2021-05-27

Small-Target Detection and Observation with Vision-Enabled Fixed-Wing Unmanned Aircraft Systems

Hayden Matthew Morgan
Brigham Young University

Small-Target Detection and Observation with Vision-Enabled Fixed-Wing Unmanned Aircraft Systems

Hayden Matthew Morgan

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Randal W. Beard, Chair
Timothy W. McLain
Cameron K. Peterson

Department of Electrical and Computer Engineering

Brigham Young University

ABSTRACT

Small-Target Detection and Observation with Vision-Enabled
Fixed-Wing Unmanned Aircraft Systems

Hayden Matthew Morgan
Department of Electrical and Computer Engineering, BYU
Master of Science

This thesis focuses on vision-based detection and observation of small, slow-moving targets using a gimballed fixed-wing unmanned aircraft system (UAS). Generally, visual tracking algorithms are tuned to detect motion of relatively large objects in the scene with noticeably significant motion; therefore, applications such as high-altitude visual searches for human motion often ignore target motion as noise. Furthermore, after a target is identified, arbitrary maneuvers for transitioning to overhead orbits for better observation may result in temporary or permanent loss of target visibility.

We present guidelines for tuning parameters of the Visual Multiple Target Tracking (Visual MTT) algorithm to enhance its detection capabilities for very small, slow-moving targets in high-resolution images. We show that the tuning approach is able to detect walking motion of a human described by $10-15$ pixels from high altitudes.

An algorithm is then presented for defining rotational bounds on the controllable degrees of freedom of an aircraft and gimballed camera system for maintaining visibility of a known ground target. Critical rotations associated with the fastest loss or acquisition of target visibility are also defined. The accuracy of these bounds are demonstrated in simulation and simple applications of the algorithm are described for UAS.

We also present a path planning and control framework for defining and following both dynamically and visually feasibly transition trajectories from an arbitrary point to an orbit over a known target for further observation. We demonstrate the effectiveness of this framework in maintaining constant target visibility while transitioning to the intended orbit as well as in transitioning to a lower altitude orbit for more detailed visual analysis of the intended target.

Keywords: unmanned aircraft systems, fixed-wing UAS, vision-constrained control, computer vision, constrained path planning, autonomous UAS flight, visual target tracking, small target tracking

ACKNOWLEDGMENTS

I consider myself incredibly fortunate for the opportunity I have had to pursue a master's degree at Brigham Young University and am particularly grateful for the coveted opportunity I had to do so within the academically rich environment of the MAGICC Lab. I am sincerely grateful for the access I had to exceptional classes, phenomenal professors, and extraordinary peers in helping me mature intellectually, spiritually, and professionally.

I am deeply thankful for my advisor Dr. Randy Beard for his seemingly endless capacity to uplift, encourage, and educate in the midst of seemingly insurmountable responsibilities. His example has distilled a great sense of determination and fortitude within me that I hope to instill upon those with whom I will associate. His mentorship in isolating, dissecting, and solving incredibly difficult problems has become foundational for me.

I would also like to thank the many incredible friends and colleagues I was privileged enough to work with over the course of my experience. In particular, I wish to thank the many mentors and friends I interacted with in the MAGICC Lab. They not only provided perspective, understanding, and insight but also helped create an open environment of friendship, humor, and comradery that I will greatly miss.

Finally, surpassing all else, I am extraordinarily grateful for the endless support of my wife Kelcie in encouraging me to pursue my passion in this field. Her continuous selflessness, compassion, encouragement, kindness, service and patience have helped make my dreams become my reality. Thanks for letting me play with drones.

TABLE OF CONTENTS

[1]This work has been accepted for publication in the 2021 International Conference on Unmanned Aircraft Systems (ICUAS)

[2]This work has been accepted for publication in the 2021 International Conference on Unmanned Aircraft Systems (ICUAS)

[3]This work is being submitted for publication in the 2021 AIAA Journal of Guidance, Control, and Dynamics (JGCD)

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

1.1 Motivation

Target detection and observation missions have increasingly sought to utilize unmanned aircraft systems (UAS) equipped with a gimballed camera system for both autonomous and semi-autonomous missions. When deployed at high altitudes, these systems are capable of covering large search areas at notably faster rates than human search teams and additionally eliminating risk in hostile environments. However, the coverage and security afforded by high-altitude search teams encounter difficulties in detecting and observing small ground targets due to their minimal pixel footprint and relatively slow motion characteristics. Additionally, transitioning a UAS from a search maneuver to an orbiting observation maneuver introduces greater risk for losing sight of a small ground target of interest. Without sufficiently defined vision-based trajectory constraints, tracked targets may escape UAS visibility long enough to evade further detection and localization. The interaction between gimbal constraints and flight dynamics is particularly challenging for fixed-wing UAS as opposed to multirotors due to fewer degrees of freedom in the dynamics.

1.2 Previous Work

The problem of visual target tracking has been treated in a variety of computer vision works for many different applications [1], [2]. The objective of small, slow target detection and tracking has important applications in a wealth of fields such as high-altitude traffic management, wildlife detection, security surveillance, border patrol, and search and rescue. In [3], the Visual Multiple Target Tracking (Visual MTT) algorithm is introduced as a front-end interface for providing data points to the Recursive RANSAC (R-RANSAC) algorithm [4] and shows promising real-time performance capabilities. In [5], the Visual MTT algorithm is tuned for tracking targets for applications involving the descent of aerial vehicles. However, no work has been done to tune

the Visual MTT algorithm for detection and tracking of small, slow-moving targets. In [1], [2], [6]–[8], a variety of additional approaches are developed for multiple object tracking from vision data with varying methodologies for track initialization, track propagation, and object detection and correlation. The performance of these tracking methodologies vary by intended application and performance criteria; however, the objective of very small, slow-moving target detection is generally overlooked.

Defining bounds on constrained vision systems has been an area of interest for many authors. Some researchers have sought to identify gimbal or UAS orientation limits for keeping a target within the field-of-view. One author defined limits on target azimuth and elevation angles relative to the optical axis of an inertially-facing camera [9]; however, the approach requires that the camera be statically mounted so that the roll and pitch axes of the UAS are normal to the span of the camera's longitudinal and lateral fields-of-view respectively. In [10], a limit on the UAS roll angle is defined to maintain a road within the field-of-view by projecting the set of camera pixels onto the observation plane. The pixel projection with the shortest distance from a street point along the pixel translation path produced by rolling motion is used to parameterize the UAS roll angle limits. This approach is once again limited exclusively to static cameras pointed along the UAS z-axis and only constrains the roll axis. The authors in [11] develop a method for determining whether a target exists in the field-of-view of a camera; however, no approach is given for defining allowable motion constraints for maintaining or acquiring target visibility. In applications where a gimbal is needed, these methods define neither the UAS nor the gimbal limits needed to constrain vehicle maneuvers for keeping targets within the field-of-view.

Planning trajectories under system constraints has been a growing field of interest over the last few decades. Many works have developed approaches for controlling a UAS so that vision constraints are satisfied. In [12], a reactive control law successfully guides a fixed-wing aircraft to land on a linear runway while maintaining runway visibility. However, the method in [12] is designed specifically for tracking linear structures rather than generalized targets. Another approach is considered in [9] for maintaining a moving target in the camera field-of-view (FOV) with wind disturbances by controlling and constraining the UAS roll angle toward steady orbits. This approach is effective for transitioning to constant orbits about stationary targets, but was not designed to ensure visibility for gimbal angle limits commonly found on inexpensive off-the-shelf

gimbals. The authors in [13] use game theoretic and stochastic optimal control for maintaining a moving, stochastic point target within a constrained FOV. The results of [13] show successful reactive control of a fixed-wing aircraft in following and maintaining line-of-sight of a non-holonomic ground vehicle; however, the control design does not attempt to predict future trajectory points where target visibility may be dynamically unattainable. By proving that a UAS system is differentially flat, the authors of [14]–[20] develop frameworks for successfully planning dynamically feasible trajectories for UAS. However, these works do not account for feasibility of gimbal states or target visibility throughout the trajectory in their flatness models.

This thesis resolves the issue of vision-based multiple target detection for small, slow-moving targets by presenting guidelines and sample tuning parameters in Chapter 2 for the Visual MTT algorithm that enhances its detection capabilities for these applications. To bridge the gap of defining rotational bounds for camera systems with field-of-view limitations and a variety of available degrees of freedom, we present a method for determining both the rotational bounds along controllable axes in Chapter 3. The challenge of defining trajectories that are both dynamically and visually feasible for maintaining continuous target visibility is enabled by the results of Chapter 4 examined for the case trajectories transitioning to an orbit over the observed target in Chapter 5.

1.3 Summary of Contributions

The contributions of this thesis are listed below.

- This thesis introduces guidelines and tuning parameters for tracking very small, slow-moving targets in high-resolution images with the Visual MTT algorithm and demonstrates its effectiveness on real flight test data.
- It derives an algorithm for determining the rotational limits of gimballed UAS tracking a known target.
- It defines an algorithm for determining the most critical UAS or gimbal rotation resulting in the fastest loss or acquisition of target visibility.
- It introduces a novel differential flatness model and control framework for gimballed fixed-wing aircraft allowing trajectory tracking under vision constraints.

- It presents a path planning framework for feasibly transitioning to an overhead orbit over a target while maintaining constant target visibility.

1.4 Thesis Organization

This thesis focuses on the significant components of small target detection and observation under dynamic and visual constraints. Chapter 2 summarizes the Visual MTT algorithm used for visual target tracking and discusses guidelines for tuning parameters to enhance the performance of the system in detecting and tracking very small, slow-moving targets. The effectiveness of the guidelines is demonstrated by successful tracking of walking human targets from a high-altitude aircraft equipped with a camera.

Chapters 3 to 5 are original works submitted for publication in various conferences and journals over the course of this master's program. In Chapter 3, the rotational constraints of a UAS observing a target from a gimballed camera system are considered. A novel algorithm for determining the rotational limits of the system along its available rotational degrees of freedom is derived and its accuracy is demonstrated in simulation.

Chapter 4 derives a differential flatness model for a gimballed fixed-wing aircraft which can be utilized in controlling an aircraft to a desired trajectory while providing a direct mathematical mapping from the trajectory to the inputs required to track it. The effectiveness of the proposed control model in tracking a desired trajectory and predicting required inputs is demonstrated in a simulation study.

Chapter 5 constructs a path planning framework for transitioning a gimballed fixed-wing UAS from an arbitrary starting point to an orbit over a known target as well as a framework for transitioning between orbits of different altitudes. It is shown that the differential flatness model derived in Chapter 4 can be used to select path parameters of these transition trajectory models that account for both dynamic constraints and gimbal constraints of the system in maintaining constant target visibility throughout. The effectiveness of this framework is demonstrated in a ROS/Gazebo simulation.

Finally, Chapter 6 concludes with a summary of the various topics discussed in this thesis as well as a discussion of future research directions.

CHAPTER 2. SMALL TARGET TRACKING WITH VISUAL MTT

2.1 Introduction

For search and observe applications of vision-enabled UAS, detection and tracking of targets from a live camera feed is fundamental in identifying and localizing targets of interest. Current target detection and tracking techniques encounter a variety of issues while attempting to track relatively small targets. Small targets appear as either too small to be considered distinguishable features by standard feature-detection algorithms or so slow in relative pixel motion that the object to be tracked is explained sufficiently by the homography estimation used to propagate correlated visual features between frames. By properly tuning detection and tracking parameters in the visual multiple target tracking (visual MTT) algorithm [3], we are able to track objects defined by very few pixels.

2.2 Visual Multiple Target Tracking Overview

The visual MTT algorithm [3] functions as a front end interface for providing data points to the recursive RANSAC (R-RANSAC) algorithm [4] for model extraction. Its high-level purpose is to take in sequential images, extract features of interest, pass feature data points to R-RANSAC, receive R-RANSAC tracks, and republish images with overlaid markers denoting valid tracks. The most fundamental role of visual MTT is effectively extracting feature points describing targets of interest from sequential images. This is performed in three phases:

1. Feature Management (LKT Tracker): Images are passed through a standard LKT optical flow algorithm which identifies all notable visual features in both frames that correlate with one another.

2. Transform Management (Simple Homography): Correlated feature-pairs are used to estimate the homography transformation matrix describing a rotational and translational motion

estimate of the features between images assuming features exist on a locally flat observation plane.

