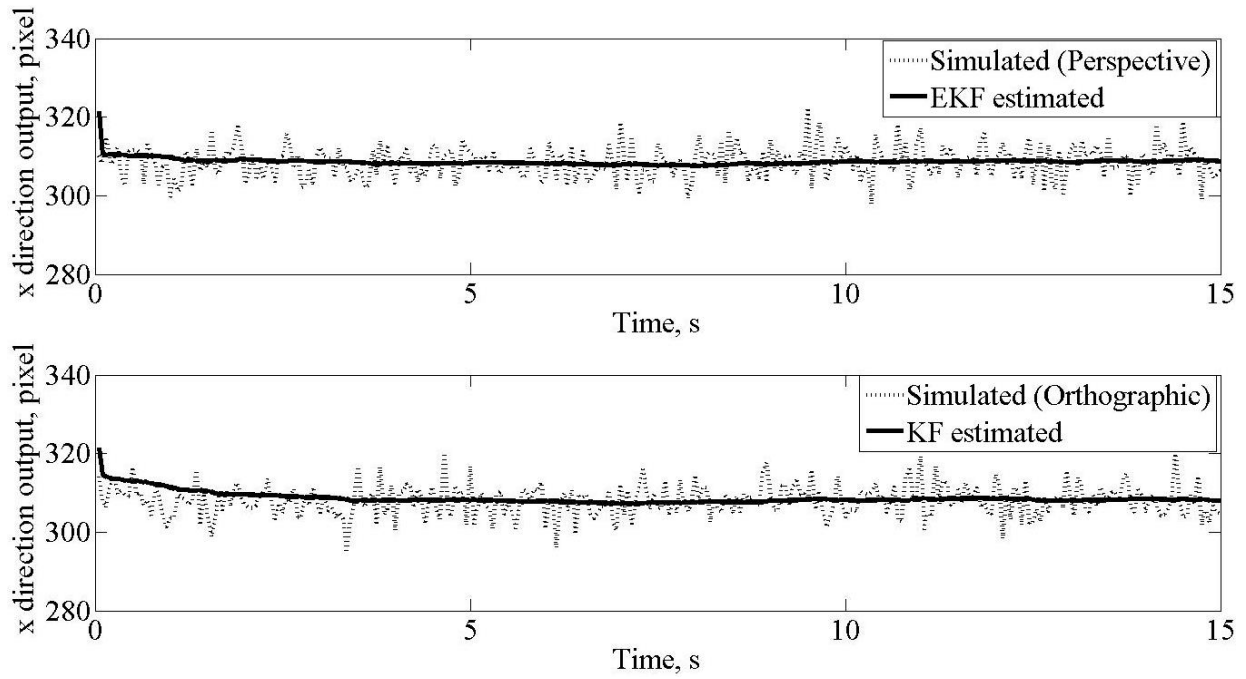


Figure 4.4.2 Simulated and predicted measurements for the perspective projection with EKF and orthographic projection with KF, in the  $x$  direction



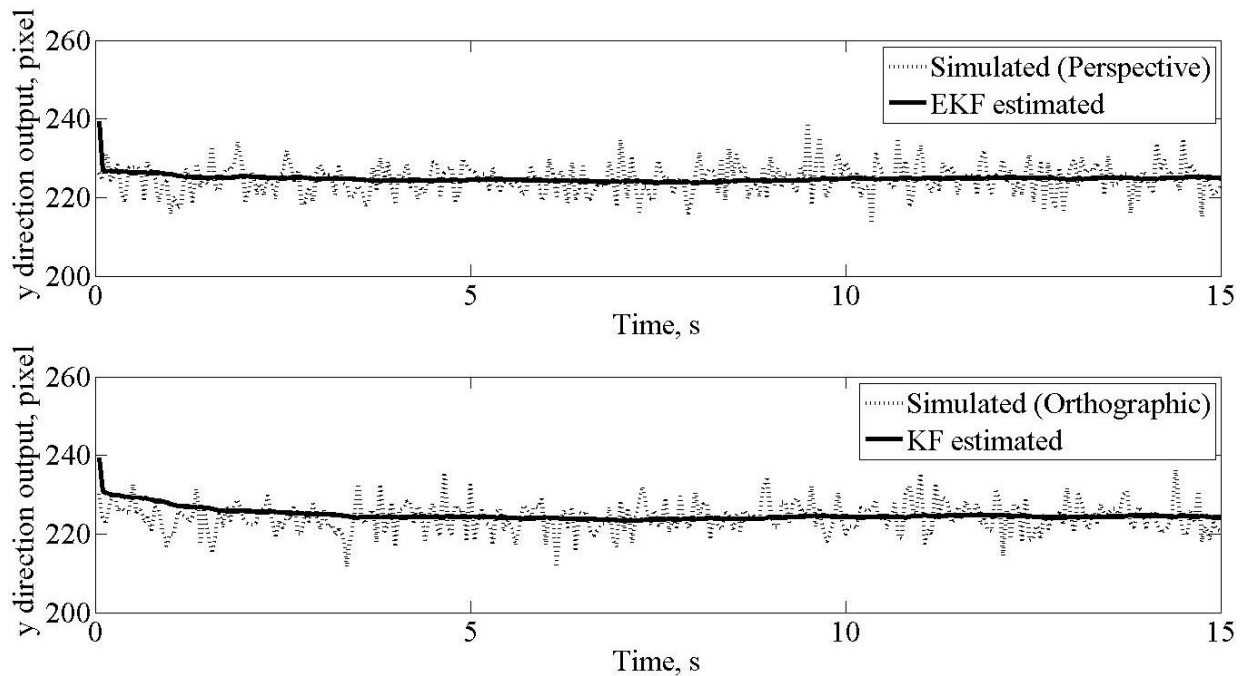


Figure 4.4.3 Simulated and predicted measurements for the perspective projection with EKF and for the orthographic projection with KF, in the y direction

The covariance trace is shown to be minimized for both the KF and EKF in Figure 4.4.4 below. Both plots demonstrate almost identical performance in the regard of optimally minimizing the covariance. Overall, this simulation demonstrates that KF and EKF, and their models, are designed to perform in very similar fashion, except for the notable difference in computational time, is an important factor when considering implementation onto the UAV.

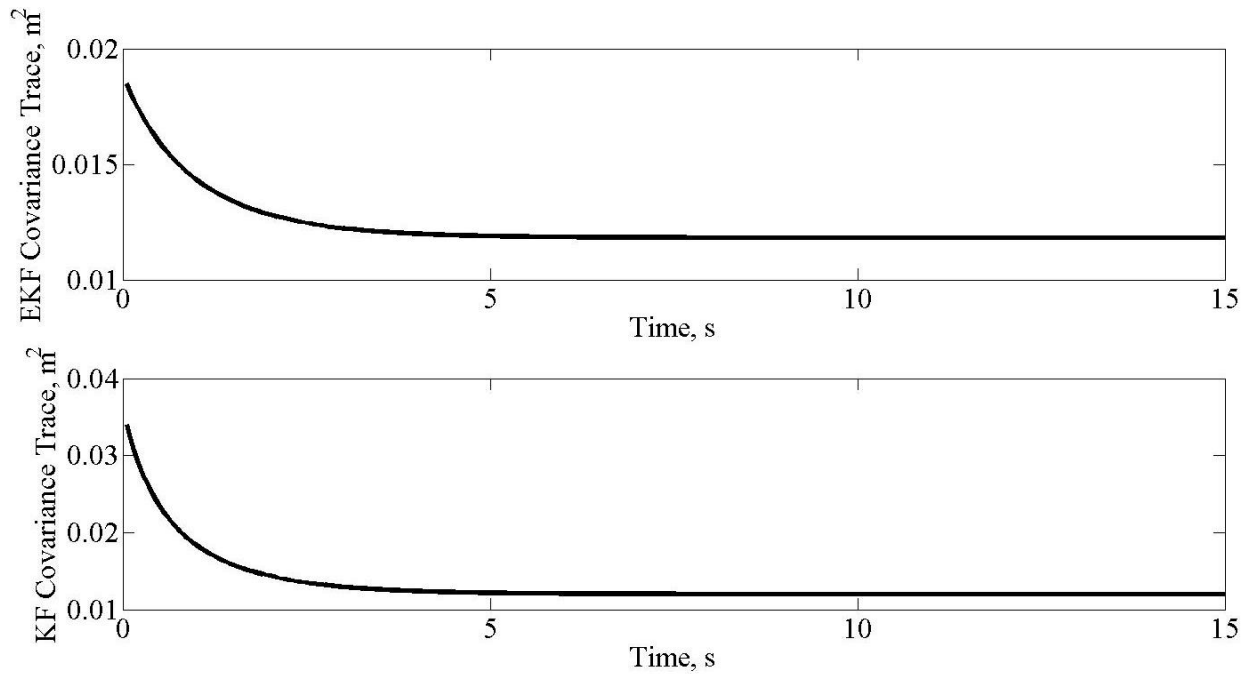


Figure 4.4.4 Position estimation covariance trace for the KF and EKF, respectively

## 4.5 Comparison Experiment Settings

An ad-hoc experiment is conducted to demonstrate the effectiveness of the orthographic projection based geolocation approach and compare it to the perspective projection based approach using real flight data. A red box is placed in a grassy field to represent a diseased leaf and the rotorotor UAV is flown in hover flight, acquiring video of the target with a Canon Powershot A90 RGB camera. The video runs for 15.7s at a frame rate of 15fps, corresponding to a step time of 0.0667s. A screenshot of the video with the red target is shown in Figure 4.5.1 below. The images acquired were post-processed in MATLAB along with the corresponding GPS values and rotational information from the UAV during flight.



Figure 4.5.1 Screenshot of the video taken from the octocopter UAV of the red box in a grassy field

The identification criterion used to find the red box in each image is based on a color intensity scheme, where pixels with red color intensity greater than 245 and green color intensity less than 200 are chosen. These pixels are then joined into contours with adjacent selected pixels and the centroid and areas of these contours are calculated. The contour with the greatest area is taken as the desired target and its centroid is used as the pixel measurements.

Due to the lack ability of the gimbal controller to log the gimbal Euler angle rotations, a perfect gimbal assumption is taken. The gimbal was balanced to the front arm of the octocopter UAV and its Euler angles are taken as  $(0^\circ, 90^\circ, 0^\circ)$ , respectively. The UAV Euler pitch and roll angles are  $0^\circ$  each, but a yaw angle of  $197.8^\circ$  is used, which is the average of the yaw angle measurements encountered during flight. The orthographic projection scale factors are calculated as

231.48 $pixel/m$  in both directions. The nonzero UAV yaw angle results in coupling of the  $x$  and  $y$  directions, which in turn gives a non-diagonal measurement matrix,

$$C = \begin{bmatrix} -220.40 & 70.76 \\ 70.76 & 220.40 \end{bmatrix} \frac{pixel}{m} \quad (4.5.1)$$

The known adjustment vector  $D$  results in

$$D = \begin{pmatrix} 811.42 \\ 115.35 \end{pmatrix} pixel \quad (4.5.2)$$

and the UAV yaw angle produces a rotation matrix from the inertial frame  $L$  to the camera frame  $C$  as

$$R_L^C = \begin{bmatrix} 0 & 0 & -1 \\ -0.952 & 0.306 & 0 \\ 0.306 & 0.952 & 0 \end{bmatrix} \quad (4.5.3)$$

The filter covariance matrices used are

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \frac{m^2}{s} \quad (4.5.4)$$

$$R_v = \begin{bmatrix} 1722.3 & 0 \\ 0 & 1332.2 \end{bmatrix} s \cdot pixel^2 \quad (4.5.5)$$

and

$$P_0 = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix} m^2 \quad (4.5.6)$$

respectively. The position estimate was initialized at the first encountered UAV position, or  $(3.174, -1.557)m$ , corresponding to the center of the image frame.

#### 4.6 Experiment Results and Discussion

The octorotor position, shown in Figure 4.6.1 below, depicts slight movements in all three directions of  $1.97m$ ,  $0.7m$ , and  $2.38m$  range, respectively. These movements are due to errors in stable hover resulting from remote controlled, manual flight. Since the octorotor yaw angle was about  $197.8^\circ$  during the flight, the octorotor movements reflect upon the position and measurement movements in a coupled manner.

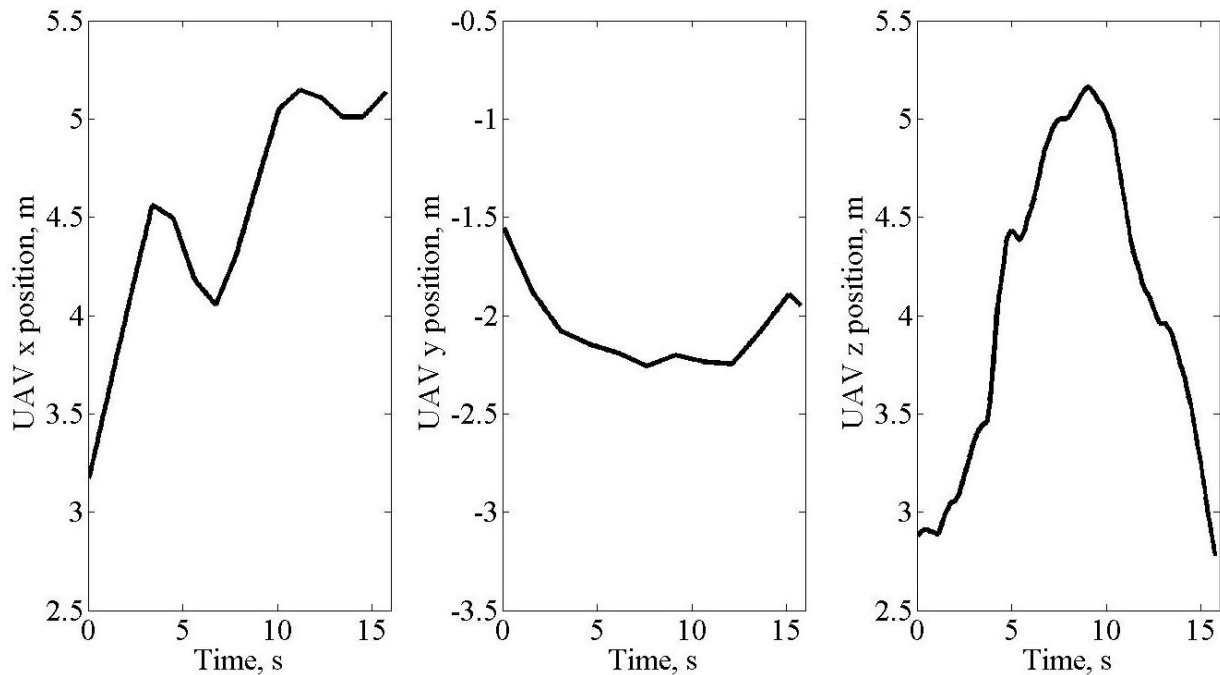


Figure 4.6.1 Octorotor position in all three directions, respectively

The measured centroid pixels of the box for the duration of the video are shown in Figure 4.6.2 and Figure 4.6.3 below, along with the predicted measurement of both the extended Kalman filter and Kalman filter, in both directions respectively. It can be observed from the figures that both the filtering methods tracked the measurement movements very well in either direction, even though the measurements spanned across a range of about 400 pixels in the  $x$  direction and 240 pixels in the  $y$  direction. Also note that, since the UAV is flying at a non-zero yaw angle, the movement of the UAV in both directions is cross-coupled into the movement of the measurements, in both directions.