3. Measurement Management (Feature Motion): The algorithm approximates the motion of the features that do not align sufficiently after transformation by the estimated homography and extracts feature points that have moved within a user-defined distance from each other. All points with these motion characteristics are then passed to R-RANSAC as potential moving target points.

The high-level visual MTT workflow is summarized visually in Figure 2.1.

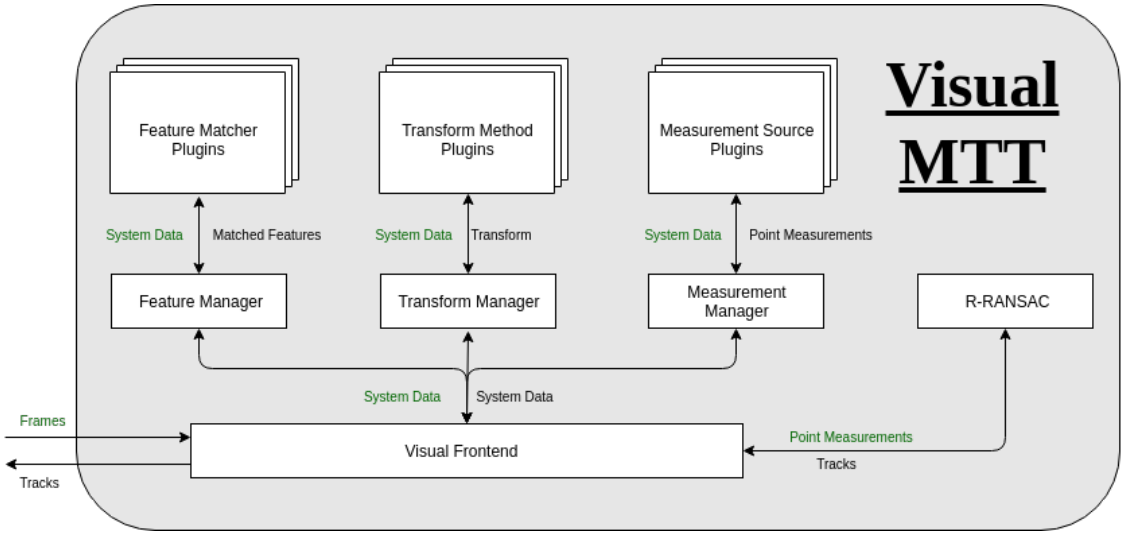


Figure 2.1: Visual MTT algorithm block diagram

The tracks generated by visual MTT are defined in terms of normalized image plane coordinates of the form

$$\varepsilon = [u, v, 1]^\top. \tag{2.1}$$

2.3 Parameter Tuning for Small Target Tracking

Visual MTT was designed with a variety of adjustable parameters to allow application-specific tuning for robust performance in various scenarios. Tuning parameters of interest for small target detection for both the visual front end and R-RANSAC portions of the visual MTT workflow include:

Visual Front End

- **Frame stride:** Frame count between frames passed to visual MTT (e.g. a frame stride of 2 would process every other image)
- **Resize scale:** Resolution reduction percentage of the incoming frame before processing in visual MTT
- **LKT max features:** Maximum features allowed for consideration in LKT optical flow algorithm
- **Minimum feature velocity:** Minimum velocity bound between feature-pairs after homography transformation
- **Maximum feature velocity:** Maximum velocity bound between feature-pairs after homography transformation

R-RANSAC

- **Measurement window:** Maximum size of consecutive data point sets considered for defining tracks
- **Lifetime threshold:** Minimum number of measurements before track initialization allowed
- **Max CMD:** Maximum number of continuously missed detections (CMDs) in the inlier region before track removal
- **Inlier threshold:** Size of region defining acceptable deviation from estimated track path for incoming feature points

Tuning visual MTT to work for relatively small, slow-moving target tracking involves a balance between decreasing computational cost and increasing algorithmic accuracy. For small, slow-moving target tracking, the maximum and minimum feature velocities are tuned to within the range of anticipated target motion observed at the intended altitude for the provided camera resolution. Lowering the minimum velocity bound too much may result in noisy false detections from feature-pair variation caused by homography estimation inaccuracies. Increasing the maximum

velocity bound too much introduces detections from inaccurately correlated features on distant parts of the image (e.g. similar looking bushes or similar corners of dashed street lines).

The resize scale of the image also plays a significant role in small target detection. In theory, it is beneficial for the accuracy of the algorithm to reduce the resolution as little as possible to prevent the loss of small features; however, in practice, too little of a reduction could significantly increase the computation time required for proper evaluation of high-resolution images resulting in dropped frames and, consequently, jumpy feature motion between processed frames. Thus, proper tuning requires balancing justifiable feature loss with computational cost. This balance can be greatly affected by the computational performance capabilities of the host machine. Additionally, the frame stride can be adjusted to reduce computational load of higher resolution images. However, if image motion is too great between frames the LKT tracker may fail to correlate features correctly and produce a poor homography estimate.

R-RANSAC can be tuned for small target tracking by increasing the window of measurements considered for track creation to increase the odds of picking up potentially noisy detections of small target features. If camera motion is rapid enough that targets only briefly enter the camera field-of-view, the lifetime threshold should be reduced to increase the likelihood of detection over a short window. If the anticipated target does not have easily-detectable features, the max CMD limit can be increased to allow model survival in the midst of intermittent detections. The inlier threshold should be large enough to account for motion not described by the selected motion model (e.g. constant velocity), but small enough to prevent correlation of noisy data.

2.4 Results

To demonstrate the effectiveness of visual MTT in detecting small, slow-moving targets, a high-resolution camera was fixed to the bottom of an aircraft flying at approximately 60 meters per second at an altitude of about 1400 meters over two walking human targets. The camera had a horizontal field-of-view of 16.09 degrees with 2048×1500 image resolution and publication rate of 20Hz. Under this configuration, targets were described by approximately $10 - 15$ pixels in the resulting images. Visual MTT parameters were hand-tuned using the proposed guidelines to the values shown in Table 2.1 with all other parameters set to their default nominal values. The resulting output of visual MTT successfully demonstrates detection of both intended targets as

Table 2.1: Visual MTT small target parameters

frame stride	1.0
resize scale	1.0
min velocity	0.0002
max velocity	0.001
LKT max features	600
measurement window	20
lifetime threshold	10
max CMD	5
inlier threshold	0.04

shown in Fig. 2.2 in real time. We then conclude that the visual MTT algorithm is capable of detecting small, slow-moving targets from a relatively fast moving camera.

2.5 Visual MTT Target Localization

To be used in a path planning framework, generally the normalized camera coordinates of the detected target provided by visual MTT must be used to estimate an inertial target position expressed in a north-east-down coordinate frame. For ground targets on relatively flat terrain, we assume the target exists on the north-east plane at zero altitude. Localization is then performed by a simple projection onto the observation plane. Let the camera frame target coordinates be defined by Eq. (2.1). The vector can be defined as a vector in the inertial frame as

$$\ell = R_c^i \varepsilon,$$

where $R_c^i \in SO(3)$ defines a passive rotation from the camera to the inertial frame. Using the properties of similar triangles, the inertial target position $\mathbf{p}_t$ can be obtained using the inertial camera position $\mathbf{p}$ as

$$\mathbf{p}_t = \mathbf{p} + \ell \frac{-\mathbf{p}^\top \mathbf{e}_3}{\ell^\top \mathbf{e}_3},$$

where $\mathbf{e}_3 \in \mathbb{R}^3$ is the third canonical basis vector.

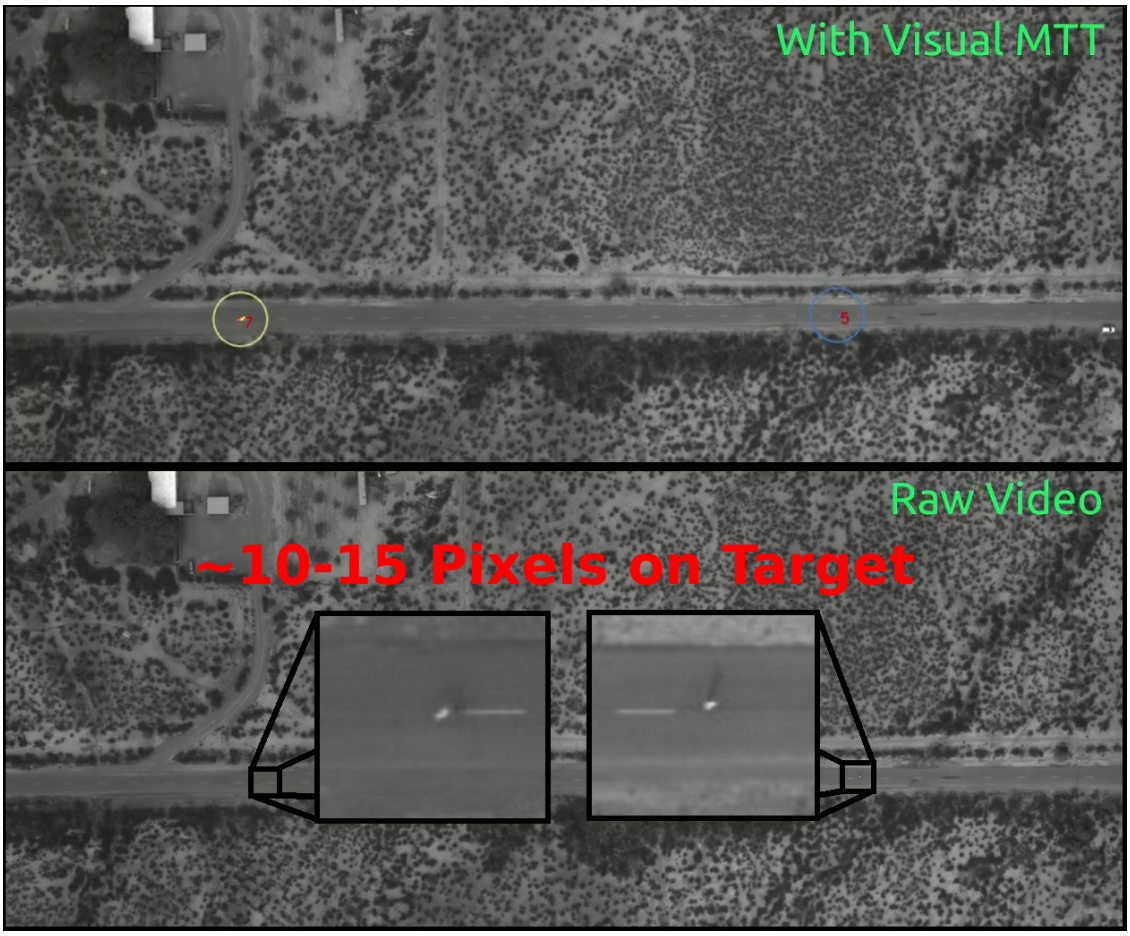


Figure 2.2: Detection of two small walking human targets from a downward-facing camera rigidly attached below an aircraft moving at 60 meters per second from an altitude of 1400 meters. Targets were nearly imperceptible by human monitoring.

CHAPTER 3. VISUAL ROTATIONAL CONSTRAINTS OF GENERAL UAS CONFIGURATIONS[1]

3.1 Introduction

The use of a static or gimballed camera on unmanned aircraft systems (UAS) is common in ground target tracking. For UAS, monitoring a target typically requires maneuvering to a desired vantage point for ideal observation. However, such maneuvers may result in losing visibility of the target of interest. This is particularly true for fixed-wing UAS which rely heavily on roll and pitch adjustments to modify their trajectory. In many cases, maintaining constant visibility of a target may take precedence over closely following a desired trajectory. If so, maintaining a target within the camera's field-of-view may limit the possible motion of an aircraft. Determining the limitations of aircraft motion is then greatly aided by understanding the rotational limits of a UAS and gimbal along their controllable axes to ensure constant target visibility.

Many control techniques have been developed to perform general vision-based target tracking and following. Several authors have approached the problem by managing the commanded flight trajectory of the system [9] [21] [22] [10] [23] [24] [25] [26], while another author approached the problem using hybrid trajectory and gimbal control [27]. Certain approaches have also been developed specifically for maintaining target visibility with robotic arm end effectors or ground vehicles [28] [29]. In many of these applications, the proposed control techniques may have reduced performance or even become unstable if visual contact with the target is lost. It is then critical to understand the visual limitations of the system to ensure proper visual contact is held on the target of interest.