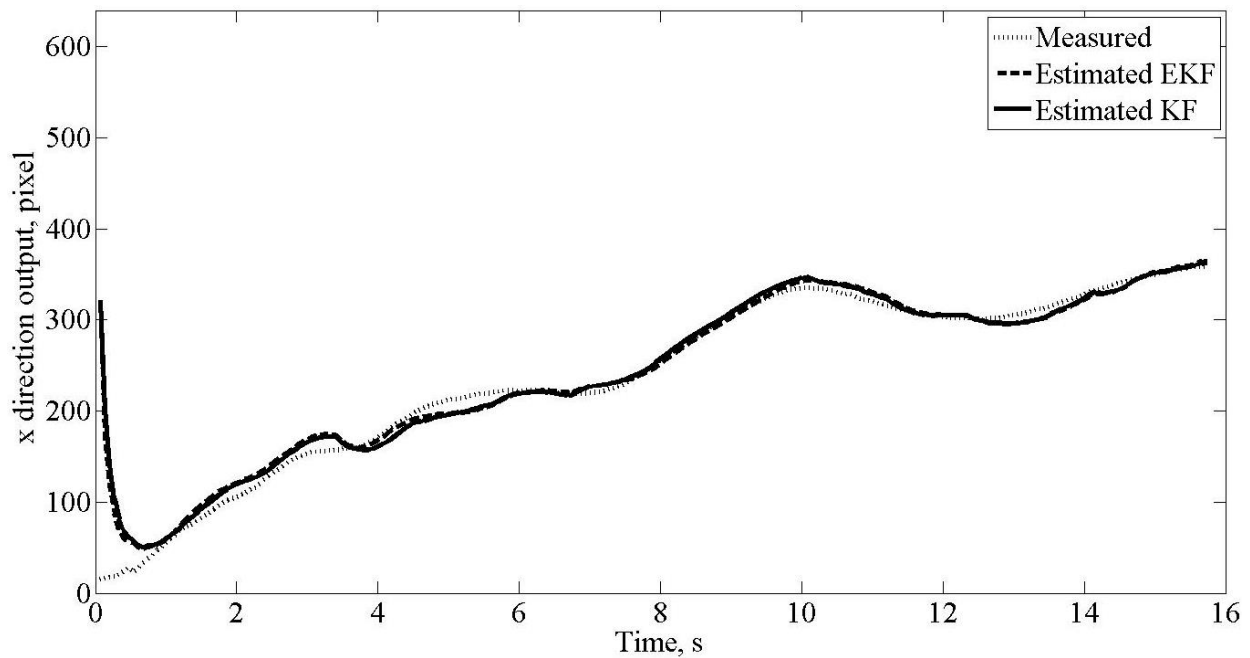


Figure 4.6.2 Actual and predicted measurements, for both the geolocation methods, respectively, in the  $x$  direction

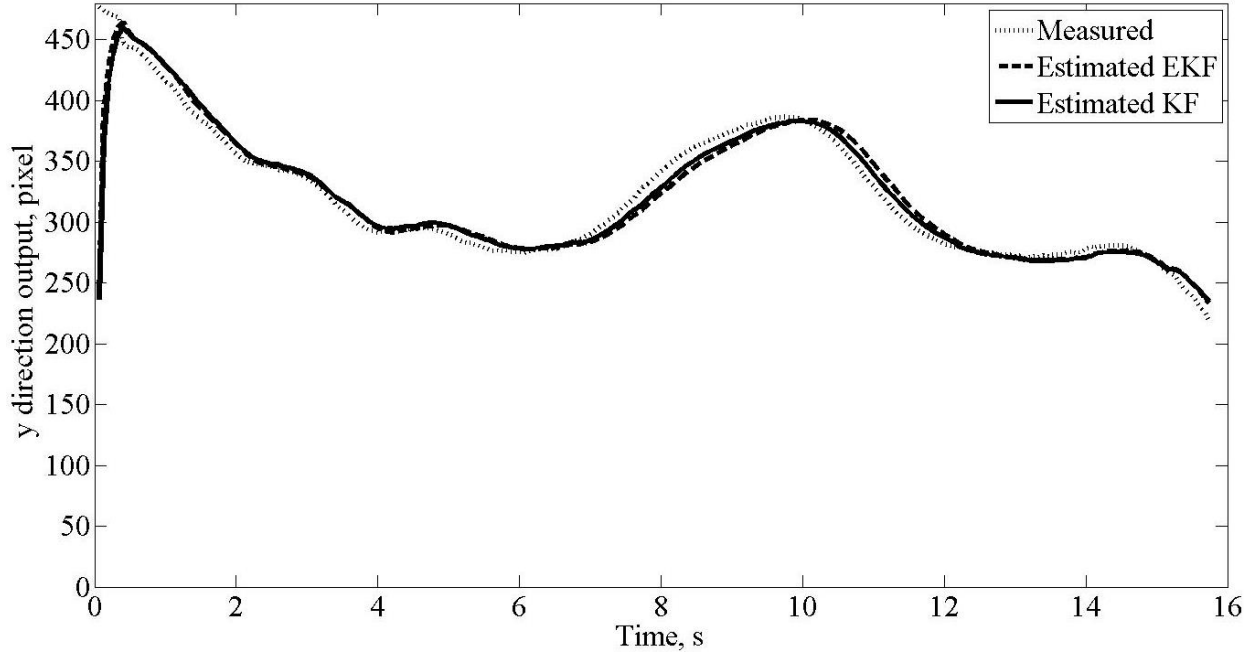


Figure 4.6.3 Actual and predicted measurements, for both the geolocation methods, respectively, in the y direction

The estimated position for both the estimation methods is shown in Figure 4.6.4, for each of the directions respectively. The actual target position was determined by taking the octorotor position at the instance when the red box cross the middle of the image. The octorotor position is also added to the figure for comparison purposes. It is shown that the position estimates converge to a value with a minimal error of  $0.168m$  and  $0.058m$  for the Kalman filter geolocation method and of  $0.185m$  and  $0.051m$  for the extended Kalman filter geolocation method. The EKF had the greatest encountered error, seen in the y direction estimation plot of Figure 4.6.4 and tabulated in Table 4.6.1, where the maximum error is shown to be  $0.769m$ . The maximum error for the KF was  $0.534m$ , which barely does not satisfy the goal of  $0.5m$ . This can be attributed to the octorotor movements in flight, which violate the hover assumption.

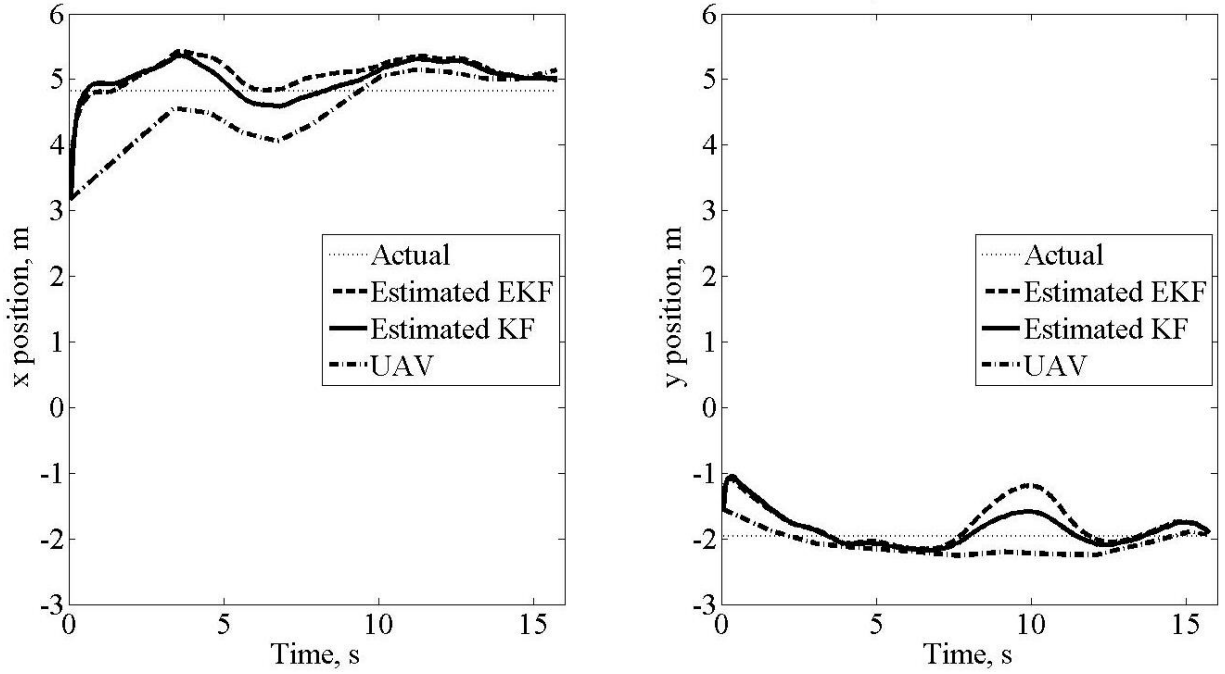


Figure 4.6.4 Estimated position for both geolocation methods, along with the actual position and octorotor position, in the  $x$  direction

Table 4.6.1 Experiment results comparing KF to EKF performance

	Orthographic Projection Geolocation		Perspective Projection Geolocation	
	$x$	$y$	$x$	$y$
Actual position, $m$	4.829	-1.961	4.829	-1.961
Estimated position, $m$	4.995	-1.931	5.015	-1.938
Final position error, $m$	0.168	0.058	0.185	0.051
Maximum position error, $m$	0.534	0.373	0.598	0.769
Minimum covariance trace, $m^2$	0.337		0.291	
Settling time, $s$	0.80		0.80	
Computational time, $s$	0.0428		0.166	



The trace of the covariance for both filtering methods is given in Figure 4.6.5 below, where convergence can be seen for both methods. However, the EKF demonstrates a point of sub-optimal minimization due the fact that it has to recalculate the Jacobian measurement matrix  $H$  at every time step, which creates region of covariance trace growth, before declining back to the its final value. The clearest distinction between the two methods' performance is the computational time, where the EKF geolocation method performed four times slower than the KF method, shown in Table 4.6.1.

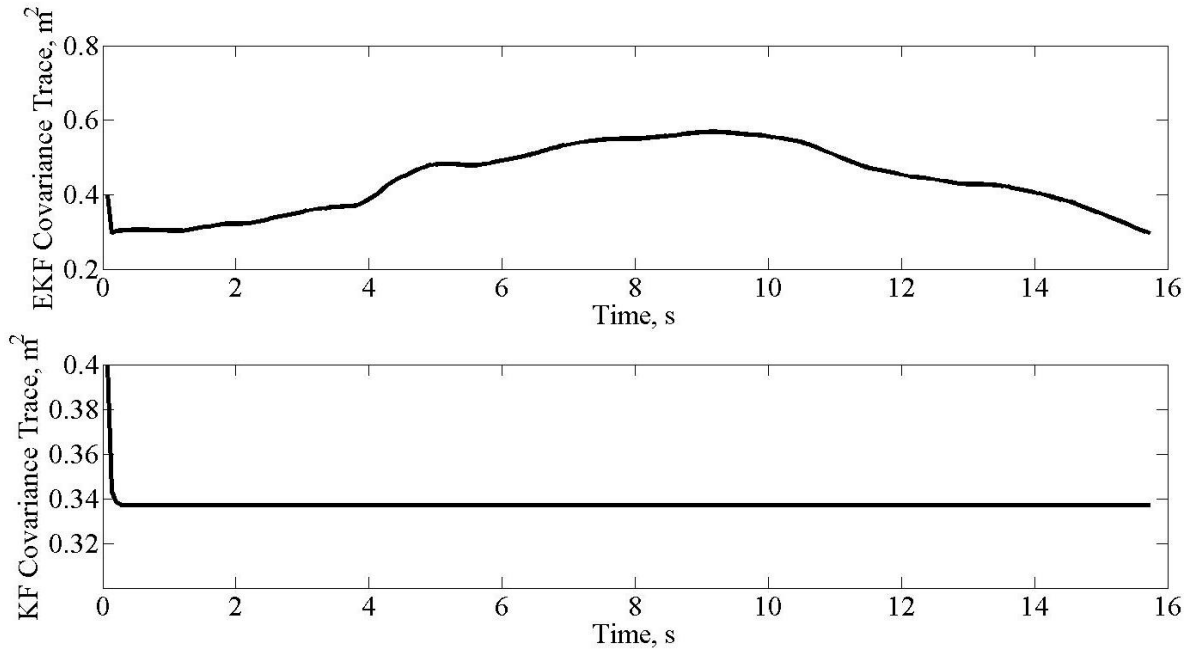


Figure 4.6.5 Trace of the covariance for both filtering methods, respectively

## **CHAPTER 5**

### **REAL-TIME IMPLEMENTATION**

#### **5.1 Real-time Implementation Hardware Setup**

The geolocation method is designed for low computational time, in order to be implemented in real-time. A Raspberry Pi 2 CPU board, or the Pi, is installed onto the octorotor UAV to host the geolocation algorithm. The Pi is directly connected to the APM 2.6 flight controller through its “power”, “ground”, *RX*, and *TX* pins, to the telemetry port of the flight controller. It is powered by its connection to the APM as well, since it receives 5V. Additionally, the Pi has four USB ports with which it can connect to its other peripherals. An RGB camera and a spectrometer can be connected through two of the USB ports to the Pi, or a spectral camera alternatively, for the disease detection portion. A USB flash drive occupies the third port and the last port is designated for the XBee radio telemetry device, used to communicate to the unmanned ground vehicle. A diagram of the flow of information is provided in Figure 5.1.1 below. The avionics will provide the flight measurements to the APM which will organize them and send them to the Pi through a MAVLink connection, established through the serial communication. This will include all flight data available and useful to the geolocation algorithm. Additionally, the camera setup will provide the image along with the suspected pixel measurements to the Pi directly. Once the location of the diseased region is known, the Pi will send those coordinates by radio frequency through the XBee telemetry device to the device’s pair on the UGV.