Many applications exist where maintaining target visibility becomes largely more important than maintaining a predetermined trajectory. One significant application of interest are search and rescue missions. In these scenarios, a high-flying, fixed-wing UAS equipped with either a gim-

[1]THIS WORK HAS BEEN ACCEPTED FOR PUBLICATION IN THE 2021 INTERNATIONAL CONFERENCE ON UNMANNED AIRCRAFT SYSTEMS (ICUAS)

balled or statically-attached, downward-facing camera may be deployed over a region of interest to search for, detect, and follow a target. Initially it may be important to maximize camera footprint coverage of the search space; however, once the object of interest is sighted, it becomes critically important to keep the identified object in the camera's field-of-view for classification or localization until a rescue team arrives. Maintaining visibility may be crucial if the object is moving and response teams are unable to get to the object location quickly. A similar fixed-wing or multi-rotor UAS may be deployed in law-enforcement tracking scenarios involving evasive threats. In these cases, surveillance provided by vision-enabled UAS reduces the risk of injury to law-enforcement personnel and civilians. It is then indispensable that threats do not exit the UAS surveillance region. If they do, then threat localization may become inaccurate, and therefore insufficient, for safe containment strategies.

Some researchers have sought to identify gimbal or UAS orientation limits for keeping a target within the field-of-view. One author defined limits on target azimuth and elevation angles relative to the optical axis of an inertially-facing camera [9]; however, the approach requires that the camera be statically mounted so that the roll and pitch axes of the UAS are normal to the span of the camera's longitudinal and lateral fields-of-view respectively. In [10], a limit on the UAS roll angle is defined to maintain a road within the field-of-view by projecting the set of camera pixels onto the observation plane. The pixel projection with the shortest distance from a street point along the pixel translation path produced by rolling motion is used to parameterize the UAS roll angle limits. This approach is once again limited exclusively to static cameras pointed along the UAS z-axis and only constrains the roll axis. The authors in [11] develop a method for determining whether a target exists in the field-of-view of a camera; however, no approach is given for defining allowable motion constraints for maintaining or acquiring target visibility. In applications where a gimbal is needed, these methods define neither the UAS nor the gimbal limits needed to constrain vehicle maneuvers for keeping targets within the field-of-view.

The main contribution of this paper is a novel algorithm for calculating rotational limits of a camera system about any axis or set of axes for maintaining a target of interest in the field-of-view. Typical rotation axes of interest for vision-enabled UAS may include, but are not limited to, UAS body-frame axes and gimbal azimuth and elevation axes. An additional algorithm is presented for identifying the axis and angle defining the shortest transition of target visibility given a set

of available axes. The result identifies the fastest available route for either losing or acquiring target visibility, depending on current visibility conditions. We show that this approach is able to accurately identify all critical rotational degrees of freedom about a standard set of rotational axes (i.e. UAS body x-axis, UAS body y-axis, UAS body z-axis, gimbal elevation, and gimbal azimuth) in simulation.

The problem is formally described in Section 3.2. The algorithm for identifying all critical rotation bounds for a given camera and target configuration is developed in Section 3.3. The algorithm for extracting the closest critical rotation bound and its associated axis is described in Section 3.4. Simulation results for a gimballed UAS are presented in Section 3.5. Various illustrative examples of the algorithm are shown for vision-enabled UAS in Section 3.6. Conclusions are provided in Section 3.7.

3.2 Problem Description

In this paper, we will use the following notation. Let $\mathscr{F}^a = \{\mathbf{i}^a, \mathbf{j}^a, \mathbf{k}^a\}$ denote coordinate frame a with right-handed orthogonal axes $\mathbf{i}^a, \mathbf{j}^a, \mathbf{k}^a$. Let $P^c_{a/b} \in \mathbb{R}^3$ denote the position of frame $\mathscr{F}^a$ relative to frame $\mathscr{F}^b$ expressed in frame $\mathscr{F}^c$, and let $R^b_a \in SO(3)$ denote the rotation matrix that transforms coordinates in frame $\mathscr{F}^a$ to coordinates in frame $\mathscr{F}^b$. In this paper, we use $\mathscr{F}^i$, $\mathscr{F}^b$, $\mathscr{F}^g$, $\mathscr{F}^c$, and $\mathscr{F}^t$ to denote the inertial, UAS body, gimbal, camera, and target frames respectively. We also define the skew-symmetric operator

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}_\times \triangleq \begin{pmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{pmatrix}.$$

To properly define geometric rotational constraints, we will define a UAS equipped with a two-axis gimballed camera system located at a position $P^i_{c/i}$. Let $P^i_{t/i}$ be the position of the target of interest expressed in the inertial frame.

The rotation matrix from the inertial to the camera frame is given by

$$R^c_i = R^c_g R^g_b R^b_i$$

as described by Fig. 3.1.

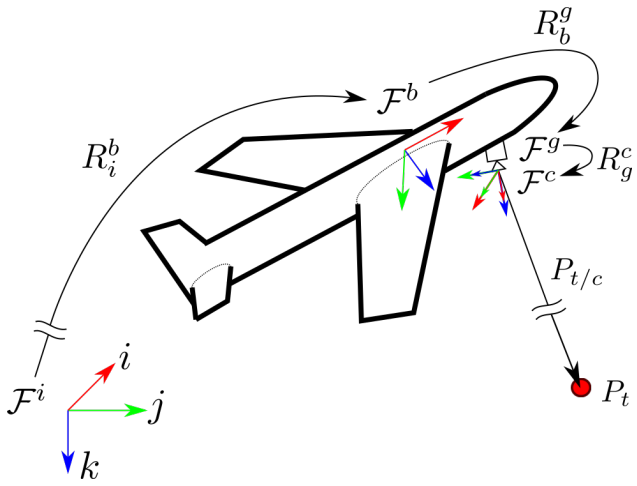


Figure 3.1: Frame rotations and target vector relative to UAS.

We define the position of the target relative to the UAS in the camera frame to be $P_{t/c}^c$, where

$$P_{t/c}^c = R_i^c(P_{t/i}^i - P_{c/i}^i).$$

Let $\ell_{-1,1}, \ell_{1,1}, \ell_{1,-1}$, and $\ell_{-1,-1}$ represent the four corner vectors projected from the camera in the camera frame defining the bounds of the field-of-view. These boundary vectors are defined by rotating the $\mathbf{e}_3$ basis vector in the camera frame about the $\mathbf{i}^c$ and $\mathbf{j}^c$ axes by half of the horizontal and vertical camera field-of-view angles θ_{hfov} and θ_{vfov} respectively as

$$\ell_{i,j} = R_y\left(i\frac{\theta_{vfov}}{2}\right) R_x\left(j\frac{\theta_{hfov}}{2}\right) \mathbf{e}_3, \tag{3.1}$$

where $R_x(\theta)$ and $R_y(\theta)$ are $SO(3)$ rotation matrices defined as

$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{pmatrix}$$
$$R_y(\theta) = \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{pmatrix},$$

and where $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$ are the canonical basis vectors. These boundaries of the camera field-of-view are shown in Fig. 3.2. We define the set of four inward-pointing vectors normal to each of the lateral faces defining the field-of-view as

$$N = \{\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3, \mathbf{n}_4\}, \tag{3.2}$$

where

$$\mathbf{n}_1 = \ell_{1,-1} \times \ell_{-1,-1} \quad \mathbf{n}_2 = \ell_{-1,-1} \times \ell_{-1,1}$$
$$\mathbf{n}_3 = \ell_{-1,1} \times \ell_{1,1} \quad \mathbf{n}_4 = \ell_{1,1} \times \ell_{1,-1}$$

as shown in Fig. 3.2. For reference, we define $\mathbf{p}_t \in \mathscr{F}^c$ to be a vector defining the direction of the target of interest relative to the camera. This can either be calculated as the relative target position vector in the camera frame

$$\mathbf{p}_t = P^c_{t/c} \tag{3.3}$$

or by defining a vector using known pixel coordinates in the image frame

$$\mathbf{p}_t = K^{-1}\mathbf{p}_c, \tag{3.4}$$

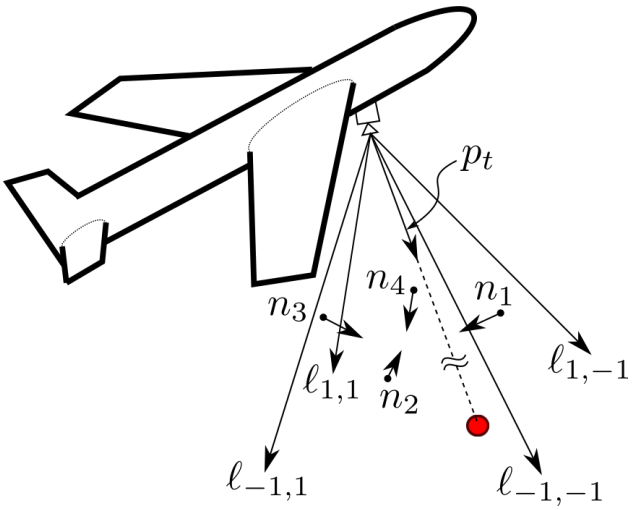


Figure 3.2: UAS field-of-view boundaries and generalized target vector definitions.

where $\mathbf{p}_c$ is the homogeneous pixel coordinate of the target

$$\mathbf{p}_c = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$$

and u and v are pixel column and row coordinates respectively of the target centroid in the image and K is the intrinsic camera matrix. All rotational constraints are calculated relative to a given UAS position, UAS orientation, and target position. Therefore, constraint calculations are only accurate relative to the estimation accuracy of these states. It is also assumed that the horizontal and vertical camera field-of-view angles are less than 180 degrees. All analysis assumes that nothing obstructs the field-of-view of the camera from the target (e.g. tree cover or the wing of the UAS). More conservative constraint buffers may need to be defined to prevent obstruction by UAS components. It is also assumed that translations between the body and camera frames are small relative to the distance between the UAS and the target of interest, and can therefore be ignored.

The following lemma verifies whether or not a given target is considered to be in the camera field-of-view of a given UAS configuration.

Lemma 1 *Let* $\mathbf{p}_t \in \mathbb{R}^3$ *be a vector defining the position of the target of interest relative to the camera in the camera frame* $\mathcal{F}^c$ *with the set of vectors both internal and normal to the four lateral*

field-of-view faces in Eq. (3.2) *defining the matrix*

$$F = \begin{pmatrix} \mathbf{n}_1, & \mathbf{n}_2, & \mathbf{n}_3, & \mathbf{n}_4 \end{pmatrix} \in \mathbb{R}^{3\times 4}. \tag{3.5}$$

Then the target is in the field-of-view if and only if

$$F^\top \mathbf{p}_t \geq 0, \tag{3.6}$$

where the inequality is understood to hold element-wise.

Proof: In order for $\mathbf{p}_t$ to be in the camera field-of-view, it must be encapsulated by the four lateral faces defining the camera field-of-view. If $\mathbf{p}_t$ is internal to a lateral field-of-view face, it necessarily follows that for the inward-pointing vector $\mathbf{n}_i$ normal to the lateral field-of-view face $\mathbf{n}_i^\top \mathbf{p}_t \geq 0$. Therefore, $\mathbf{p}_t$ is in the field-of-view if $\mathbf{n}_i^\top \mathbf{p}_t \geq 0$ for each $\mathbf{n}_i$ giving Eq. (3.6). ■

The field-of-view is then defined as

$$\mathbb{F} = \left\{ x \in \mathscr{F}^c : F^\top x \geq 0 \right\}. \tag{3.7}$$

3.3 Critical Axial Rotations

Consider the problem of a camera sensor with a rotational degree of freedom about a given axis tasked with observing a known target. To understand its rotational limits for maintaining target visibility, it becomes important to understand what rotations would cause the target to either leave or enter the camera field-of-view. To identify these critical rotation states, solutions must be found that result in all possible intercepts of the relative target vector $\mathbf{p}_t$ with any critical field-of-view boundary.

Let $\mathbf{v}$ be the unit vector defining the rotational axis of interest expressed in the camera frame. A desired rotational axis can be calculated in an arbitrary frame $\mathscr{F}^a$ such that the axis can

be described using a canonical basis vector $\mathbf{e}_i \in \mathbb{R}^3$ as

$$\mathbf{v} = R_a^c \mathbf{e}_i^a, \tag{3.8}$$

where the rotation matrix R_a^c is a rotation from $\mathscr{F}^a$ to $\mathscr{F}^c$. For example, if defining the UAS body x-axis, the frame $\mathscr{F}^a$ would be the UAS body frame $\mathscr{F}^b$ with the rotational vector defined as $\mathbf{e}_i = \mathbf{e}_1 = [1,0,0]^\top$ and the rotation R_a^c would be R_b^c. The rotation axis $\mathbf{v}$ may be calculated in this form for simplicity; however, the rotation axis may be calculated as needed as long as it is expressed in $\mathscr{F}^c$. The following theorem then identifies the two valid rotation angles along about $\mathbf{v}$ that result in alignment of $\mathbf{p}_t$ with a selected critical field-of-view boundary.

Theorem 1 *Let $\mathbf{p}_t \in \mathbb{R}^3$ be a vector defining the target of interest in the camera frame $\mathscr{F}^c$. Also, let $\mathbf{v} \in \mathbb{R}^3$ be a unit vector expressed in $\mathscr{F}^c$ about which $\mathbf{p}_t$ is to be rotated. Let N be the set of four field-of-view normal vectors in Eq.* (3.2) *and let F be field-of-view boundary matrix given in Eq.* (3.5)*. Then the set of all rotations Θ_v about $\mathbf{v}$ resulting in critical alignment with any field-of-view boundary are given by*

$$\Theta_v = \{\theta_i \in \Theta_c : F^\top R(\mathbf{v}, \theta_i)^\top \mathbf{p}_t \geq 0\}$$

where $R(\mathbf{v}, \theta_i)$ is the standard Rodriguez formula for rotation by θ_i about a vector $\mathbf{v}$ given by [30]

$$R(\mathbf{v}, \theta_i) = (I - \mathbf{v}\mathbf{v}^\top)\cos\theta_i + \mathbf{v}_\times \sin\theta_i + \mathbf{v}\mathbf{v}^\top, \tag{3.9}$$

and

$$\Theta_c = \left\{ 2\arctan\left(\frac{-b \pm \sqrt{a^2 + b^2 - c^2}}{c - a} \right) : \forall \mathbf{n}_i \in N \right\}, \tag{3.10}$$

where

$$\begin{aligned} a &= \mathbf{n}_i^\top (I - \mathbf{v}\mathbf{v}^\top)\mathbf{p}_t \\ b &= \mathbf{n}_i^\top \mathbf{v}_\times \mathbf{p}_t \\ c &= \mathbf{n}_i^\top \mathbf{v}\mathbf{v}^\top \mathbf{p}_t. \end{aligned}$$

Proof: Let $\hat{\mathbf{p}}_t$ be the vector produced by rotating the target vector $\mathbf{p}_t$ about $\mathbf{v}$ by an angle θ onto the desired field-of-view boundary. The standard Rodriguez formula for a rotation θ about a vector $\mathbf{v}$ is defined in Eq. (3.9). Therefore, rotating $\mathbf{p}_t$ about $\mathbf{v}$ by θ gives

$$\begin{aligned}\hat{\mathbf{p}}_t &= R(\mathbf{v},\theta)\mathbf{p}_t \\ &= (I-\mathbf{v}\mathbf{v}^\top)\mathbf{p}_t\cos(\theta)+\mathbf{v}_\times\mathbf{p}_t\sin(\theta)+\mathbf{v}\mathbf{v}^\top\mathbf{p}_t.\end{aligned} \tag{3.11}$$

Since $\hat{\mathbf{p}}_t$ is the vector result of rotating $\mathbf{p}_t$ onto the lateral field-of-view face created by adjacent boundary vectors, it then holds that for a field-of-view normal vector $\mathbf{n}_i \in N$

$$\mathbf{n}_i^\top \hat{\mathbf{p}}_t = 0.$$

Substitution from Eq. (3.11) then produces the form

$$\mathbf{n}_i^\top \left((I-\mathbf{v}\mathbf{v}^\top)\mathbf{p}_t\cos(\theta)+\mathbf{v}_\times\mathbf{p}_t\sin(\theta)+\mathbf{v}\mathbf{v}^\top\mathbf{p}_t\right) = 0.$$

We define the general form

$$a\cos(\theta)+b\sin(\theta)+c=0, \tag{3.12}$$

where

$$\begin{aligned}a &= \mathbf{n}_i^\top(I-\mathbf{v}\mathbf{v}^\top)\mathbf{p}_t \\ b &= \mathbf{n}_i^\top\mathbf{v}_\times\mathbf{p}_t \\ c &= \mathbf{n}_i^\top\mathbf{v}\mathbf{v}^\top\mathbf{p}_t.\end{aligned}$$

To solve for this generalized form, it can be shown that each trigonometric function can be redefined in terms of $t=\tan\left(\frac{\theta}{2}\right)$ as

$$\sin(\theta)=\frac{2t}{1+t^2}$$

and

$$\cos(\theta)=\frac{1-t^2}{1+t^2}.$$

We can then substitute into Eq. (3.12) to get

$$a\left(\frac{1-t^2}{1+t^2}\right)+b\left(\frac{2t}{1+t^2}\right)+c=0,$$

which can further be simplified into the form

$$t^2(c-a)+t(2b)+(c+a)=0.$$

Using the quadratic formula, we get

$$t=\frac{-b\pm\sqrt{a^2+b^2-c^2}}{c-a}.$$

Using $t=\tan\left(\frac{\theta}{2}\right)$ gives

$$\theta=2\arctan\left(\frac{-b\pm\sqrt{a^2+b^2-c^2}}{c-a}\right).$$

To include rotational intersections with all existing boundaries, we can then state that the set Θ_c of all candidate rotation solutions resulting in a boundary alignment is

$$\Theta_c=\left\{2\arctan\left(\frac{-b\pm\sqrt{a^2+b^2-c^2}}{c-a}\right):\forall\mathbf{n}_i\in N\right\}.$$

There are then at most two solutions for each of the four field-of-view boundaries. Since solutions are found by only enforcing boundary alignment, it is possible to obtain invalid solutions due to rear-alignment as shown in Fig. 3.3 or inconsequential solutions when alignment occurs with one field-of-view bound but extends beyond another bound as shown in Fig. 3.4. Both of these conditions can be avoided by constraining the point rotated by the proposed solution to be within the field-of-view as described in Lemma 1. It then holds that the set of all valid rotations Θ_v is given by

$$\Theta_v=\{\theta_i\in\Theta_c:F^\top\hat{\mathbf{p}}_t\geq 0\}$$

with

$$\hat{\mathbf{p}}_t=R(\mathbf{v},\theta_i)^\top\mathbf{p}_t,$$

where $\hat{\mathbf{p}}_t$ is the vector obtained by rotating the coordinate frame about $\mathbf{v}$ by the candidate $\theta_i\in\Theta_c$ such that $\mathbf{p}_t$ is rotated as defined by the matrix $R(\mathbf{v},\theta_i)$ from Eq. (3.9). This passive rotation of the coordinate frame is therefore the transpose of Eq. (3.9). ■

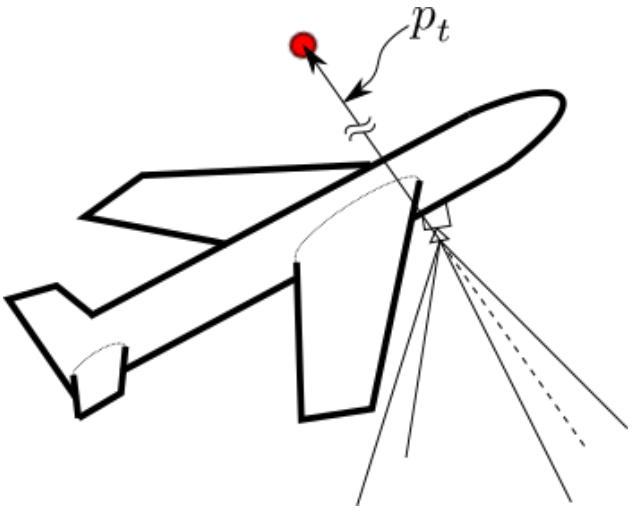


Figure 3.3: Invalid rear-alignment of target vector with field-of-view boundary face.

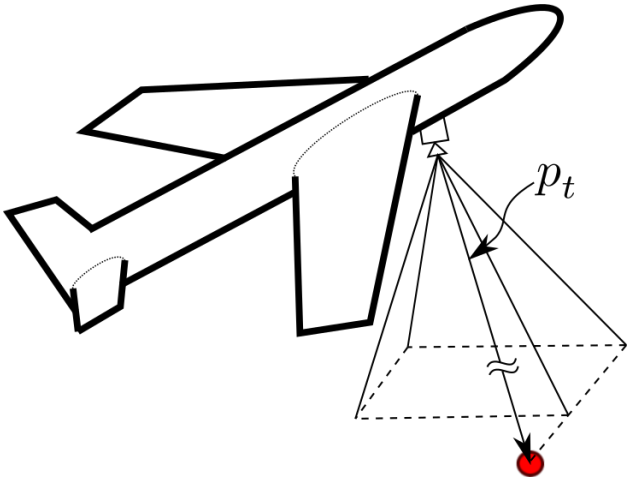


Figure 3.4: Projection is collinear with one critical boundary but extends beyond the other. This bound extension case is trivial and does not result in a change of the target visibility status when the proposed solution is perturbed across the critical bound.

A border threshold δ_b can also be added to validate solutions within the specified error bound of the border. This approach is useful in ensuring that very small offsets due to computation error are still considered acceptable when determining the validity of proposed bound solutions. The following corollary defines the adjusted conditions necessary for this condition.

Corollary 1 *Let* $\mathbf{p}_t \in \mathbb{R}^3$ *be a vector defining the target of interest in the camera frame* $\mathcal{F}^c$*. Let* $\mathbf{v} \in \mathbb{R}^3$ *be a unit vector expressed in* $\mathcal{F}^c$ *about which* $\mathbf{p}_t$ *is to be rotated. Let* N *be the set of four field-of-view normal vectors in Eq.* (3.2) *and let* F *be field-of-view boundary matrix given in*

Eq. (3.5). *Then the set of all rotations* Θ_v *about* $\mathbf{v}$ *resulting in critical alignment with any field-of-view boundary within an acceptable, positive offset score* δ_b *are given by*

$$\Theta_v = \{\theta_i \in \Theta_c : F^\top \bar{\mathbf{p}}_t \geq -\delta_b\}$$

with

$$\bar{\mathbf{p}}_t = \frac{R(\mathbf{v},\theta_i)^\top \mathbf{p}_t}{\left\| R(\mathbf{v},\theta_i)^\top \mathbf{p}_t \right\|}, \tag{3.13}$$

where $R(\mathbf{v},\theta_i)$ *is a Rodrigues rotation matrix defined in Eq.* (3.9)*, and*

$$\Theta_c = \left\{ 2\arctan\left(\frac{-b \pm \sqrt{a^2+b^2-c^2}}{c-a} \right) : \forall \mathbf{n}_i \in N \right\},$$

where

$$a = \mathbf{n}_i^\top (I - \mathbf{v}\mathbf{v}^\top)\mathbf{p}_t$$
$$b = \mathbf{n}_i^\top \mathbf{v}_\times \mathbf{p}_t$$
$$c = \mathbf{n}_i^\top \mathbf{v}\mathbf{v}^\top \mathbf{p}_t.$$

Proof: Let Θ_c be the set of candidate rotational solutions obtained from Eq. (3.10) of Theorem 1. Lemma 1 can then be used to filter invalid solutions by enforcing solutions to be within the field-of-view. Let $\bar{\mathbf{p}}_t$ be the normalized target vector defined as

$$\bar{\mathbf{p}}_t = \frac{\hat{\mathbf{p}}_t}{\|\hat{\mathbf{p}}_t\|}.$$

Given that N is the set of all inward-pointing normal vectors $\mathbf{n}_i$ to the four lateral field-of-view faces as defined in Theorem 1, where previously $\mathbf{n}_i^\top \bar{\mathbf{p}}_t = 0$ for solutions that aligned with the field-of-view bound defined by $\mathbf{n}_i$, we can say that that for a slight offset from the bound considered acceptable that

$$\mathbf{n}_i^\top \bar{\mathbf{p}}_t = -\delta_b,$$

where δ_b is a small positive constant. To expand all field-of-view bounds to anticipate slight computational error, it then holds that

$$F^\top \bar{\mathbf{p}}_t \geq -\delta_b \tag{3.14}$$

where the inequality is element-wise. ■

Defining $\mathbf{p}_t$ using either inertial target coordinates or current target pixel coordinates calculated in Eqs. (3.3) and (3.4) respectively, an algorithmic process for obtaining the results of Corollary 1 is proposed in Algorithm 1.