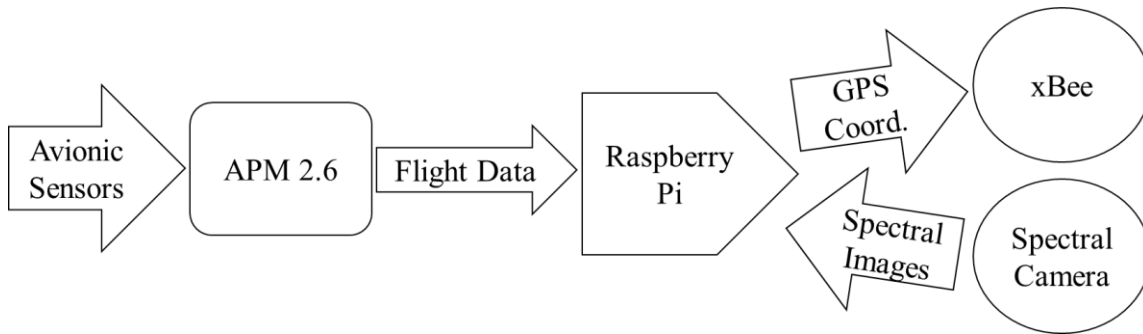


Figure 5.1.1 Information flow diagram for the real-time implementation

## 5.2 Geolocation Algorithm for Real-Time Use

The proposed geolocation method was coded in Python, in order to implement onto the Raspberry Pi 2 board. A daemon script is put into place on the Pi bash files, that will initiate the geolocation algorithm upon start-up of the Pi. This algorithm begins with establishing the MAVLink connection, in which the Pi maintains up to date with any changes in the UAV position and attitude information. If the gimbal controller allowed for logging capability, this would also be updated onto the Pi. The algorithm then runs the camera until a suspected region and its measurement is acquired, in which a loop is then initiated, shown in Figure 5.2.1 under the Image Processing component. The loop, seen in the Estimator component of Figure 5.2.1, first updates the UAV information through MAVLink. If this is the first time step in the estimation, it initializes the covariance matrices and the position estimate at the current UAV position, which corresponds to the center of the image. Then the Kalman filter equations are performed, updating the position estimate and its covariance. The current step measurements, position estimate, and covariance update are all logged onto the USB flash drive, along with the images.

An additional step is required for real-time implementation, which is to have a criterion for estimation convergence, in order to properly terminate the loop, shown in the Convergence component of Figure 5.2.1. The position estimate and its covariance, after being updated, are hence checked against their previous value. If the error in the estimate and its covariance are both small enough to satisfy a set tolerance, then convergence has occurred. The position estimate, which is in the local inertial frame  $L$ , is then converted back to GPS coordinates through the inverse of the method in which the octorotor position vector is acquired. Once this has occurred, a serial connection through the XBee radios is established. The XBee radio on the UGV side is connected to an Arduino Mega 2560. The Pi will first send a “g” indicating that it has converged and wait for a signal, determining whether the UGV is ready to receive data or not. Once received, the Pi will send the latitude and longitude values of the final estimate. If the estimator diverges, it will also initialize the serial communication and send a “d” over to the UGV, to indicate that a region has been detected, but the position estimation diverged. The Pi will then wait for confirmation from the UGV in the form of a “c”, which tells the Pi to continue scanning, or “k” which will indicate to the Pi to finish the geolocation procedure and begin the shutdown process.

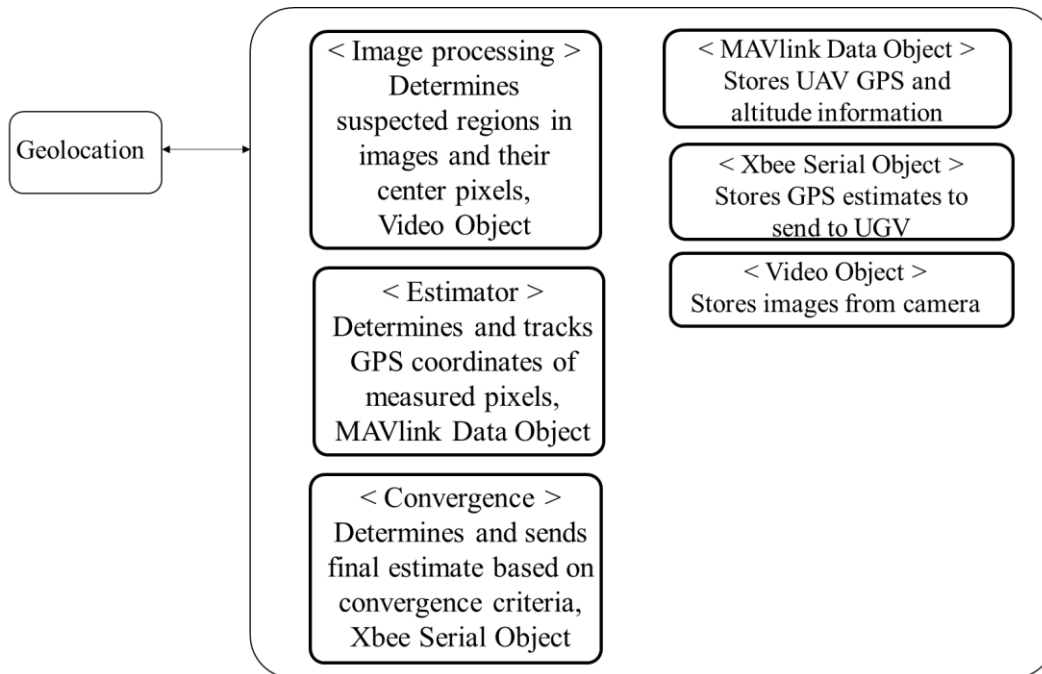


Figure 5.2.1 Real-time geolocation algorithm structure components and objects

## **CHAPTER 6**

### **MULTI-TARGET GEOLOCATION WITH DATA ASSOCIATION**

#### **6.1 Multi-Target Geolocation Problem Formulation**

The multi-target geolocation problem in this study is a set of simultaneous position estimation for multiple diseased leaves among many suspected measurements seen in the acquired images. The process for multiple target geolocation is shown in Figure 6.1.1. The required additions for considering multiple targets are a method for correlating each measurement to its respective target correctly and a method for incorporating these correlated measurements into the position estimate. A logic-based correlation method is proposed, where the measurements are validated based on a chi-squared probability criterion. The Probabilistic Data Association technique is adopted for measurement association. Both of these approaches are detailed in this chapter, and their roles are shown in Figure 6.1.1 below.

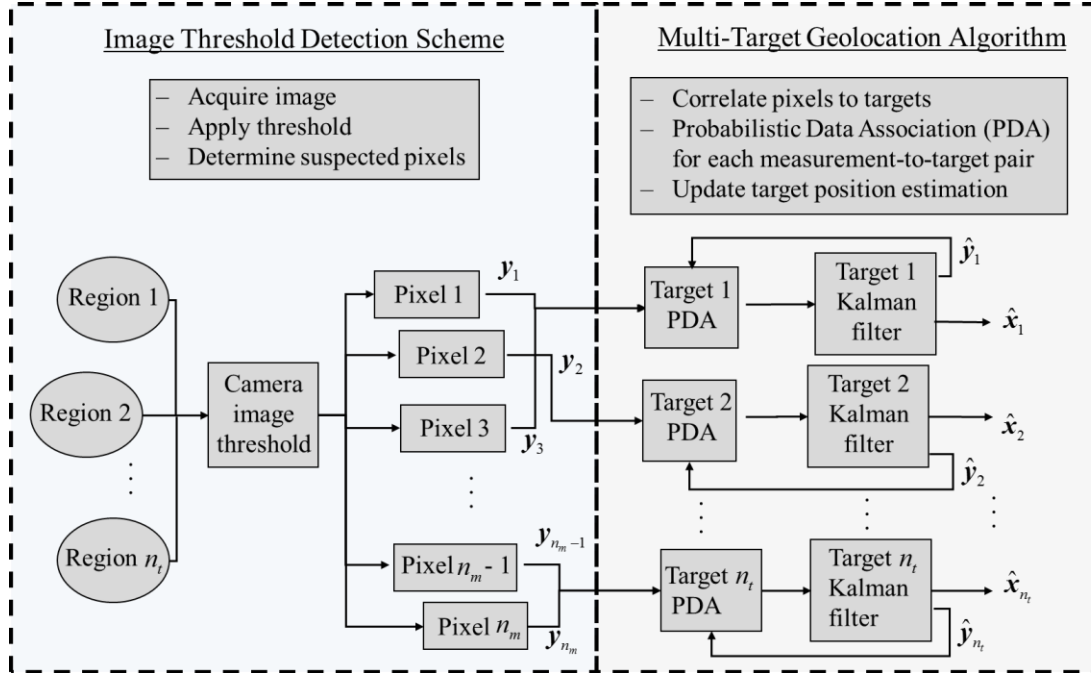


Figure 6.1.1 Multi-target geolocation algorithm overall structure

It is assumed that there are  $n_t$  diseased plants in each image, which are still considered stationary at their roots and their leaves move only as perturbed by winds. Any large, slow movements in the predicted measurement trajectories are still attributed to the UAV flight movements, whereas any small, faster fluctuations can be attributed to the measurement association. Additionally, since all plants are stationary, the position and measurement trajectories should be similar for all targets. Hence, the case of colliding or intersecting targets, briefly discussed in [20], can be ignored. The shallow plant assumption and face-down gimbal configuration is still valid, hence the orthographic projection measurement model still applies for multiple targets. The position error of  $0.5m$  or less is still desired.

## 6.2 Multi-Target Geolocation Process Model

It is assumed that there are  $n_t$  diseased plants in an image. Each target plant is stationary, meaning that the state dynamics is modeled as

$$\dot{\mathbf{x}}_j^L = \begin{pmatrix} \dot{x}_j^L \\ \dot{y}_j^L \end{pmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \mathbf{w}_j, \quad j = 1, \dots, n_t \quad (6.2.1)$$

The process noise  $\mathbf{w}_j$ ,  $j = 1, \dots, n_t$  accounts for disturbances, such as wind, and is assumed to be zero-mean, Gaussian white noise with variance matrix  $Q$  as

$$\mathbf{w}_j \sim N(0, Q), \quad j = 1, \dots, n_t \quad (6.2.2)$$

## 6.3 Multi-Target Geolocation Measurement Model

The octorotor position vector in the inertial frame  $L$ , the rotation matrix from the inertial frame  $L$  to the octorotor body frame  $B$ , the rotation matrix from the body frame  $B$  to the camera frame  $C$ , and the camera position vector in the inertial frame  $L$  are all determined in the same fashion as for the single target geolocation, described in Section 3.3. As shown in Figure 6.3.1 below, the target position vector, its projection in the image, and the camera origin must still lie in-line, for each target. The vectors between the targets and the camera, as viewed in the camera frame, are expressed as

$$\mathbf{x}_{rel,j}^C = R_L^C (\mathbf{x}_j^L - \mathbf{x}_c^L), \quad j = 1, \dots, n_t \quad (6.3.1)$$



where  $\mathbf{x}_j$  is the unknown position vector of the  $j^{\text{th}}$  target plant and the subscript “rel” denotes that it is the relative vector between the camera position vector and the  $j^{\text{th}}$  target position vector.

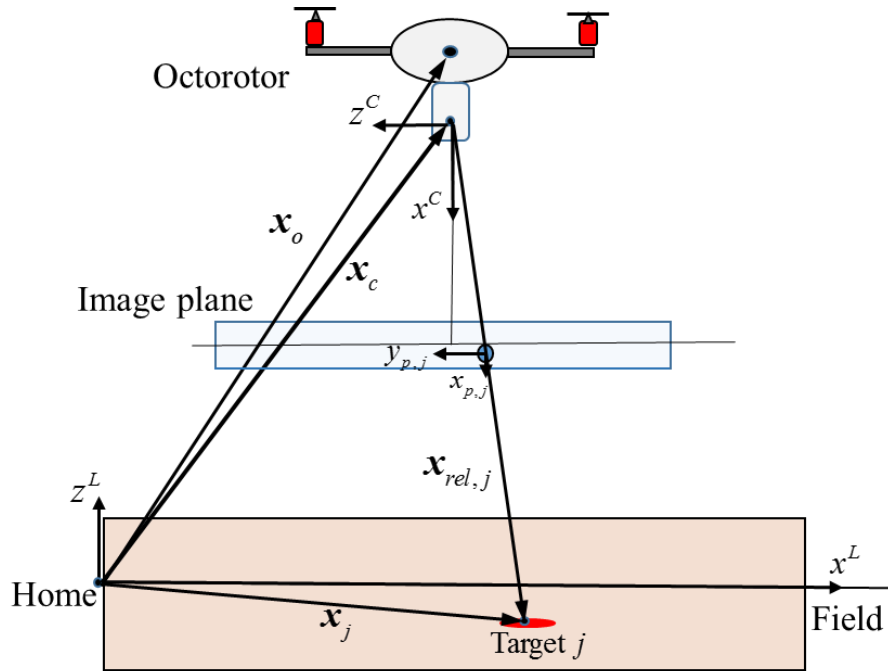


Figure 6.3.1 Orthographic projection diagram for multiple targets

In addition to dealing with multiple targets, there is the added consideration of dealing with an even greater number of measurements, where more than one measurement is within a validation region for each target. Hence the measurement equation, using the orthographic projection method, will be expressed as

$$\mathbf{y}_i = \begin{pmatrix} x_{p,i} \\ y_{p,i} \end{pmatrix} = \begin{bmatrix} 0 & l_x & 0 \\ 0 & 0 & l_y \end{bmatrix} \mathbf{x}_{rel,j} + \mathbf{v}_i, \quad j = 1, \dots, n_t, \quad i = 1, \dots, n_m \quad (6.3.2)$$

where  $\mathbf{y}_i$  is the  $i^{\text{th}}$  measurement and  $n_m$  is the total number of measurements considered, which can be different than the total number of targets  $n_t$ . The scale factors are calculated as before and the measurement noise term is assumed to be a zero-mean Gaussian for each measurement  $i$  with a constant covariance matrix  $R_v$  as

$$\mathbf{v}_i \sim N(0, R_v), \quad i = 1, \dots, n_m \quad (6.3.3)$$

The measurement equation for each measurement is expanded, as before, to

$$\mathbf{y}_i = C \begin{pmatrix} x_j^L \\ y_j^L \end{pmatrix} + D + \mathbf{v}_i, \quad j = 1, \dots, n_t, \quad i = 1, \dots, n_m \quad (6.3.4)$$

where  $C$  and  $D$  are found from Equations (3.3.11) and (3.3.12) respectively. The measurements are adjusted in the same fashion and result in

$$\tilde{\mathbf{y}}_i = \mathbf{y}_i - D = C \begin{pmatrix} x_j^L \\ y_j^L \end{pmatrix} + \mathbf{v}_i, \quad j = 1, \dots, n_t, \quad i = 1, \dots, n_m \quad (6.3.5)$$

## 6.4 Logic-Based Measurement-to-Target Correlation Approach

Estimating a target position is strongly dependent on whether the measurements originate from the target of interest or from other sources, such as other targets, false alarms, or disturbances. In the case of identifying targets by color or spectral intensity thresholding, as occurs in agricultural disease detection, a common occurrence is that false-alarm sources satisfy the threshold momentarily, due to varying lighting or other applicable conditions, which can provide misleading

measurements. More importantly however is the occurrence of multiple suspected regions in an image, which is the primary motivator for addressing uncertain measurements in this case.