Algorithm 1 Visual Rotational Constraint (VRC) Solver

procedure VRC
  Input: $\mathbf{p}_t, \mathbf{v}, \theta_{vfov}, \theta_{hfov}, \delta_b$
  Initialize solution set $\Theta_v = \{\}$
  Calculate boundary vectors $\ell_{i,j}$ using Eq. (3.1)
  Calculate field-of-view boundary matrix F using Eq. (3.5)
  Calculate candidate solution set Θ_c using Eq. (3.10)
  for all $\theta_i \in \Theta_c$ **do**
    Calculate $R(\mathbf{v}, \theta_i)$ using Eq. (3.9)
    Calculate $\bar{\mathbf{p}}_t(\theta_i)$ using Eq. (3.13)
    if $\bar{\mathbf{p}}_t(\theta_i)$ satisfies Eq. (3.14) **then**
      Append θ_i to Θ_v as a valid solution
    else
      Reject θ_i
    end if
  end for
end procedure

It should be noted that the smallest magnitude positive angle and smallest magnitude negative angle in the solution set θ_v from Theorem 1 or Corollary 1 define the two closest rotational solutions on either side of the current orientation resulting in a change in target visibility.

3.4 Closest Critical Bound Analysis

In this section, we define an algorithm for identifying the closest critical rotation bound resulting in altered target visibility from a set of normalized rotation axis candidates. Interpretation of the result, depending on initial visibility conditions, is also explored.

In many cases, it is critical to identify the axis and angle rotation associated with the closest critical boundary intersection from a set of identified rotation axes V. This is especially the case when seeking to avoid losing the target over the closest image boundary or seeking to find the optimal direction for obtaining target visualization when the target is out of sight. This can be done by identifying the rotational axis about which the smallest magnitude rotation bound exists from all critical angle solutions about all axes of interest. This closest critical motion assessment is summarized in Algorithm 2

Algorithm 2 Visual Rotational Closest Constraint (VRCC) Solver

procedure VRCC
  Input: $\mathbf{p}_t, V, \theta_{vfov}, \theta_{hfov}, \delta_b$
  Initialize closest axis $v_{\text{cc}} = \{\}$ and angle $\theta_{\text{cc}} = \infty$
  for all rotation axes $\mathbf{v}_i \in V$ **do**
    $\Theta_v \leftarrow \text{VRC}(\mathbf{p}_t, \mathbf{v}_i, \theta_{vfov}, \theta_{hfov}, \delta_b)$ from Algorithm 1
    if $|\min \Theta_v| < |\theta_{\text{cc}}|$ **then**
      $\theta_{\text{cc}} \leftarrow \min \Theta_v$
      $\mathbf{v}_{\text{cc}} \leftarrow \mathbf{v}_i$
    end if
  end for
end procedure

Note that when the target is in the field-of-view the closest boundary intercept will be associated with the closest rotational perturbation about which target visibility is lost. However, when the target is out of the field-of-view, the closest boundary intercept is associated with the closest rotational perturbation about which target visibility is acquired. Thus, the target's current

field-of-view status, as determined by Eq. (3.7), can be used to interpret the meaning of, and inherently the response to, the closest critical boundary rotation axis and angle.

3.5 Results

To assess the accuracy of critical angle calculations, a simple Python simulation was created for visualizing a UAS with an inertially-mounted, two-axis gimbal camera system observing a red target in free space. This simulation allowed orientation state inputs for both UAS Euler angles and gimbal azimuth and elevation angles as well as $\mathbb{R}^3$ positional inputs for the UAS and target. All UAS and gimbal orientation states and UAS and target position states were initialized randomly and then given random-amplitude sinusoidal perturbations in every rotational and translational degree of freedom of the UAS and gimbal over the course of the simulation. The randomized initial UAS, gimbal, and target configurations are shown in Fig. 3.5 alongside a visualization of the initial normalized image plane describing the camera field-of-view with the resulting target projection.

To verify the validity of Algorithm 1, the relative target vector was projected onto the normalized image plane at discrete time steps alongside the field-of-view bounds by division of the third vector element

$$\mathbf{p}_i = \frac{\mathbf{p}_t}{\mathbf{p}_t^\top e^3},$$

where $\mathbf{p}_i \in P^2$ is the resulting projection of $\mathbf{p}_t$ onto the normalized image plane such that the third vector element is 1. These results are shown in Fig. 3.6, where the projection color was set to blue if the configuration met the field-of-view condition in Eq. (3.7) and red otherwise.

The range of rotational perturbations from the current rotation states about each UAS and gimbal axis resulting in target visibility was plotted as a line at each time step using the calculated bounds calculated with Algorithm 1 as shown in Figs. 3.7 and 3.8.

The blue lines in Figs. 3.7 and 3.8 indicate that the target met the field-of-view condition in Eq. (3.7) at the time step while red lines indicate that the target was out of the line of sight. Time steps with no visibility range plotted indicate that no rotation about the proposed axis could result in target visibility. As would be expected, the visibility region is only blue when the region includes the dashed reference line indicating visibility with no perturbation. For the UAS yaw

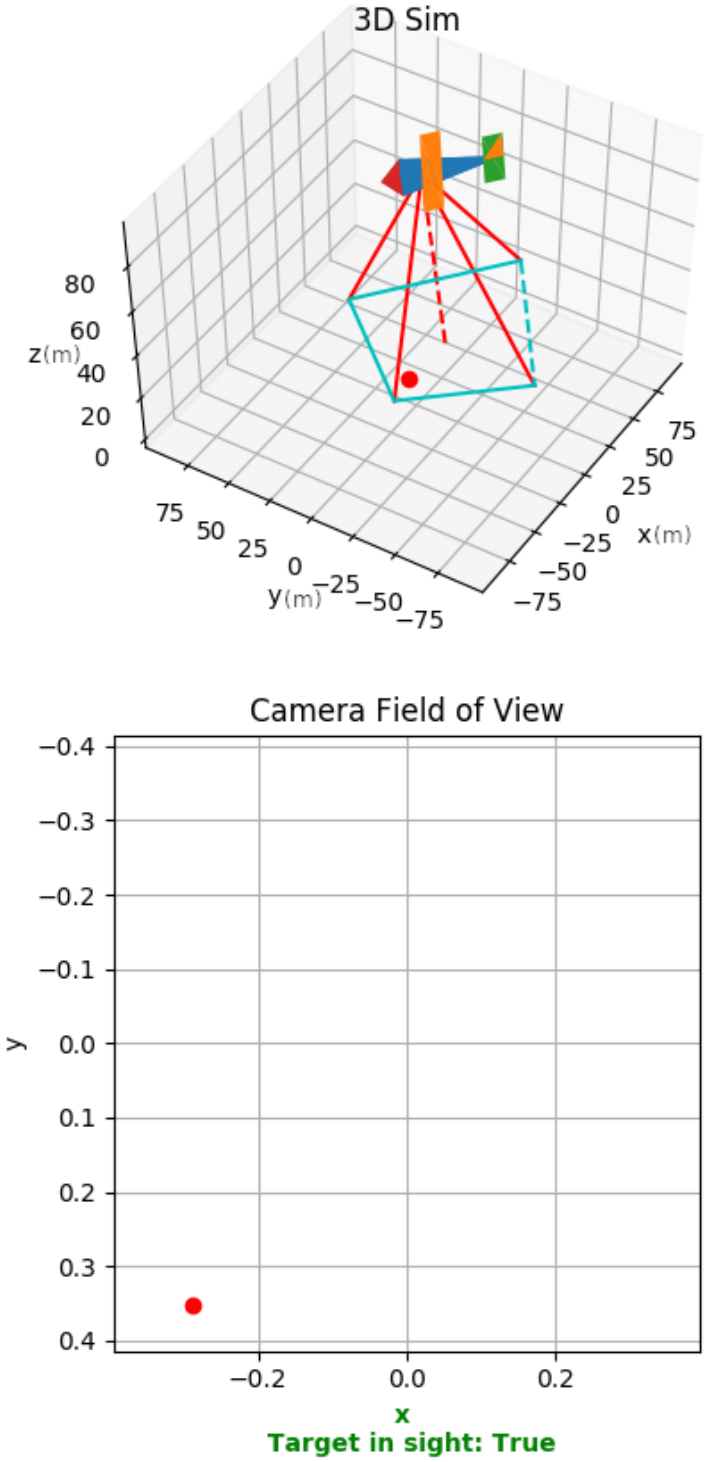


Figure 3.5: Starting configuration of UAS, gimbal and target in simulation with initial camera field-of-view. Red dashed line represents the optical axis. Red lines represent camera field-of-view corner vectors. Cyan lines represent camera field-of-view region projected onto the ground plane. Dashed cyan line represents positive $\mathbf{i}^c$ field-of-view bound for reference.

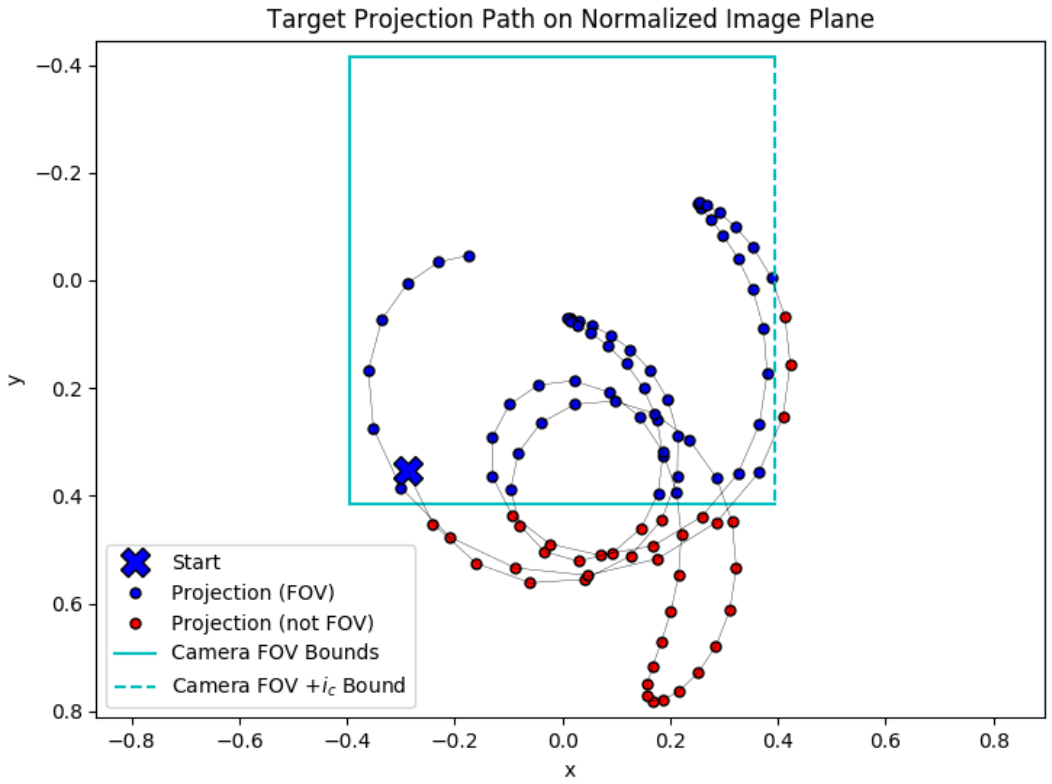


Figure 3.6: Trajectory of target projection on camera's normalized image plane over time with sinusoidal rotational and translational perturbations of UAS, gimbal, and target. Projection points are only blue when within the field-of-view bounds as expected.

bounds and gimbal azimuth bounds, many instances include more than one possible visible region per time step due to the rectangular shape of the field-of-view. Note that in some configurations as the target projection sufficiently aligns with the gimbal azimuth axis, the gimbal azimuth bounds include all rotational values as would be expected.

3.6 Applications for Unmanned Aircraft Systems

The potential applications of Algorithms 1 and 2 extend to any camera system affected by a single or set of rotational axes. Though the approach is generic, many interesting applications exist for vision-enabled UAS systems. In this section, we describe several illustrative examples of the implementation of these algorithms on a UAS system as well as how the results can be used to enhance the design and control of a UAS.