For single target measurement probabilistic data association, a validation region is formed to include the more probable measurements in the estimation, based on chi-squared probability analysis, and disregard the other measurements [20]. However, this discards the possibility of estimating additional targets outside the set validation region. Instead, all measurements are to be analyzed for their fit into the existing target validation regions, and considered as new targets if they do not fit into any region, similar to topics discussed in [20]. If a target trajectory no longer contains any valid measurements, it can be discontinued. The target estimates are expressed as

$$\hat{\mathbf{x}} = \begin{pmatrix} \hat{x}_j \\ \hat{y}_j \end{pmatrix}, \quad j = 1, \dots, n_t \quad (6.4.1)$$

and their predicted measurements as

$$\hat{\mathbf{y}}_j = C\hat{\mathbf{x}}_j, \quad j = 1, \dots, n_t \quad (6.4.2)$$

A validation criterion was established based on the innovation of the  $i^{\text{th}}$  measurement from the  $j^{\text{th}}$  target, as

$$\omega_{j,i} = \begin{cases} 1, & \boldsymbol{\eta}_{j,i}^T S_j^{-1} \boldsymbol{\eta}_{j,i} \leq \sigma \\ 0, & \boldsymbol{\eta}_{j,i}^T S_j^{-1} \boldsymbol{\eta}_{j,i} > \sigma \end{cases}, \quad i = 1, \dots, n_m, \quad j = 1, \dots, n_t \quad (6.4.3)$$

where  $\sigma$  is the innovation tolerance based on chi-squared probability, as done in [20]. Note that the total number of measurements changes with each acquired image, and the total number of targets changes based on the decisions made in this approach. The innovation is determined as

$$\boldsymbol{\eta}_{j,i} = \tilde{\mathbf{y}}_i - \hat{\mathbf{y}}_j = \tilde{\mathbf{y}}_i - C\hat{\mathbf{x}}_j, \quad j = 1, \dots, n_t, \quad i = 1, \dots, n_m \quad (6.4.4)$$

where  $y_i$  is the  $i^{\text{th}}$  pixel measurement,  $\hat{y}_j$  is the  $j^{\text{th}}$  target estimated pixels, and the target innovation covariance  $S_j$  is

$$S_j = CP_jC^T + R_v, \quad j=1, \dots, n_t \quad (6.4.5)$$

Each of these  $\omega_{ji}$  values are placed into a binary matrix  $\Omega$ , as shown in Figure 6.4.1 below. If measurement  $i$  satisfies the above criterion (its corresponding column contains at least one positive value) then it is considered as a possible measurement for target  $j$  (which is the target corresponding to the row where the nonzero value resides). Once all targets are checked and the corresponding column of the binary matrix is filled, further conditions are evaluated for that measurement. If measurement  $i$  does not belong to any target, denoted by an empty column  $i$  in  $\Omega$ , or

$$\sum_{j=1}^{n_t} \omega_{j,i} = 0 \quad (6.4.6)$$

a new target is initialized at that measurement location ( $n_t$  increases). The estimates are initialized by inverting the measurement Equation (3.3.10) for the first measurement considered. For the case where a measurement corresponds to two or more targets or

$$\sum_{j=1}^{n_t} \omega_{j,i} > 1 \quad (6.4.7)$$

the target regions are combined into one estimate by taking the average of the two or more pixel estimates  $\hat{y}_j$  and summing the innovation covariance of each target ( $n_t$  decreases). This provides a new, single trajectory centered in between the previous pixel estimates with an increased validation region that will include all measurements previously belonging to the combined targets. It also

avoids needing to evaluate the joint probability of a measurement considered for multiple targets, which can be computationally intensive and unnecessary for stationary targets. After these checks have been performed for all given measurements, an additional condition is analyzed for each target being estimated. If a target has no measurements corresponding to it, denoted by an empty row  $j$  in  $\Omega$ , or

$$\sum_{i=1}^{n_m} \omega_{j,i} = 0 \quad (6.4.8)$$

that target is discarded and no longer estimated ( $n_t$  decreases). If not, then the probabilistic data association technique is performed for the remaining target tracks and their validated measurements.

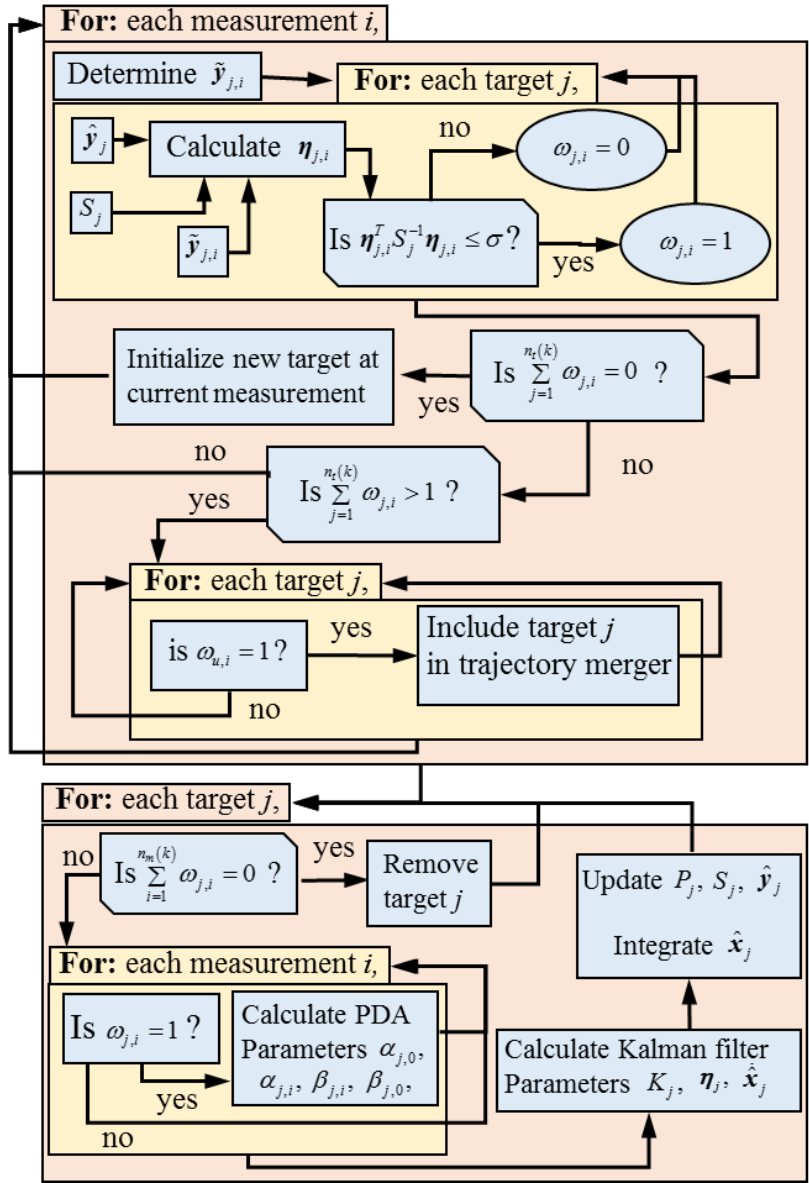


Figure 6.4.1 Logic-based approach to measurement-to-target correlation flow chart

## 6.5 Probabilistic Data Association for Each Target

The measurements that fall within a target validation region are used to update the state estimation Kalman filter for that trajectory as follows. Depending on the level of confidence

desired in the measurement validation, a certain chi-squared value  $\sigma$  is chosen which corresponds to a level of validation gate probability, denoted as  $P_G$ . The probability of each measurement being the correct measurement is

$$\beta_{j,i} = \frac{\alpha_{j,i}}{\alpha_{j,0} + \sum_{r=1}^{n_m} \alpha_{j,r}}, \quad i=1, \dots, n_m, \quad j=1, \dots, n_t \quad (6.5.1)$$

based on a non-parametric, diffuse prior model [20], where a form of the likelihood function for each measurement innovation is used to determine the probabilities as

$$\alpha_{j,i} = \exp\left(-\frac{1}{2} \boldsymbol{\eta}_{j,i}^T \mathbf{S}_j^{-1} \boldsymbol{\eta}_{j,i}\right), \quad i=1, \dots, n_m, \quad j=1, \dots, n_t \quad (6.5.2)$$

$$\alpha_{j,0} = 2 \cdot n_m \cdot \frac{(\pi)^2 (1 - P_G P_D)}{\sigma P_D}, \quad i=1, \dots, n_m, \quad j=1, \dots, n_t \quad (6.5.3)$$

and where  $P_D$  is the target detection probability,  $\alpha_{j,i}$  is the likelihood function of measurement  $i$ ,  $\alpha_{j,0}$  is the likelihood that no measurement is correct. Likewise, the probability that no measurement considered is correct is given as

$$\beta_{j,0} = \frac{\alpha_{j,0}}{\alpha_{j,0} + \sum_{r=1}^{n_m} \alpha_{j,r}}, \quad j=1, \dots, n_t \quad (6.5.4)$$

where all of the measurement probabilities satisfy the Bayesian constraint that

$$\sum_{i=0}^{n_m} \beta_{j,i} = 1, \quad j=1, \dots, n_t \quad (6.5.5)$$

Note that these parameters are dependent on the total number of measurements and targets and they are calculated for each target separately.

## 6.6 Data Association in the Kalman Filter

Since the approach is still modeled linearly, the continuous time Kalman filter, used in CHAPTER 3, is kept. The estimated states propagate nominally as

$$\dot{\hat{\mathbf{x}}}_j = \begin{pmatrix} \dot{\hat{x}}_j \\ \dot{\hat{y}}_j \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad j = 1, \dots, n_t \quad (6.6.1)$$

The probabilistic data association is incorporated into the Kalman filter through a combined innovation, calculated as

$$\boldsymbol{\eta}_j = \sum_{i=1}^{n_m} \beta_{ji} \boldsymbol{\eta}_{ji}, \quad i = 1, \dots, n_m, \quad j = 1, \dots, n_t \quad (6.6.2)$$

The state estimate measurement update is found to be

$$\dot{\hat{\mathbf{x}}}_j = K_j \boldsymbol{\eta}_j, \quad i = 1, \dots, n_m, \quad j = 1, \dots, n_t \quad (6.6.3)$$

Where the Kalman gain is used in the form

$$K_j = P_j C^T S_j^{-1}, \quad j = 1, \dots, n_t \quad (6.6.4)$$

and the covariance update, expressed in discrete form, accounting for the probabilistic data association, is calculated as

$$P_j = \beta_0 (P_j + Q_j \Delta t) + (1 - \beta_0) (I - K_j C) (P_j + Q_j \Delta t) + K_j \left[ \sum_{i=1}^{N_k} \beta_{ji} \boldsymbol{\eta}_{ji} \boldsymbol{\eta}_{ji}^T - \boldsymbol{\eta}_j \boldsymbol{\eta}_j^T \right] K_j^T \quad (6.6.5)$$



for  $j = 1, \dots, n_t$  .

The first term in the covariance Equation (6.6.5) corresponds to the covariance for no correct measurements, the second is based on a covariance update weighted with the sum of measurement probabilities, and the last term is added based on the covariance of each measurement's innovation squared, weighted with probability, from the combined innovation squared. The term in parentheses containing the matrix  $Q$  is the a priori covariance. These filter equations apply to each target trajectory that is initialized. Filter implementation was performed by using a finite time step, Euler integration approximation with a small time step to avoid discretization errors, as before.

# CHAPTER 7

## MULTI-TARGET GEOLOCATION VALIDATION THROUGH SIMULATION

### 7.1 Simulation Settings

A simulation is performed in MATLAB to demonstrate the effectiveness of the proposed algorithm in distinguishing and geolocating multiple targets. The simulation runs for 10 s with a step size of 0.0667s, corresponding to the video frame rate used in the experiment. The home position (latitude, longitude, and altitude) is set at (28.599859°, -81.196549°, 0m) with the Earth radius of 6,373.2km. The simulated octorotor is hovering at (28.599848°, -81.1965378°, 10m), which is (1.09m, -1.22m, 10m) in the local frame  $L$ . A zero-mean Gaussian white noise with a covariance of 16m<sup>2</sup> is added to the simulated octorotor motion in each direction. A moving average filter with a sliding window of 75 steps is applied to smoothen the noisy octorotor GPS signal. Three simulated diseased leaf targets with a constant height of 0.04m are placed at (28.59984157°, -81.1965374°), (28.5998495°, -81.1965488°), and (28.5998543°, -81.1965227°) latitude and longitude respectively.

The camera scale factors,  $l_x$  and  $l_y$ , are found to be 66.67 pixels per meter in both directions. The targets have corresponding pixel values of (125, 126), (51, 67), and (221, 32) respectively. A random number of measurements from 1 to 11 is generated per target in a random distribution. These disturbance measurements are given a zero-mean Gaussian white noise with covariance of 16pixel<sup>2</sup> in both directions each. The process noise covariance matrix  $Q$  is set as a diagonal matrix with values of 2 m<sup>2</sup>/s. The filter estimates are initialized at the first valid measurement for that target, as mentioned above, and the estimate covariance  $P$  is initialized per target as a diagonal

matrix with  $0.25 \text{ m}^2$  in both the  $x$  and  $y$  direction diagonals. The detection probability is chosen as 1 and a chi-squared parameter of 4 is selected, which corresponds to a gate probability of 0.865.