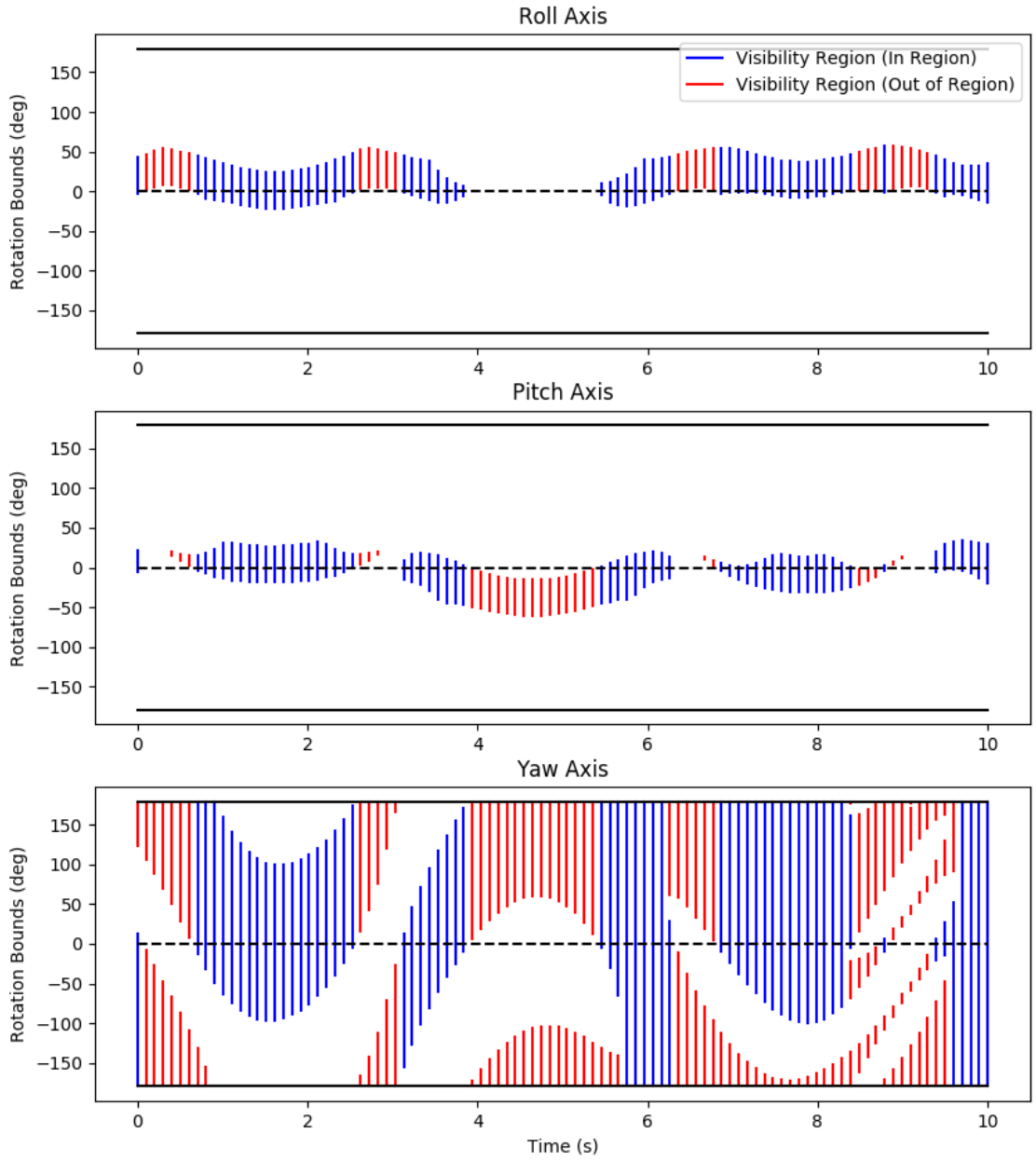


Figure 3.7: Evolving UAS rotation angle perturbation ranges resulting in target visibility at discrete time steps given randomized initial states and sinusoidal rotational and translational perturbations of the UAS, gimbal, and target. Regions are colored blue if the current configuration meets the target visibility constraint with red indicating otherwise. Solid black lines indicate gimbal bounds. Dashed black line denotes no perturbation along axis of interest for reference. The visibility range is only blue when inclusive of the current state indicated by the dashed line as would be expected. Note that multiple visibility regions occur for the yaw axis as a result of crossing multiple FOV boundary corners.

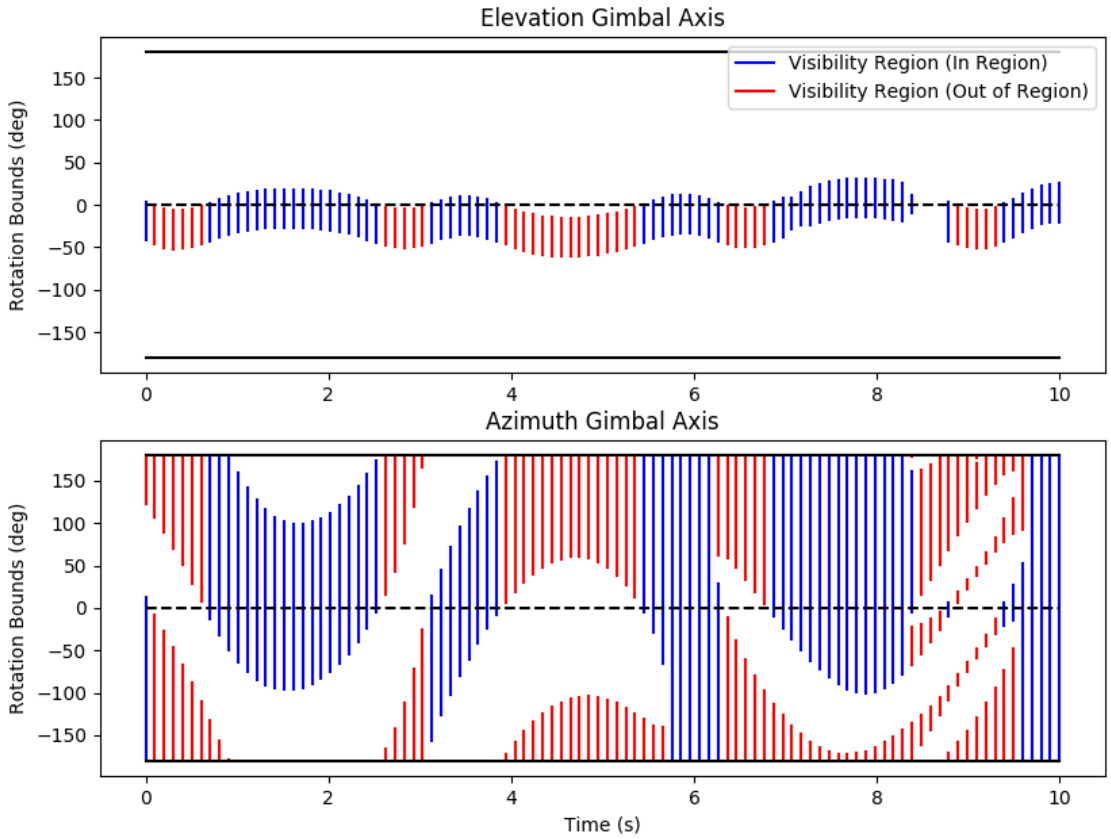


Figure 3.8: Evolving gimbal rotation angle perturbation ranges resulting in target visibility at discrete time steps given randomized initial states and sinusoidal rotational and translational perturbations of the UAS, gimbal, and target. Regions are colored blue if the current configuration meets the target visibility constraint and red otherwise. Solid black lines indicate gimbal bounds. The dashed black line denotes the zero-perturbation state along the axis of interest for reference. The visibility range is only blue when inclusive of the current state indicated by the dashed line as would be expected.

3.6.1 UAS Target Tracking Optimal Gimbal Elevation for Straight, Level Flyover

Suppose a UAS is instructed to quickly fly over a structure of interest for observation and reconnaissance. The primary objective is to collect as much video feed as possible of the structure and its immediate surroundings from a downward-facing camera system mounted to the bottom of the UAS with a single, controllable gimbal elevation axis. The static, inertial position of the center of the structure P_t and inertial UAS position P_u are given. Assuming wind disturbances are negligible, the intended ground speed V_g and altitude of the UAS are calculated to be 50 meters/second and 1000 meters respectively, resulting in relatively level flight. The camera system has vertical

and horizontal field-of-view angle parameters of 10 degrees. Due to resolution limitations of the camera, video data is only useful when the target is within 2000 meters of the camera. Assuming the flight path is a straight line path over the structure with relatively level flight required to maintain the intended airspeed and altitude, the video data can be calculated to have sufficient resolution once the UAS has a north-east distance of 1732 meters from the target. The discrete UAS position can be simply modeled as

$$P_u(t) = P_u(0) + V_g t,$$

where P_u is the UAS position and $t = 0$ is the time at which the UAS is within 2000 meters of the structure. The generalized target vector $\mathbf{p}_t \in \mathscr{F}^c$ can then be calculated as

$$\mathbf{p}_t = P_t - P_u.$$

For this single-axis gimbal case, the gimbal elevation axis aligns with the $\mathbf{i}^c$ axis in the camera frame; therefore, the axis of rotation $\mathbf{v}$ can be defined in the camera frame to be $\mathbf{v} = [1,0,0]^\top$. Selecting a very small computational error buffer $\delta_b = 10^{-10}$, Algorithm 1 can be used at each time step to calculate the rotational perturbations about the gimbal elevation axis resulting in target alignment with the boundary of the camera field-of-view. Solving for the bounds at time steps of 0.25 seconds produces the plot shown in Fig. 3.9.

Note that, as expected, the elevation axis must pitch upward to acquire target visibility when heading toward the target until it is nearly under the aircraft. The bounds then indicate the relative rotations possible while maintaining target visibility.

At each time step, the optimal gimbal elevation angle for keeping the target in the field-of-view is selected as the mean angle between the minimum and maximum boundary angles calculated as

$$\theta_m = \frac{\theta_{min} + \theta_{max}}{2}. \tag{3.15}$$

The set of mean elevation angles θ_m at each discrete time step is fitted to an arctangent curve by optimizing the constant parameters a, b, c, and d with the function model

$$\theta_{opt}(t) = a \arctan\left(b(t+c)\right) + d \tag{3.16}$$

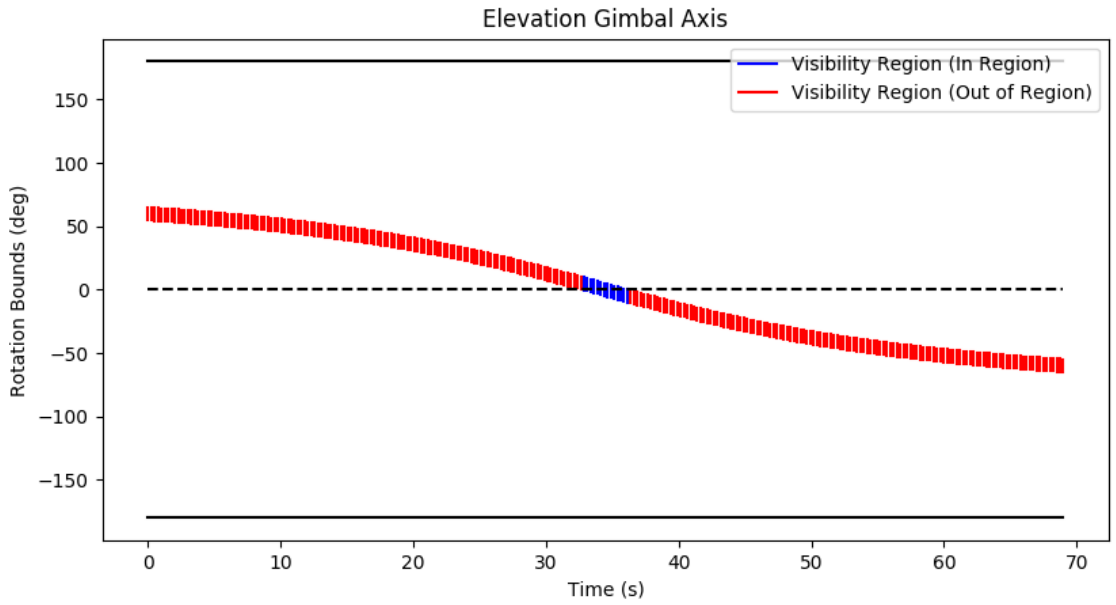


Figure 3.9: Gimbal elevation axis perturbation ranges resulting in target visibility at discrete time steps for flyover of known altitude with constant velocity and static, downward-facing camera. Solid black lines indicate gimbal bounds. Dashed black line denotes no perturbation along axis of interest for reference. As expected, the gimbal elevation must pitch upward at first to capture the target in the field of view and then decreases as the target passes below the aircraft.

producing the optimal coefficients -1.0, 0.05, -1.723, and -1.794×10^{-9} respectively where $\theta_{opt}(t)$ is the optimal gimbal elevation state function for maintaining visibility of the target of interest. Thus, the optimal gimbal elevation angle trajectory is

$$\theta_{opt}(t) = -\arctan\left(0.05(t - 1.723)\right). \tag{3.17}$$

Repeating the simulation with these optimal commanded states for gimbal elevation produces bounds centered on the commanded gimbal elevation state at each time step, which indicates that the target has the largest possible rotational visibility buffer producible by that axis of motion as shown in Fig. 3.10.