## 7.2 Simulation Results and Discussion

As shown in Figure 7.2.1, the simulated UAV position is noisy in all three directions and the moving average filter is used to smoothen them, which matches well with the anticipated values. There are shifts displayed in the moving average filter components for the octorotor motion in Figure 7.2.1, especially at the beginning of the filter where the number of values being averaged is low.

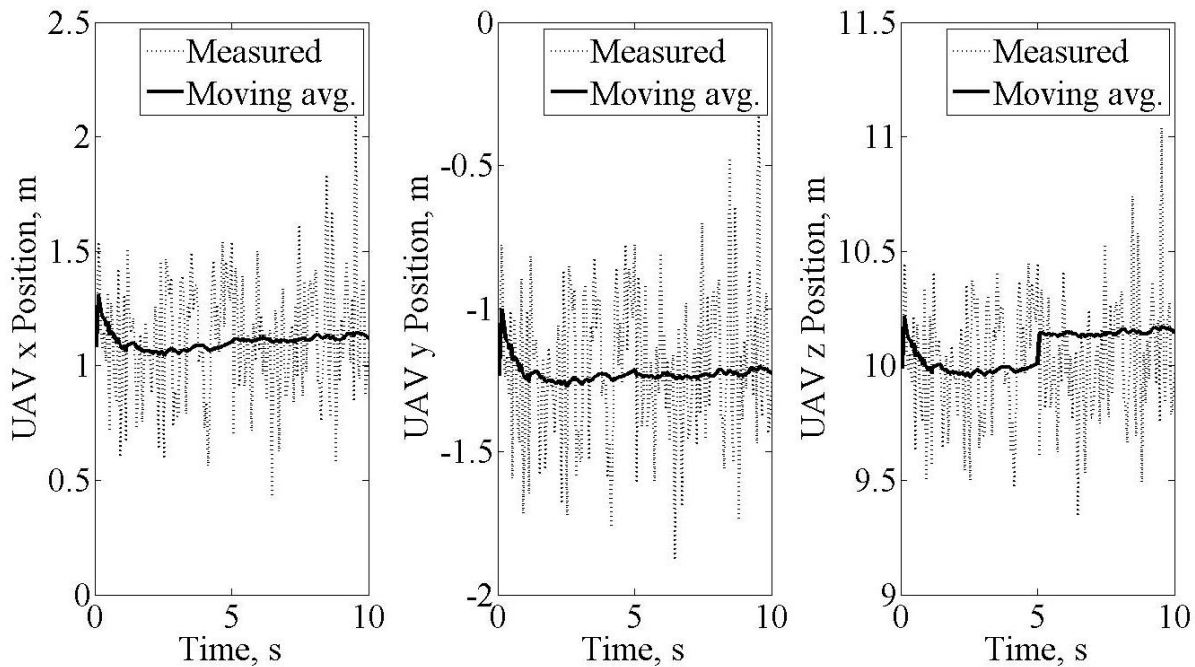


Figure 7.2.1 Simulated octorotor motion along with the moving average filter in each direction

An image frame with the proposed correlation method and data association technique is displayed in Figure 7.2.2 below, where the predicted measurement for each of the three simulated targets is shown in the center of its corresponding validation region. The multiple measurements per region are displayed as well as the boundary points of the region.

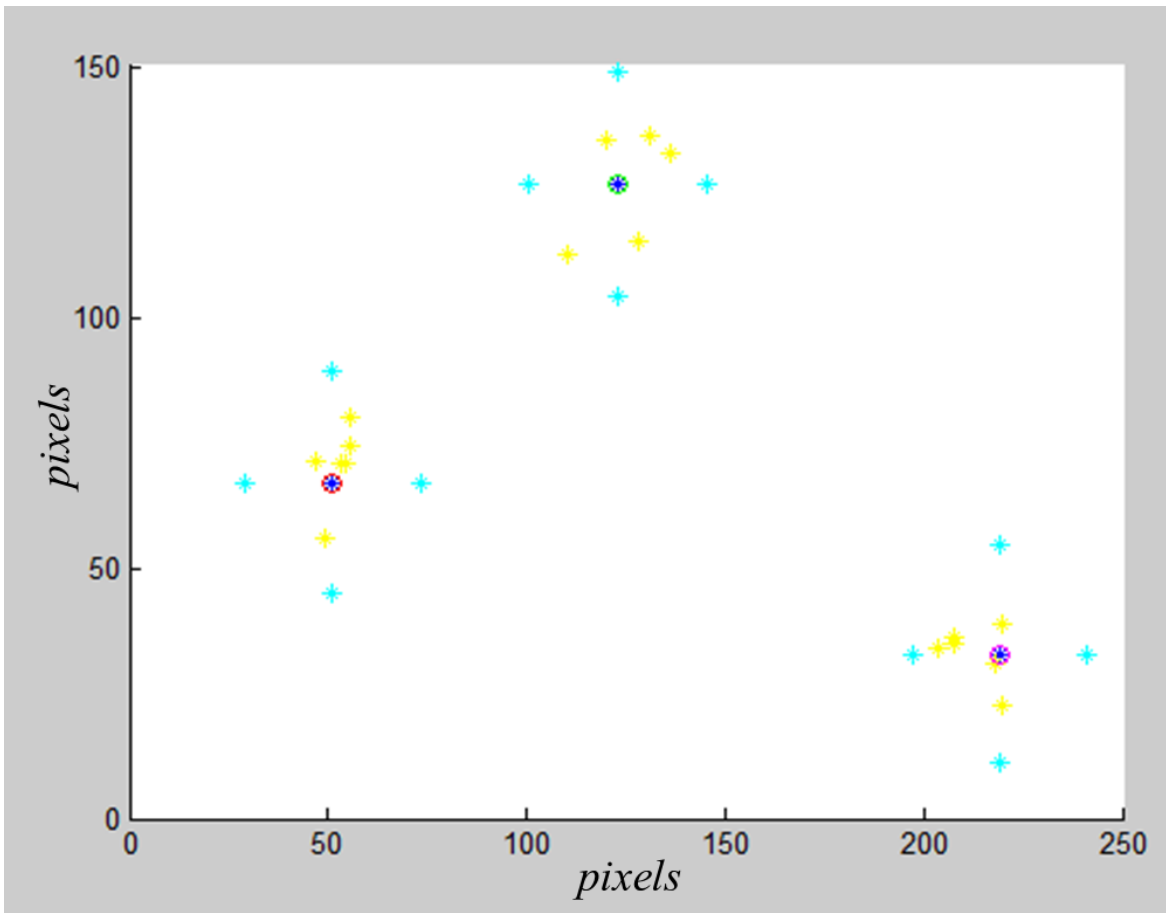


Figure 7.2.2 Measurement correlation and association performed in an image frame

Figure 7.2.3 and Figure 7.2.4 show the estimated and actual pixel values of the three targets. The random measurements per target, placed along a Gaussian distribution, cause fluctuations in the output estimate based on their probability of being the correct measurement. These fluctuations

are bounded, as the measurements all stayed within the validation gate. Similar arguments can be made for the position estimates of the three targets, shown below in Figure 7.2.5 and Figure 7.2.6.