The target projections onto the normalized image plane shown in Fig. 3.11 show that the target is held at the image center throughout the trajectory, confirming both the curve fit model validity and the effectiveness of the algorithm in identifying the optimal gimbal states along the elevation axis for maintaining the largest rotational target field-of-view buffer.

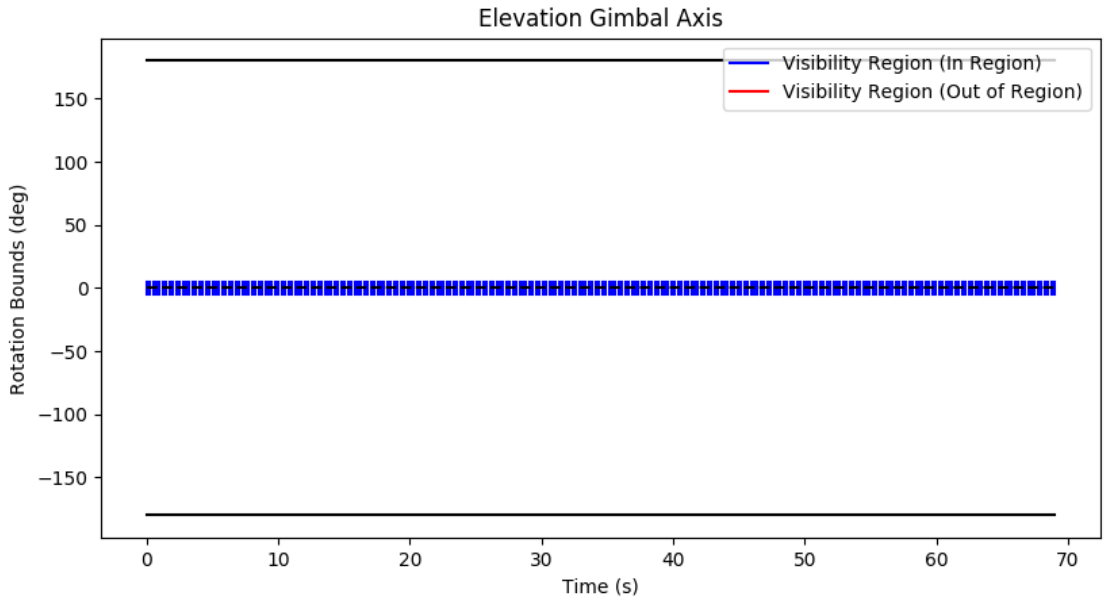


Figure 3.10: Gimbal elevation axis visibility ranges using the proposed optimal control. Dashed black line denotes no perturbation along axis of interest for reference. Optimal control produces visibility with largest possible rotational distance from visibility bounds.

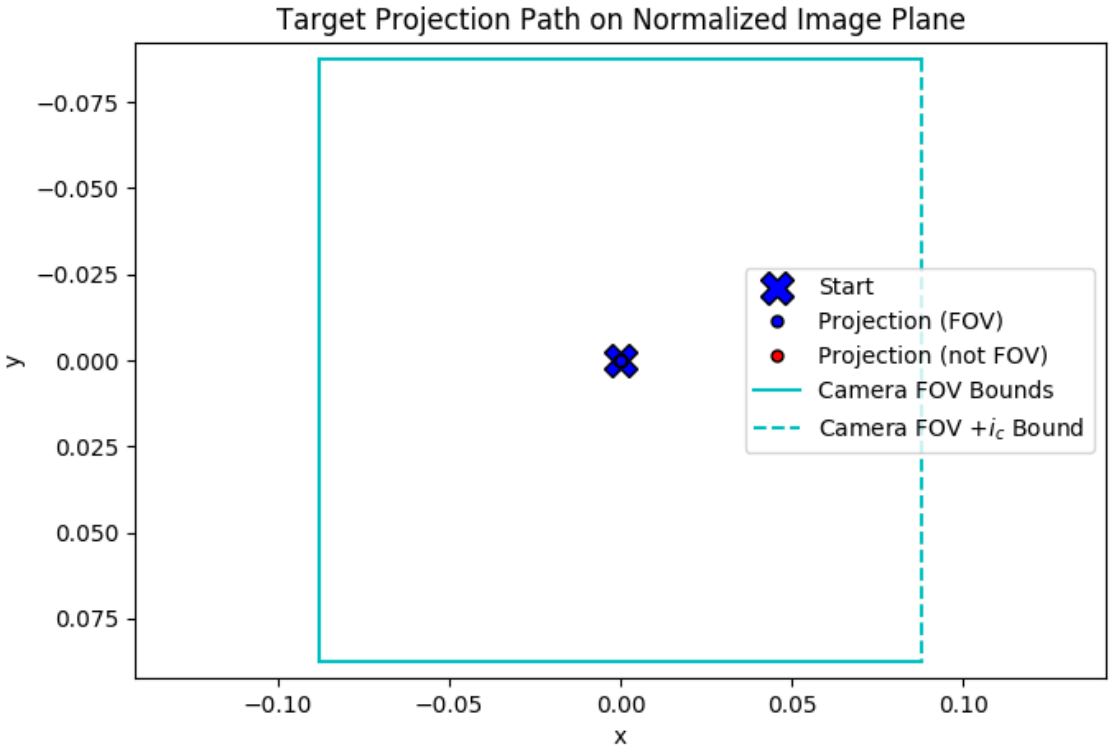


Figure 3.11: Trajectory of target projection on camera's normalized image plane using the optimal control law over time with gimbal elevation commands $\theta_{opt}(t)$ defined in Eq. (3.17). Associated VRC bounds shown in Fig. 3.10.

Additionally, θ_{opt} could be selected to be

$$\theta_{opt}(t) = \theta_{min}(t)$$

or

$$\theta_{opt}(t) = \theta_{max}(t)$$

to produce the minimum and maximum trajectory bounds over which the gimbal elevation axis is allowed to rotate while maintaining target visibility. Fitting the minimum and maximum gimbal elevation angle limits to the arctangent function described by Eq. (3.16) produces the commanded states

$$\theta_{min}(t) = -\arctan\left(0.05(t-1.723)\right) - 0.08726646$$

and

$$\theta_{max}(t) = -\arctan\left(0.05(t-1.723)\right) + 0.08726646.$$

Flexibility is therefore allowed for gimbal motion or state uncertainty as long as the desired gimbal elevation trajectory states or uncertainty bounds lie between these maximum and minimum trajectory functions for all time.

3.6.2 UAS Target Tracking Optimal Gimbal Elevation for Straight Flyover with Constant Climb

Suppose a UAS is commanded to quickly fly over a structure of interest with parameters similar to those described in Section 3.6.1 while maintaining a constant climb rate. The intended ground speed V_g is calculated to be 50 meters/second with a starting altitude of 1000 meters and desired final altitude of 1500 meters. Once within a north-east distance of 500 meters from the target the UAS pitch angle is held at a constant 10 degrees to produce a constant climb rate until the final altitude is reached. The UAS position can be simply modeled as

$$P_u(t) = P_u(0) + V_g t,$$

where P_u is the UAS position and $t = 0$ is the time at which the UAS is within 500 meters of the structure. The generalized target vector $\mathbf{p}_t \in \mathscr{F}^c$ can again be calculated as

$$\mathbf{p}_t = P_t - P_u.$$

Using the same gimbal elevation axis and error buffer as Section 3.6.1, Algorithm 1 can be used at each time step to calculate the rotational offset bounds about the gimbal elevation axis. Solving for the gimbal elevation axis constraints at time steps of 0.25 seconds produces the boundary plot shown in Fig. 3.12.

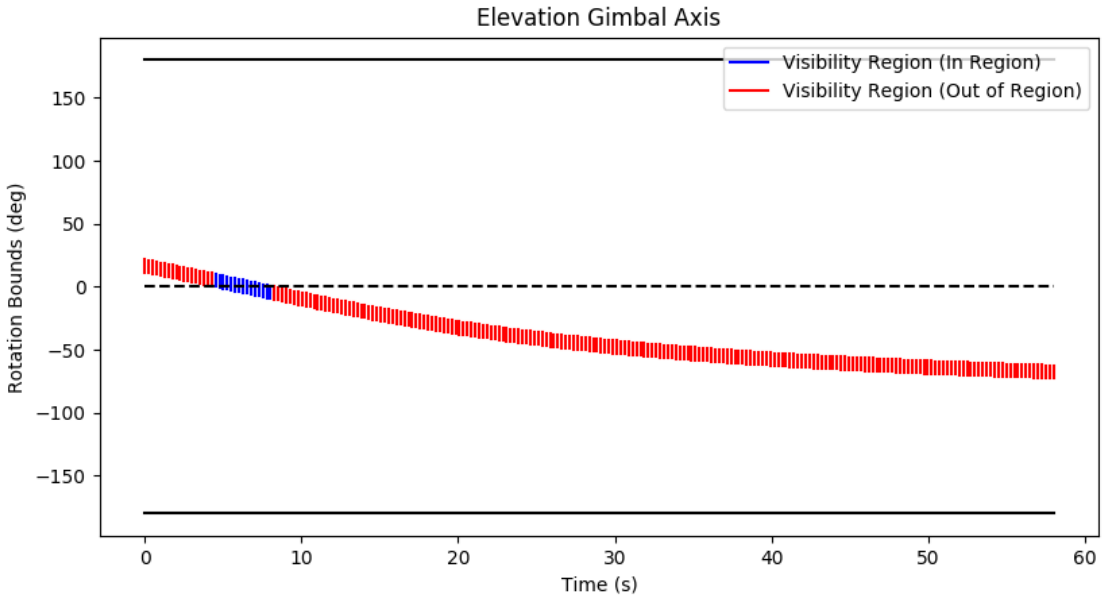


Figure 3.12: Gimbal elevation axis visibility bounds over time for flyover of constantly increasing altitude with constant velocity and a static, downward-facing camera. Dashed black line denotes current gimbal elevation configuration.

At each time step, the optimal gimbal elevation angle for keeping the target in the field-of-view is calculated using Eq. (3.15). Using the arctangent model defined in Eq. (3.16), the optimal gimbal elevation states are calculated to be

$$\theta_{opt}(t) = -\arctan\left(0.04666(t - 0.29745)\right). \tag{3.18}$$

The resulting rotational bounds using these gimbal elevation angles shown in Fig. 3.13 again indicate that the target is held centered in the camera field-of-view with a symmetric, feasible relative error bound.