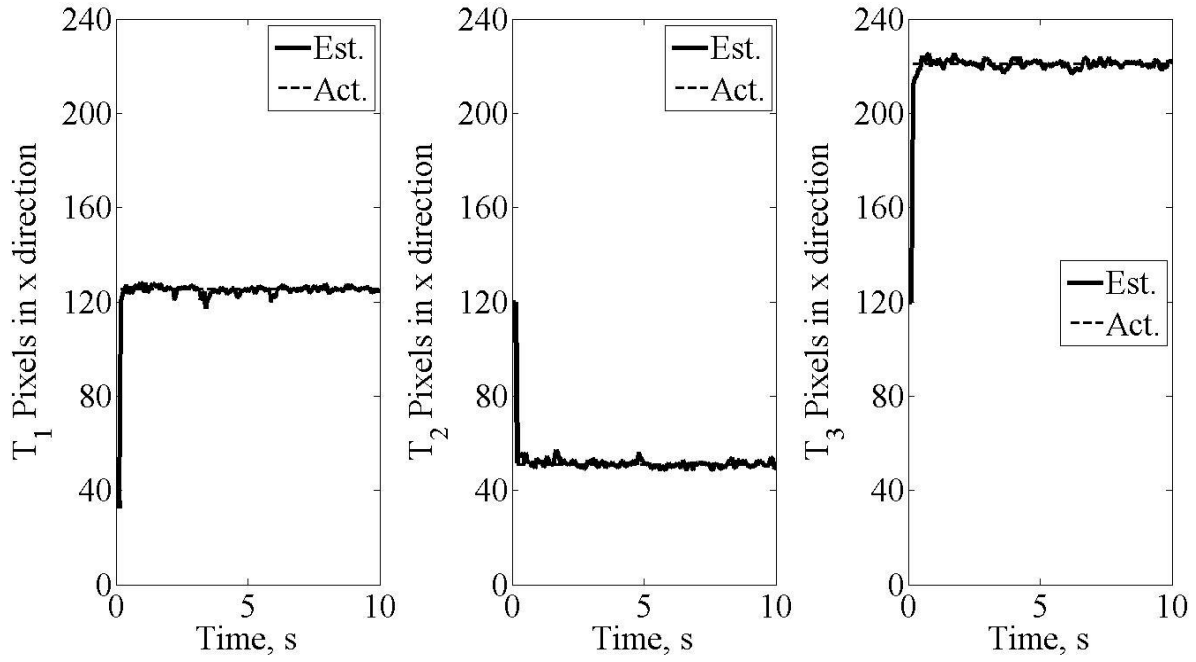


Figure 7.2.3 Actual and estimated output in the  $x$  direction for each target

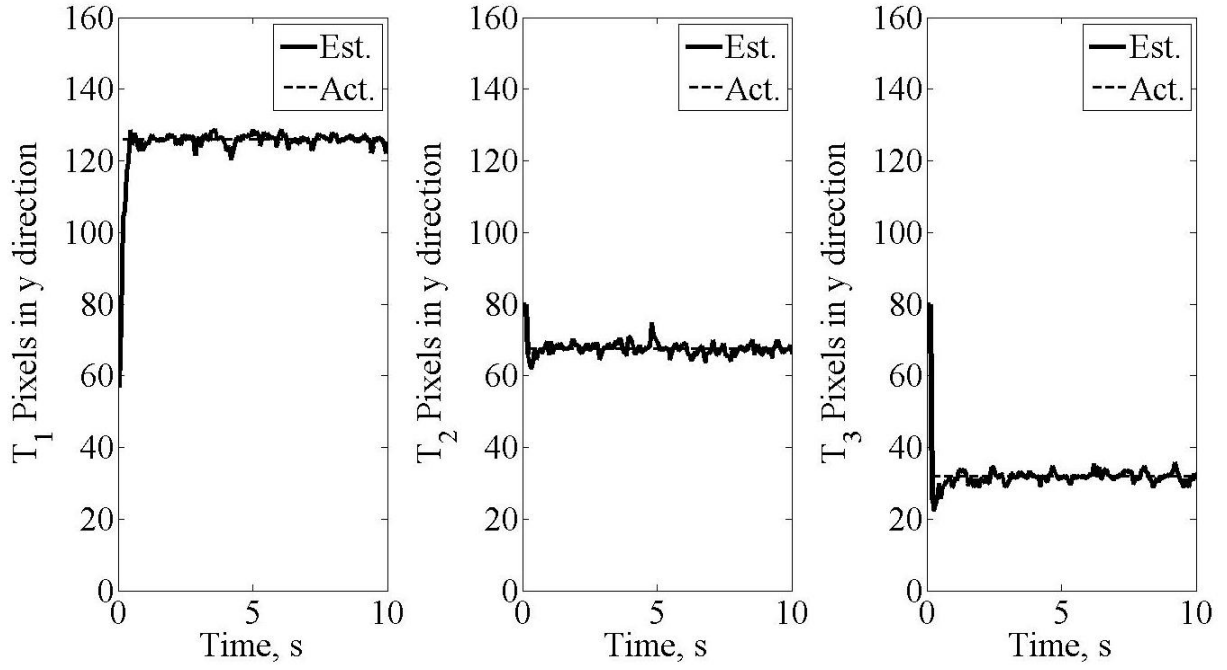


Figure 7.2.4 Actual and estimated output in the y direction for each target

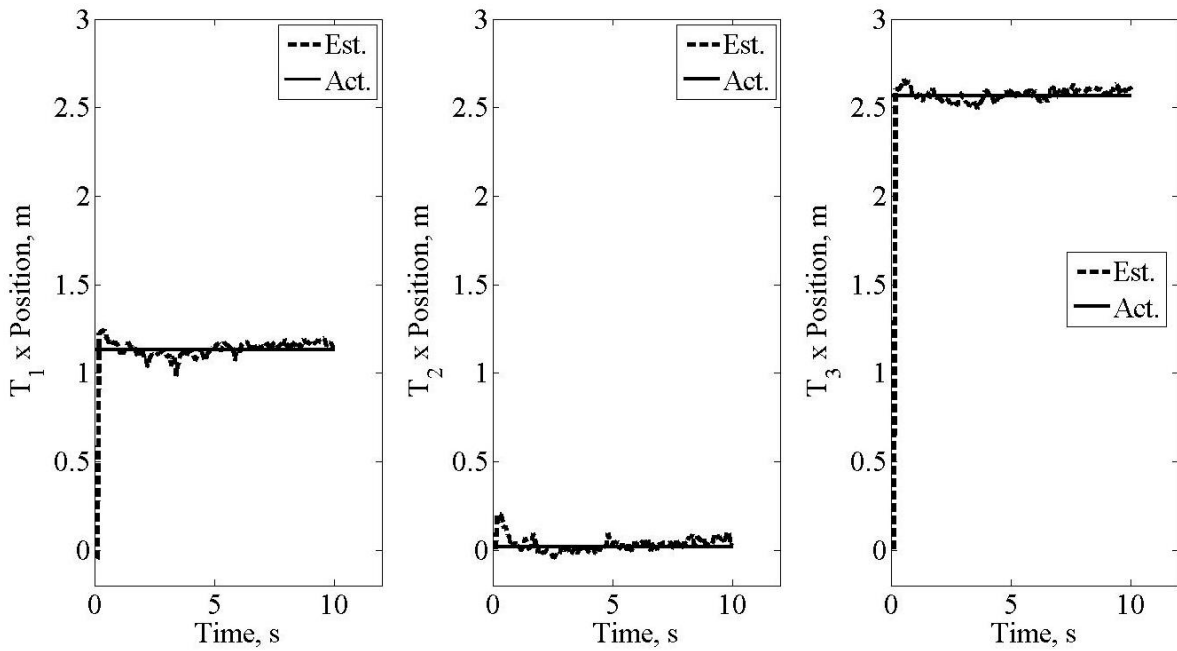


Figure 7.2.5 Actual and estimated position in the x direction for each target

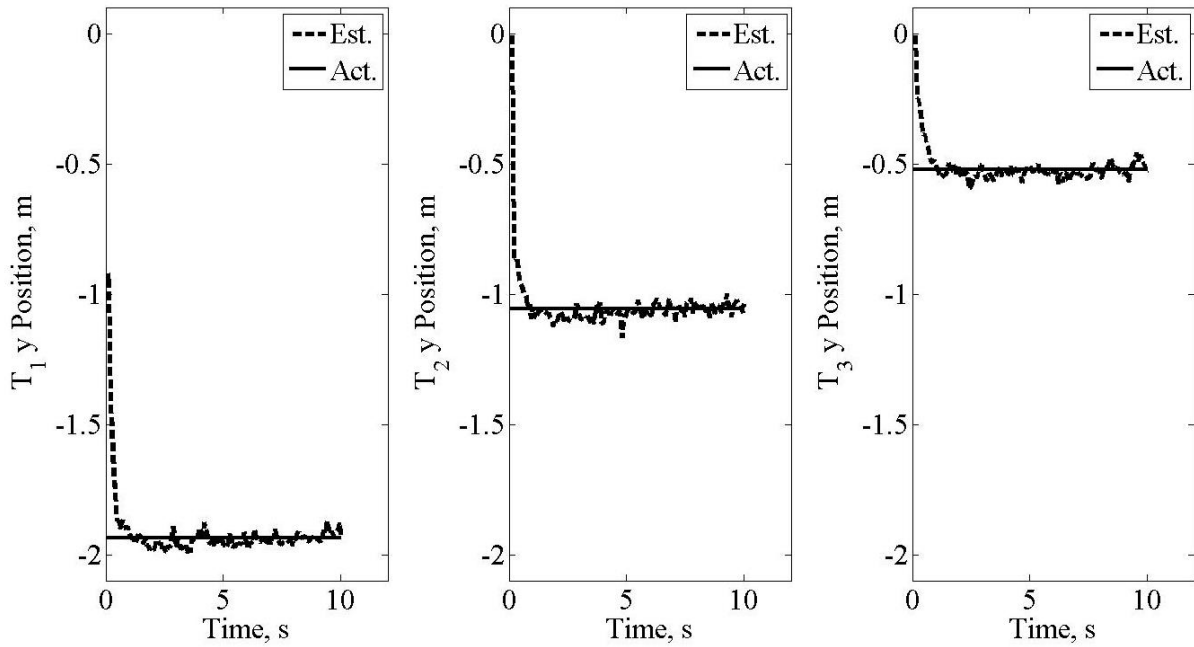


Figure 7.2.6 Actual and estimated position in the y direction for each target

The trace of the state covariance matrix, seen in Figure 7.2.7, is shown to converge to a small constant with minor fluctuations present due to the data association being performed.

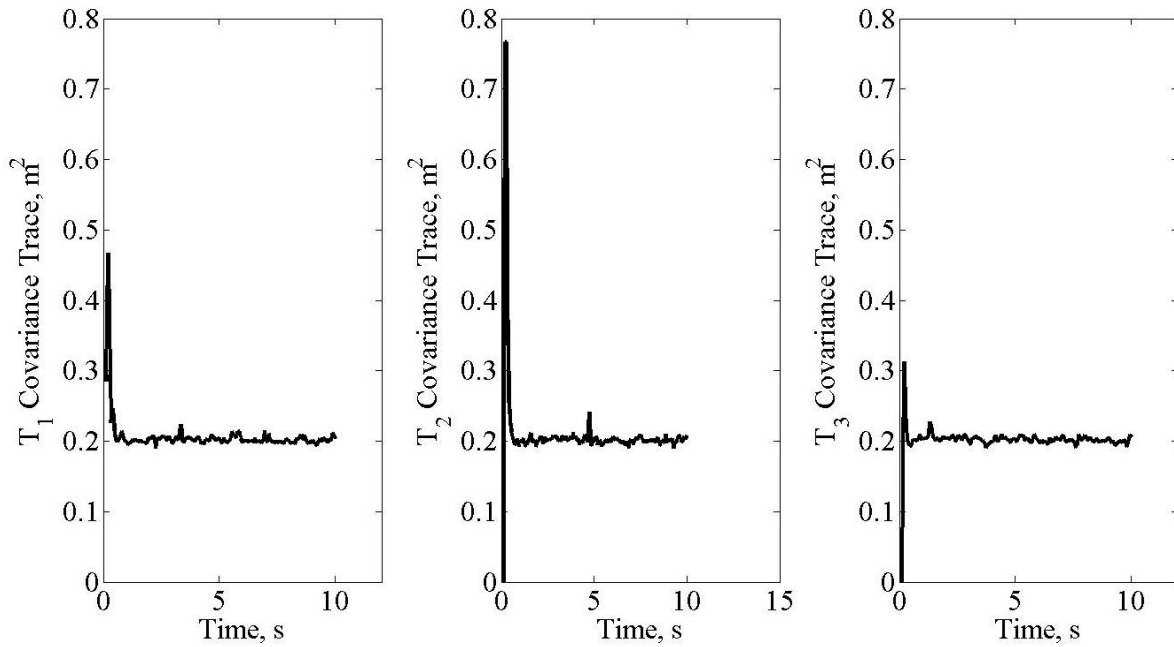


Figure 7.2.7 Trace of the estimate covariance matrix for each target

Overall, as shown in Table 7.2.1, the filter estimated the three targets accurately, and the position error is bounded by 5.4cm, which is much less than half of the width dimension of a typical strawberry row.



Table 7.2.1 Simulation results for actual and estimated target position

	Target 1		Target 2		Target 3	
	<i>x</i>	<i>y</i>	<i>x</i>	<i>y</i>	<i>x</i>	<i>y</i>
Actual position, <i>m</i>	1.133	-1.936	0.020	-1.057	2.569	-0.523
Estimated position, <i>m</i>	1.116	-1.956	0.043	-1.047	2.616	-0.469
Position error, <i>m</i>	0.017	0.020	-0.023	-0.010	-0.047	-0.054
Actual output, <i>pixel</i>	125.2	124.5	51.0	67.5	221.0	31.9
Predicted measurement, <i>pixel</i>	124.7	125.1	50.4	66.2	221.4	32.2
Minimum covariance trace, $m^2$	0.197					
Settling time, <i>s</i>	0.667					
Computational time, <i>s</i>	0.132					

## CHAPTER 8 MULTI-TARGET GEOLOCATION EXPERIMENT

### 8.1 Experiment Settings

Four red targets are placed in close proximity on a grass field and the octocopter is hovering above while recording the video as shown in Figure 8.1.1. A segment of the video is truncated from the raw video. This video is 12.6s long, with 189 frames at a time step of 0.0667s or 15 *fps*, and the image size is (240x150)*pixel*. With the fast gimbal system onboard, the camera is assumed to be well balanced and always in a face-down position.

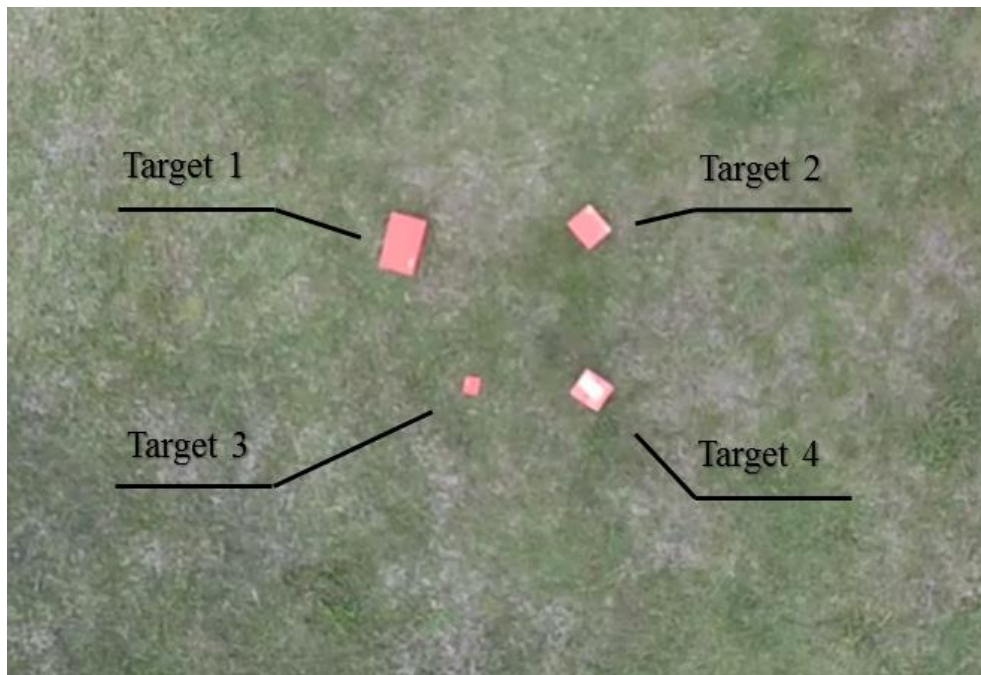


Figure 8.1.1 Video frame of the targets in the field

The video is pre-processed to obtain the target pixel locations. A color threshold that accepts pixels with a red intensity above 150 and a green intensity below 143 is used. Then the pixels are grouped into contours and all resulting contour centroids are considered as possible target measurements. Note that this relaxes the condition used in the previous comparison experiment, where only one contour, with the greatest area, was taken as the measurement. It is worth mentioning that a different identification scheme will need to be implemented when a spectral camera, instead of an RGB camera, is used in disease detection applications.

The settings of home position, Earth radius, plant height, detection probability, chi-squared parameter, and validation gate probability, are the same as those shown in the simulation validation. The octorotor GPS measurements are converted to local frame position with Equation (3.3.1). The average octorotor yaw angle encountered throughout the flight was  $35.36^\circ$ . The gimbal is aligned to the front arm of the octorotor, as before, and perfect performance was assumed. The scale factors are found to be  $75.09\text{pixel}/m$  and  $76.04\text{pixel}/m$  for the range of altitude encountered. The measurement matrix and adjust vector, respectively, are

$$C = \begin{bmatrix} 48.943 & -34.988 \\ -35.432 & -49.564 \end{bmatrix} \frac{\text{pixel}}{m} \quad (8.1.1)$$

and

$$D = \begin{pmatrix} -53.03 \\ -31.36 \end{pmatrix} \text{pixel} \quad (8.1.2)$$

The position of target 2 is determined by identifying an image frame where the target is in the middle of the image, and taking the octorotor GPS latitude and longitude at this frame as the target position. The other target positions are determined relative to target 2 and can be seen in Table 8.2.1 in the experiment results. Due to the proximity of the targets to each other, small values for

the diagonals of covariance matrices  $P_0$ ,  $Q$ , and  $R$  of  $0.01 \text{ m}^2$ ,  $0.05 \text{ m}^2/\text{s}$ , and  $10 \text{ pixel}^2$  respectively, are chosen.

## 8.2 Experiment Results and Discussion

The octorotor position trajectory is plotted first in Figure 8.2.1 below, where it is observed that the octorotor shifted  $0.39\text{m}$ ,  $1.148\text{m}$ , and  $2.11\text{m}$  in each direction, respectively. Because there is a  $35.36^\circ$  yaw angle, the UAV movement effects in the target position and measurement estimations are cross-coupled in direction.

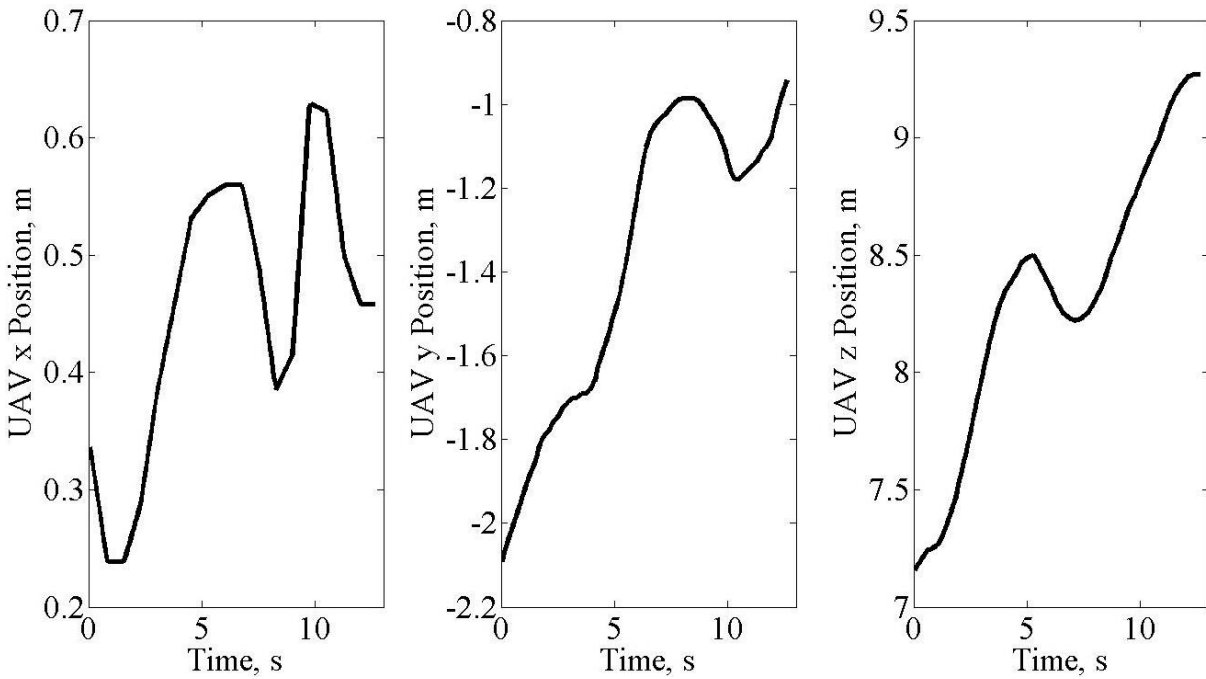


Figure 8.2.1 Octorotor position in all three directions, respectively

An image frame with the measurement correlation and data association being performed on the flight video is provided as Figure 8.2.2. The circles indicate the predicted measurement location for each target in the image, the yellow stars indicate the multiple measurements validated for each target, and the blue stars indicate the boundaries of the validation region for that time step. It will be noted that target 4 (on the bottom right) receives no measurements for a time from 3s5s, in which the estimation trajectory is terminated, then reinitialized when the measurements reappear. This occurs because the red box for target 4 did not satisfy the identification criterion for a time due to lighting. Additionally, multiple false alarm spots further away from the targets satisfy the identification criterion for a time, are estimated, then terminated when the measurements disappear.

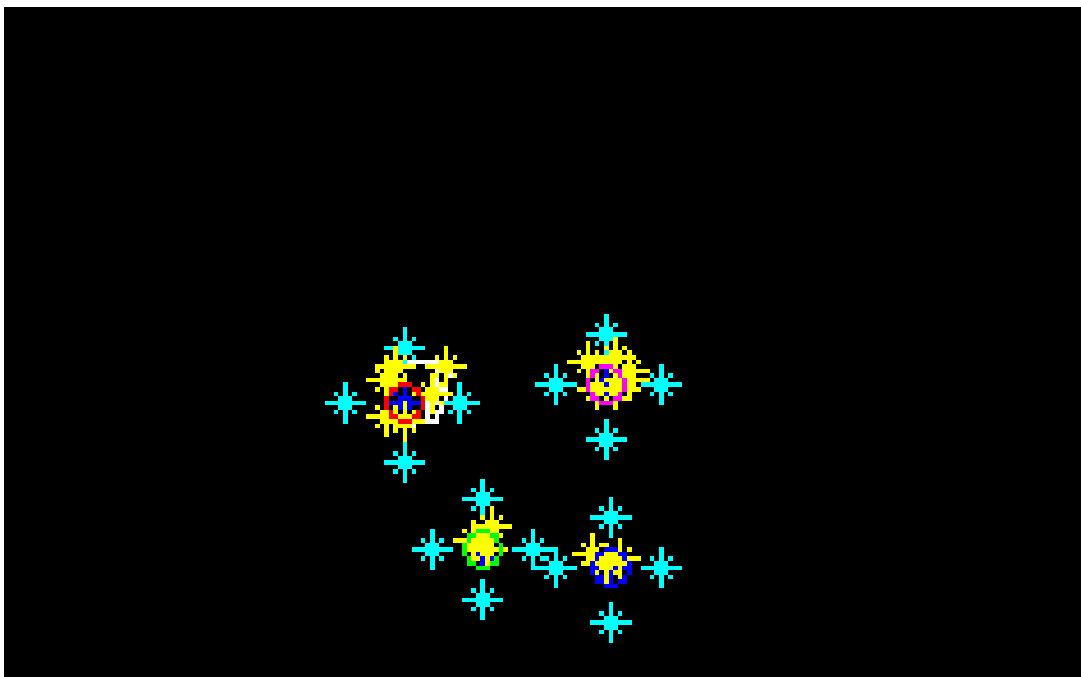


Figure 8.2.2 Image frame of the flight video with correlation and data association approaches applied

The predicted measurements for each of the target estimations appear in the  $x$  and  $y$  directions in Figure 8.2.3 and Figure 8.2.4, respectively. The predicted measurements are shown to have many small, fast fluctuations, due to the considered of multiple measurements in each time step to propagate the predicted measurement for each target. In the  $x$  direction, the larger, slow shifts, due to the octorotor motion in flight, tend to only move the estimate by about  $20\text{pixel}$  per target track. In the  $y$  direction, however, the shifts appear to move the estimated measurements about  $70\text{pixel}$  per track, which correlates with the magnitude of octorotor movement seen above, although cross-coupling is present. The disappearance of target 4 is reflected in the predicted measurement plots.

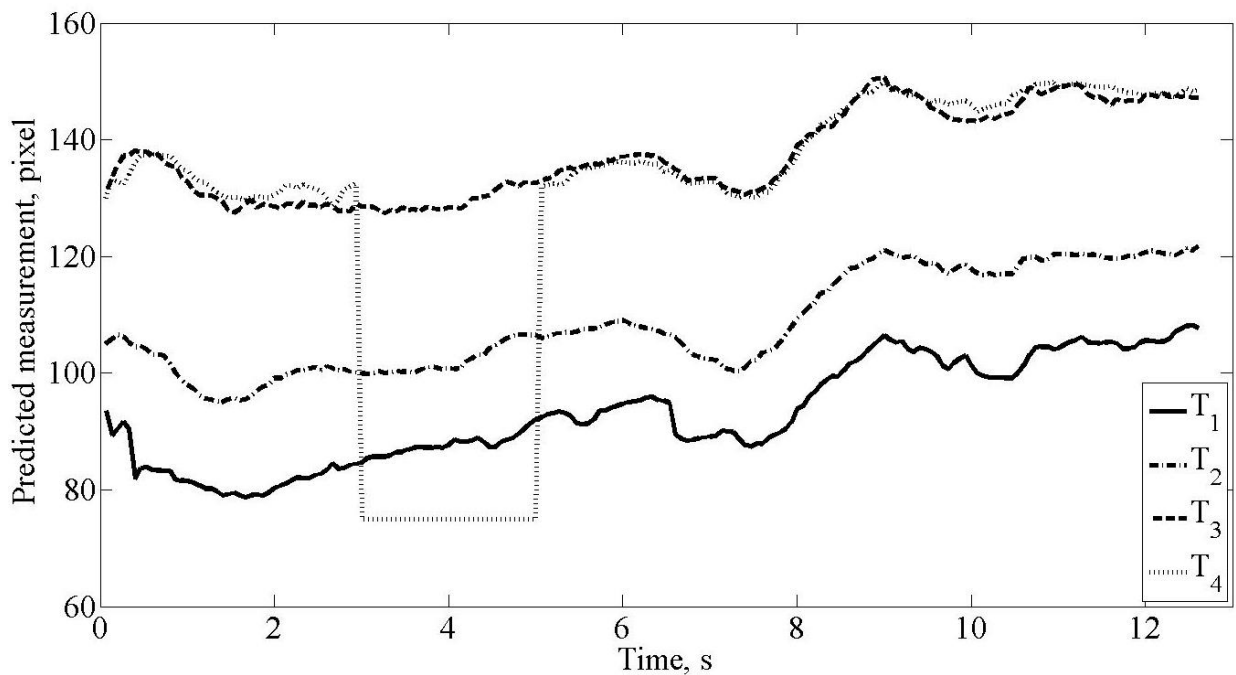


Figure 8.2.3 Predicted measurements in the  $x$  direction for each target

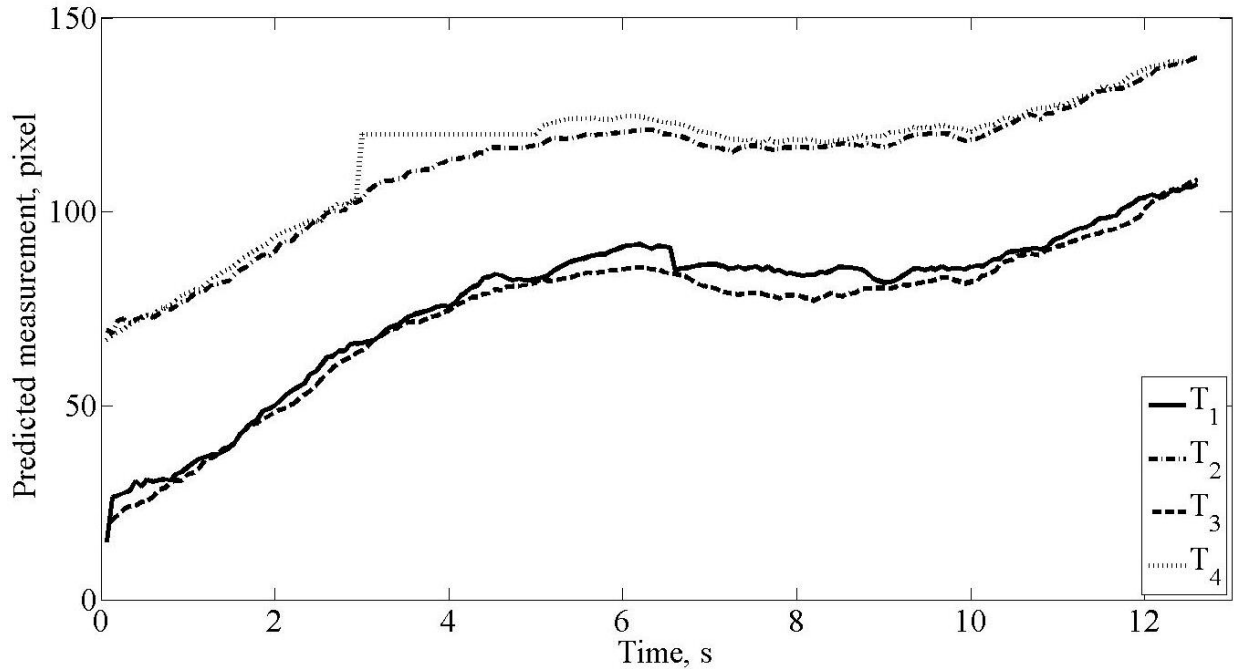


Figure 8.2.4 Predicted measurements in the y direction for each target

The target actual and estimated positions are given in Figure 8.2.5 and Figure 8.2.6 below, for the  $x$  and  $y$  directions, respectively. The position estimate trajectories converged in each direction, within a bound. The maximum error values encountered for each target in each direction is given in Table 8.2.1 below. The greatest error seen in any of the trajectories was  $0.383m$  in the  $x$  direction of the estimate for target 4, which satisfies the goal of a position error less than  $0.5m$ . Therefore, the approach presented in this thesis has demonstrated that it is effective for geolocation of multiple diseased strawberry trees.

The covariance trace plot for each of the target estimations is provided in Figure 8.2.7 below. Each plot is shown to converge to its respective value, shown in Table 8.2.1, albeit with minor fluctuations due to the data association. It should be noted how the trace for target 1 increases at certain points along the trajectory. This is due to measurements appearing outside the validation

gate for the target 1 trajectory, which creates a new target for that step. In the following step, a common measurement is seen between the two trajectories, the two are joined, and their covariance is superimposed, creating this jump in the trace plot from one step to another.

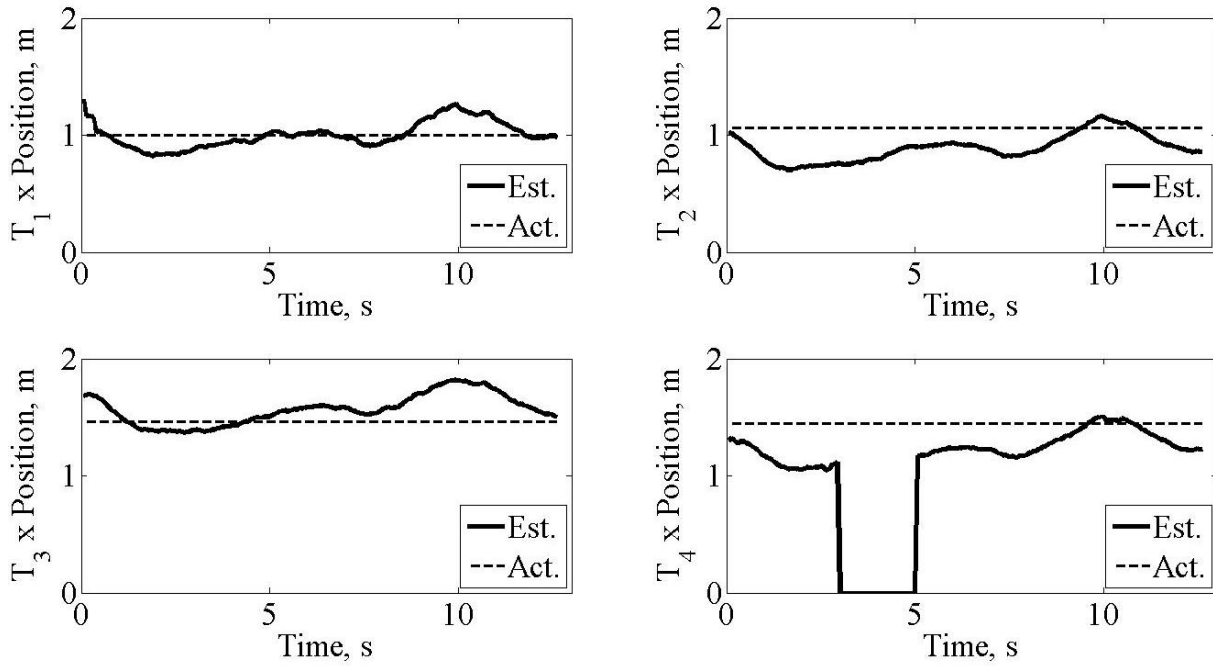


Figure 8.2.5 Actual and estimated position in the x direction for each target, respectively



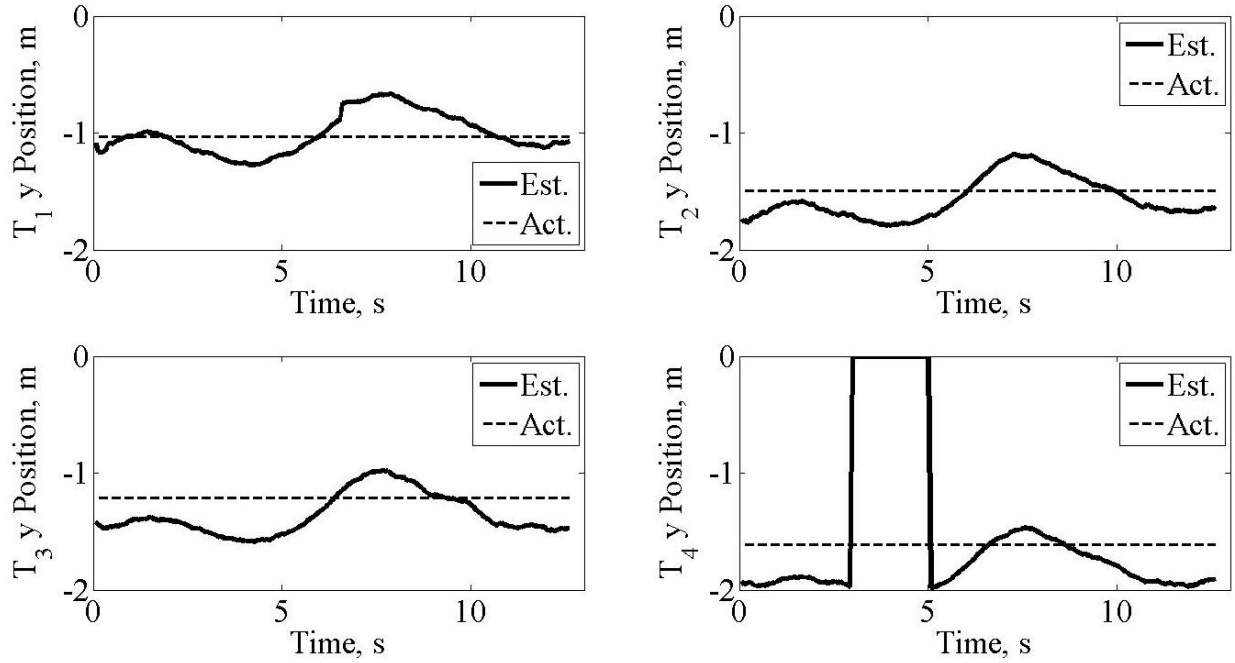


Figure 8.2.6 Actual and estimated position in the y direction for each target, respectively

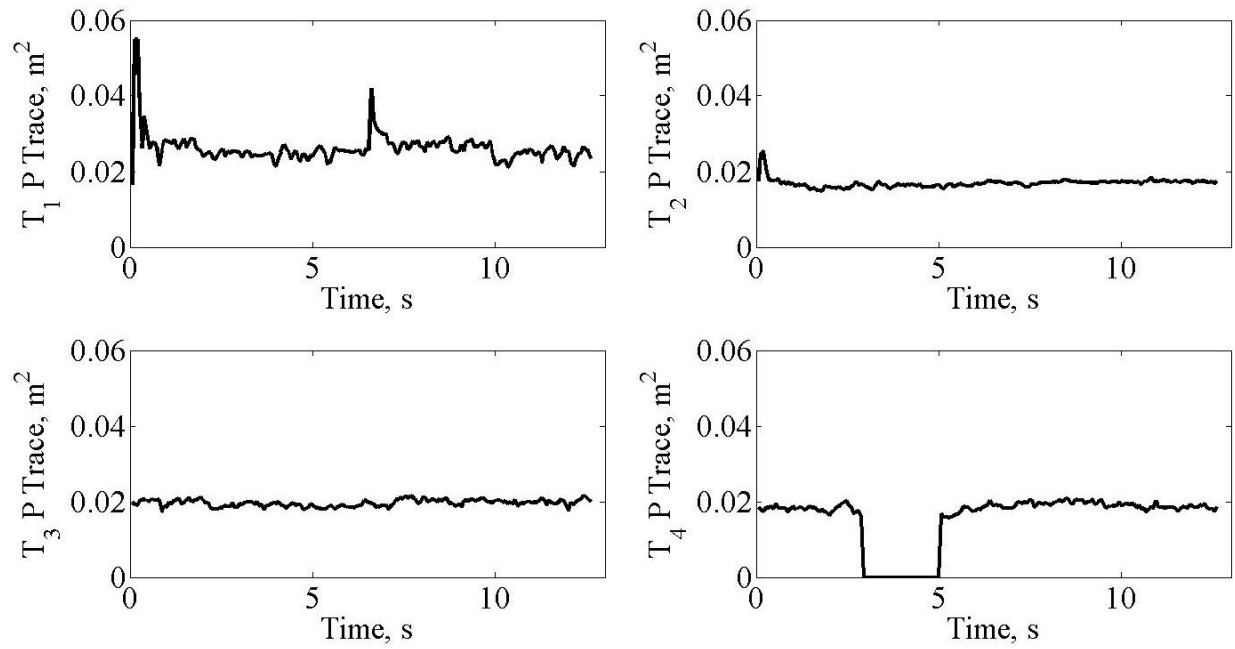


Figure 8.2.7 Estimate covariance trace for each target, respectively

Table 8.2.1 Experiment results for actual and estimated target positions

	Target 1		Target 2		Target 3		Target 4	
	$x$	$y$	$x$	$y$	$x$	$y$	$x$	$y$
Actual position, $m$	0.996	-1.035	1.065	-1.491	1.465	-1.212	1.445	-1.613
Estimated position, $m$	0.959	-1.086	0.850	-1.643	1.505	-1.456	1.216	-1.903
Position error, $m$	-0.037	-0.051	-0.215	-0.152	0.040	-0.244	-0.229	-0.290
Maximum position error, $m$	0.311	0.370	0.360	0.309	0.353	0.377	0.383	0.367
Min. covariance trace, $m^2$	0.022		0.0168		0.0175		0.018	
Computational time, $s$	0.0444							

## **CHAPTER 9 CONCLUSION**

### **9.1 Concluding Remarks**

A custom-designed octorotor UAV for disease detection in strawberry orchards is presented. The requirements used to design the octorotor, along with its overall features and specifications are provided. A geolocation method based on an orthographic projection measurement model is derived, in order to geolocate the diseased region from the images where it is encountered. It is based on assumptions applicable to the disease detection scenario for strawberry plants, and is tailored for use on-board through low computational time. This method is linear and uses a Kalman filter for its position estimation. The proposed method is then compared to nonlinear method using an extended Kalman filter, based on the commonly used perspective projection measurement model, in both simulation and an ad-hoc experiment using real flight data. The two filters are shown to perform well and in similar fashion, however the proposed geolocation method has less error and requires less computational time. The geolocation algorithm was then adjusted and implemented onto a Raspberry Pi 2 board for real-time implementation. This geolocation method will allow the detected disease to be localized in real-time aboard the UAV and in the initial scouting phase, further advancing the disease detection process towards becoming an autonomous process.

The geolocation method is extended to deal with multiple diseased regions. A logic-based method is proposed to perform proper measurement-to-target correlation by analyzing each measurement-to-target relationship based on Chi-squared probability. Then a probabilistic data association technique is used to determine the probability of each validated measurement being

the correct image location for that suspected target being estimated. The measurement is then weighted by its probability and associated into the Kalman filter estimation for its target. As compared to other association approaches, this association method is also less computationally intensive. The proposed multi-target geolocation method was validated through simulation and experiment and was demonstrated to meet the desired position error. Overall, this method will allow for multiple diseased regions to be geolocated simultaneously in the initial scouting phase of the disease detection process.

## 9.2 Future Work

Future work will include accounting for octorotor maneuvers using a maneuvering, multi-target tracking technique. This will allow the method to be used on fixed-wing UAV as well. The proposed correlation approach and probabilistic data association technique will be incorporated onto the Raspberry Pi 2 for real-time multi-target geolocation. An adaptive method for determining the covariance weight matrices  $Q$ ,  $R$ , and  $P_0$ , in real-time will also be developed. A method for incorporating the position estimation into the flight trajectory and control of the UAV will be developed. Testing in real-time with a spectral camera on-board, and with a proper disease identification scheme, in a strawberry orchard setting is desired.

## LIST OF REFERENCES

- [1] Defterli, S. G., Shi, Y., Xu, Y., and Ehsani, R., “Review of Robotic Technology for Strawberry Production,” *Applied Engineering in Agriculture*, American Society of Agricultural and Biological Engineers, Vol. 32, No. 3, 2016, pp. 301-318.
- [2] Costa, F. G. et al, “The Use of Unmanned Aerial Vehicles and Wireless Sensor Network in Agricultural Applications,” *2012 IEEE International Geoscience and Remote Sensing Symposium*, Munich, Germany, pp. 5045-5048, July 22-25, 2012.
- [3] Yang, C., Everitt, J. H., and Bradford, J. M., “Airborne Hyperspectral Imagery and Yield Monitor Data for Estimating Grain Sorghum Yield Variability,” *Transactions of the ASAE*, Vol. 47, No. 2, 2004, pp. 915-924.
- [4] Zaman, B., Jensen, A. M., and McKee, M., “Use of High-Resolution Multispectral Imagery Acquired with an Autonomous Unmanned Aerial Vehicle to Quantify the Spread of an Invasive Wetlands Species,” *2011 IEEE International Geoscience and Remote Sensing Symposium*, Vancouver, BC, pp. 803-806, July 24-29, 2011.
- [5] Xu, Y., Ehsani, R., Kaplan, J., Ahmed, I., Kuzma, W., Orlandi, J., Nehila, K., Waller, K., and Defterli, S. G., “An Octo-Rotor Ground Network for Autonomous Strawberry Disease Detection – Year 1 Status Update,” *2<sup>nd</sup> International Conference on Robotics and Associated High-Technologies and Equipment for Agriculture and Forestry*, Madrid, Spain, pp. 457-466, May 21-23, 2014.
- [6] Ehsani, R. Sankaran, S., Maja, J. M., and Neto, J. C., “Affordable Multi-Rotor Remote Sensing Platform for Applications in Precision Horticulture,” *12<sup>th</sup> International Conference on Precision Agriculture*, Sacramento, CA, July 20-23, 2014.

- [7] Teke, M., Deveci, H. S., Haliloglu, O., Gurbuz, S. Z., and Sakarya, U., "A Short Survey of Hyperspectral Remote Sensing Applications in Agriculture," *6<sup>th</sup> International Conference on Recent Advances in Space Technologies*, Istanbul, Turkey, IEEE, pp. 171-176, June 12-14, 2013.
- [8] Berni, J. A. J., Zarco-Tejada, P. J., Suarez, L., and Fereres, E., "Thermal and Narrowband Multispectral Remote Sensing for Vegetation Monitoring from an Unmanned Aerial Vehicle," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 47, No. 3, 2009, pp. 722-738.
- [9] Gomez-Candon, D., Lopez-Ganados, F., Caballero-Novella, J. J., Gomez-Casero, M., Jurado-Exposito, M., Garcia-Torres, L., "Geo-Referencing Remote Images for Precision Agriculture using Artificial Terrestrial Targets," *Precision Agriculture*, Springer Science & Business Media, Vol. 12, No. 6, 2011, pp. 876-891.
- [10] Pan, L., Zhang, W., Zhu, N., Mao, S., and Tu, K., "Early Detection and Classification of Pathogenic Fungal Disease in Post-Harvest Strawberry Fruit by Electronic Nose and Gas Chromatography-Mass Spectrometry," *Food Research International*, Vol. 62, 2014, pp. 162-168.
- [11] Xiang, H., and Tian, L., "Method for Automatic Georeferencing Aerial Remote Sensing (RS) Images from an Unmanned Aerial Vehicle (UAV) Platform," *Biosystems Engineering*, Vol. 108, No. 2, 2011, pp. 104-113.
- [12] Garcia, C., and Xu, Y., "Target Geolocation for Disease Detection Applications via an Octorotor," Accepted in the *5<sup>th</sup> IFAC Conference on Sensing, Control and Automation for Agriculture – AgriControl 2016*, Seattle, WA, August 14-17, 2016.
- [13] Savage, C. O. and La Scala, B. F., "Accurate Target Geolocation using Cooperative Observers," *2007 IEEE Information, Decision, and Control Conference*, IEEE, Adelaide, Australia, pp. 248-253, February 12-14, 2007.

- [14] Wu, Y. A., "EO Target Geolocation Determination," *Proceedings of the 34<sup>th</sup> IEEE Conference on Decision and Control*, IEEE, New Orleans, LA, pp. 2766-2771, December 13-15, 1995.
- [15] Campbell, M. E. and Wheeler, M., "A Vision Based Geolocation Tracking System for UAV's," *Proceedings of the 2006 AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, CO, pp. 1942-1959, August 21-22, 2006.
- [16] Sohn, S., Lee, B., Kim, J., and Kee, C., "Vision-Based Real-Time Target Localization for Single-Antenna GPS-Guided UAV," *IEEE Transactions on Aerospace and Electronic Systems*, Vol 44, No. 4, 2008, pp. 1391-1401.
- [17] Szeliski, R., *Computer Vision: Algorithms and Applications*, Springer Science & Business Media, New York, 2010. Chapters 1 and 2.
- [18] Sawhney, H. S. and Hanson, A. R., "Identification and 3D Description of 'Shallow' Environmental Structure in a Sequence of Images," *Proceedings of the 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, Maui, Hawaii, pp. 179-185, June 3-6, 1991.
- [19] Simon D., *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*, John Wiley & Sons, Hoboken, NJ, 2006. Chapters 4 and 8
- [20] Bar-Shalom, Y. and Fortmann, T. E., *Tracking and Data Association*, Mathematics in Science and Engineering, Vol. 179, Academic Press, San Diego, CA, 1988. Chapters 6 and 9.
- [21] Bar-Shalom, Y., "Tracking Methods in a Multitarget Environment," *IEEE Transactions on Automatic Control*, Vol. 23, No. 4, August, 1978, IEEE, pp. 618-626.
- [22] Colegrove, S. B. and Davey, S. J., "On using Nearest Neighbours with the Probabilistic Data Association Filter," *The Record of the IEEE 2000 International Radar Conference*, Alexandria, VA, pp. 53-58, May 7-12, 2000.

- [23] Bar-Shalom, Y., Daum, F., and Huang, J., “The Probabilistic Data Association Filter: Estimation in the Presence of Measurement Origin Uncertainty,” *IEEE Control Systems Magazine*, Vol 29, No. 6, 2009, pp. 82-100.
- [24] Li, N., Xu, Y., Basset, G., and Fitz-Coy, N. G., “Tracking the Trajectory of Space Debris in Close Proximity via a Vision-Based Method,” *Journal of Aerospace Engineering*, Vol. 27, No. 2, 2014, pp. 238-248.
- [25] Oh, S. and Sastry, S., “An Efficient Algorithm for Tracking Multiple Maneuvering Targets,” *Proceedings of the 44<sup>th</sup> IEEE Conference on Decision and Control and 2005 European Control Conference*, Seville, Spain, pp. 4010-4015, December 12-15, 2005.
- [26] Yechout, T.R., Morris, S. L., Bossert, D. E., and Hallgren, W. F., *Introduction to Aircraft Flight Mechanics: Performance, Static Stability, Dynamic Stability, and Classical Feedback Control*, Ed. 1, AIAA Education Series, AIAA, New York, 2003. Chapters 3 and 4