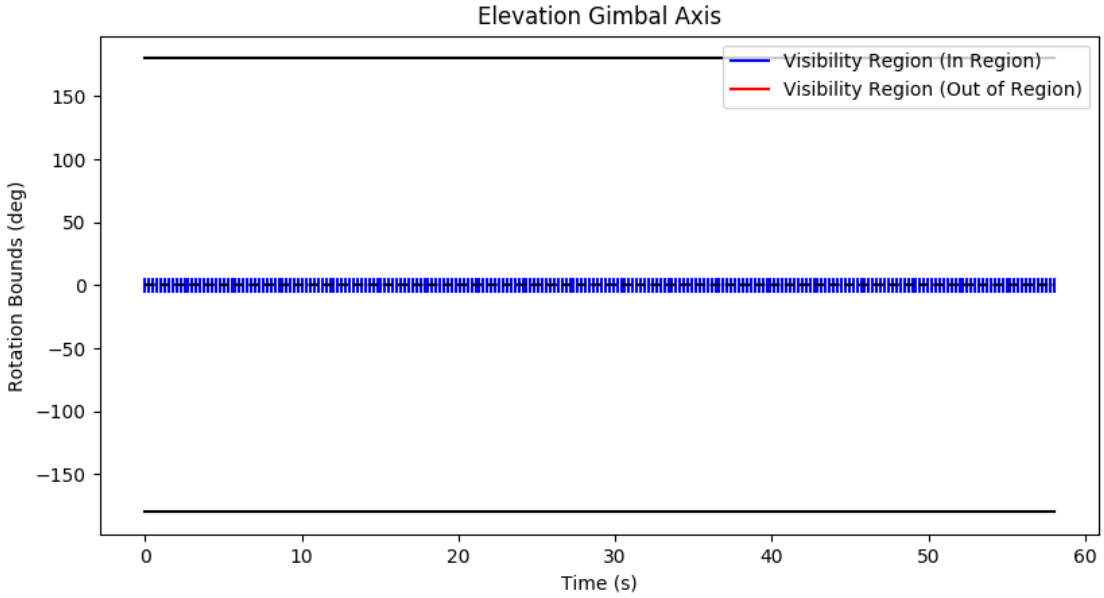


Figure 3.13: Gimbal elevation axis rotational visibility bounds over time for constant climb with constant velocity and using the optimal gimbal elevation control $\theta_{opt}(t)$ defined in Eq. (3.18). Dashed black line denotes current gimbal elevation configuration. Proposed gimbal states result in equal distance from image bounds.

3.6.3 UAS Target Tracking Optimal Orbit for Static Camera Mount Offset

Suppose a UAS is commanded to orbit clockwise over a static target of interest at a constant altitude. The primary objective is to identify the minimum allowable UAS roll angle and orbit radius for holding a structure within the field-of-view from a camera system statically mounted with a 90 degree azimuth offset to the bottom of the UAS and an elevation offset ϕ_g of -30 degrees. The static, inertial position of the center of the structure P_t is given. It is assumed that wind disturbances are negligible. The intended ground speed V_g and altitude of the UAS are defined to be 50 meters/second and 1000 meters respectively during the orbit. The camera system has vertical and horizontal field-of-view angle parameters of 10 degrees. Due to resolution limitations of the camera, video data is again only useful when the target is within 2000 meters of the camera.

The video data can then be calculated to have sufficient resolution once the UAS has a north-east distance of 1732 meters from the target. Thus, the maximum allowable orbit radius is $R_{max} = 1732$ meters.

Using a standard coordinated turn model as defined in [31], with no wind and a negligible flight path angle, we get the orbit radius relationship

$$R = \frac{V_g^2}{g \tan \phi}, \tag{3.19}$$

which simplifies to

$$\phi = \tan^{-1}\left(\frac{V_g^2}{gR}\right), \tag{3.20}$$

where R is the orbit radius, g is gravitational acceleration, and ϕ is the UAS roll angle. By simulating increasing lateral distances from the target of interest with roll angles defined by Eq. (3.20) and running the resulting states through Algorithm 1 at each time step, the roll angle constraints will identify radii where the target is within the field-of-view. All discrete configurations with confirmed target visibility as defined by Eq. (3.7) can be considered viable orbits for maintaining target visibility. Let the axis of interest be the roll axis of the UAS. This axis aligns with the $-i^c$ axis in the camera frame; therefore, the axis of rotation v can be defined in the camera frame to be $\mathbf{v} = [-1,0,0]^\top$. Selecting a very small computational error buffer $\delta_b = 10^{-10}$ and solving for the roll axis constraints at lateral distance steps of 1.0 meter produces the roll axis bounds shown in Fig. 3.14.

As seen in Fig. 3.14, there are two viable orbit radius regions defining the set $R = [181,366] \cup [696,1418]$ meters. The optimal orbit for centering the target in the camera field-of-view can be found by identifying the orbit radius where the average of the maximum and minimum roll bounds is equal to zero, indicating the maximum possible rotational motion in either direction along the roll axis before target visibility is lost. For this case, two optimal orbit radii exist at 244 meters and 1045 meters. If image quality is considered most critical, the 244 meter orbit should be selected to minimize target range; however, if a less aggressive roll or greater orbit flexibility is desired, the 1045 meter orbit radius would be a better selection.

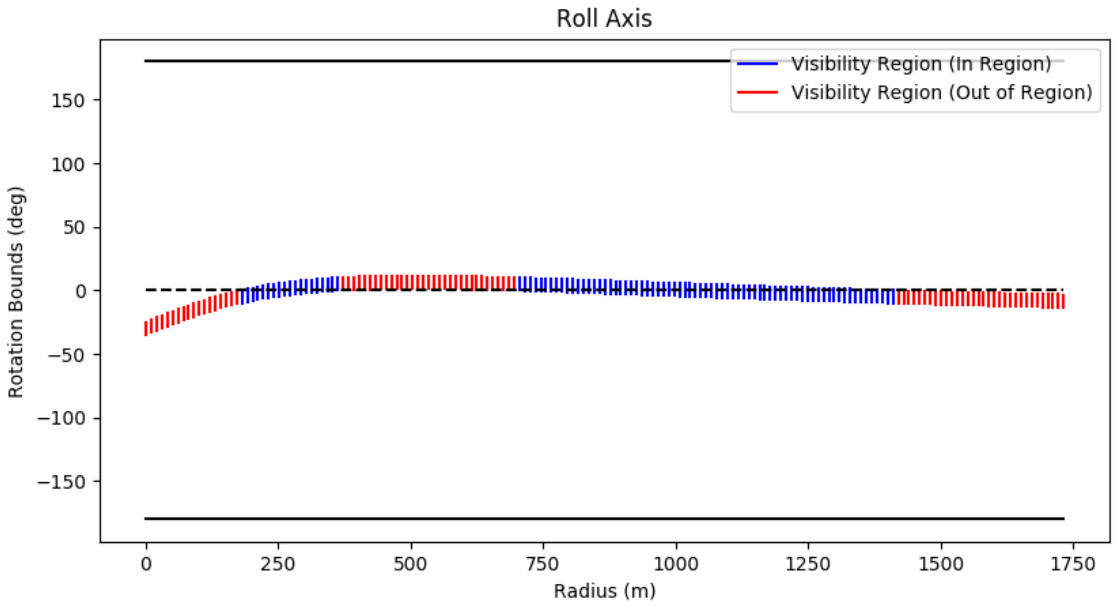


Figure 3.14: UAS roll axis rotational visibility bounds for incremented radii with known altitude, constant velocity, and static, roll-offset camera. Dashed black line denotes current roll configuration.

3.6.4 UAS Target Tracking Optimal Camera Roll Offset for Defined Orbit

Suppose a UAS is commanded to orbit clockwise over a static target of interest at a constant altitude with parameters similar to those described in Section 3.6.3. The primary objective is to identify the optimal static mount angle along the $-\mathbf{i}^b$ axis for a camera system rigidly attached to the bottom of the UAS such that it points out the right wing given a ground speed $V_g = 50$ meters/second and desired orbit radius $R = 1000$ meters. Using the coordinated turn model described in Eq. (3.20), the associated UAS roll angle is $\phi = 14.297$ degrees. Let the axis of interest be the gimbal elevation axis of the UAS given a 90 degree azimuth offset to initially point the camera out of the right wing. This axis aligns with the $\mathbf{i}^c$ axis; therefore, the axis of rotation v can be defined in the camera frame to be $\mathbf{v} = [1,0,0]^\top$. Selecting a very small computational error buffer $\delta_b = 10^{-10}$, Algorithm 1 produces minimum and maximum elevation visibility angles of -35.703 and -25.703 degrees respectively relative to the current elevation angle for target visibility. The optimal mount elevation offset is then the mean angle of -30.703 degrees with approximately ± 5.02 degrees of allowable elevation offset error before the target exits the field-of-view.

3.7 Conclusion

In this paper we presented an algorithm for determining the instantaneous critical rotational limits of a camera system along a given rotational axis for maintaining a target within a defined visual field-of-view. We demonstrated the accuracy of this approach using randomized rotational inputs on a simulation defining a UAS system with a gimballed camera mechanism and identified an approach for calculating the most critical angle and axis of rotation for a given configuration. Additional illustrative examples were discussed for various vision-enabled UAS applications. Future work includes additionally calculating critical translational motion of the camera and UAS for maintaining target visibility, accounting for occlusion points created by UAS appendages such as wings and landing gear, and accounting for UAS, gimbal, and target state uncertainties. Additional work could include using resulting visibility bound from all controllable axes to produce optimal control of all UAS and gimbal rotations simultaneously for maintaining target visibility. Additional work could be done in defining feasibility bounds of simultaneous rotational bound motion to account for more realistic dynamic system trajectories.

CHAPTER 4. FLATNESS-BASED CONTROL OF A GIMBALLED FIXED-WING UAS[1]

4.1 Introduction

Unmanned aircraft systems (UAS) equipped with a downward-facing, gimballed camera system are increasingly being used as a solution for aerial-to-ground target tracking. Both autonomous and semi-autonomous UAS target tracking have applications in fields such as search and rescue, law enforcement, national security, border patrol, covert operations, and wildlife management. In these cases, maintaining targets within the field-of-view (FOV) of the UAS is critical. If FOV constraints are not sufficiently maintained, then visually tracked targets will be lost and may evade detection and localization in the future. The interaction between gimbal constraints and flight dynamics is particularly challenging for fixed-wing UAS as opposed to multirotors due to fewer degrees of freedom in the dynamics.

Several works have developed techniques for controlling a UAS to enforce vision constraints. In [12], a reactive control law successfully guides a fixed-wing aircraft to land on a linear runway while maintaining runway visibility. While effective, this method is designed specifically for tracking linear structures rather than more generalized targets. A strategy is proposed in [9] for maintaining a moving target in the FOV with wind disturbances by controlling and constraining the UAS roll angle toward steady orbits. This approach is effective for transitioning to constant orbits about stationary targets, but was not designed for following other arbitrary UAS or target trajectories and is therefore limited in its scope. The authors in [13] use game theoretic and stochastic optimal control for maintaining a moving, stochastic target within a constrained FOV. The results of [13] show successful reactive control of a fixed-wing aircraft in following and maintaining line-of-sight of a non-holonomic ground vehicle; however, the control design does not attempt to predict future trajectory points where target line-of-sight may be outside the FOV. By proving that a UAS

[1]THIS WORK HAS BEEN ACCEPTED FOR PUBLICATION IN THE 2021 INTERNATIONAL CONFERENCE ON UNMANNED AIRCRAFT SYSTEMS (ICUAS)

system is differentially flat, the authors of [14]–[20] develop a framework for successfully planning dynamically feasible trajectories for UAS. However, these works do not account for feasibility of gimbal states or target visibility throughout the trajectory in their flatness models. The authors of [32]–[36] develop various control laws that account for input saturation constraints, however, constraints are not considered on gimbal angles for maintaining visibility of a target.

The main contribution of this work is the development of a differential-flatness-based model for a fixed-wing UAS equipped with a gimballed camera system that can be used for visual target tracking under gimbal angle saturation constraints and angular rate constraints. The flat outputs for the model are the 3D flight trajectory of the UAS and the 3D trajectory of the target. Differential flatness for this system is shown in Section 4.3. A second contribution of the paper is the derivation of an LQR controller that uses the differential flatness model to both follow the defined UAS trajectory, and to align the optical axis of the camera with the UAS-to-target line-of-sight vector. The LQR controller is briefly described in Section 4.4. A simulation study is presented in Section 4.5 that demonstrates ground target tracking from an orbiting UAS. Conclusions are provided in Section 4.6.

4.2 Problem Description

4.2.1 Mathematical notation

Coordinate frames are right-hand and denoted as $\mathscr{F}^a = \{\mathbf{i}_a, \mathbf{j}_a, \mathbf{k}_a\}$ where $\mathbf{i}_a$, $\mathbf{j}_a$, and $\mathbf{k}_a$ are the basis vectors. The expression $\mathbf{x}^c_{a/b}$ denotes a vector $\mathbf{x}$ of frame $\mathscr{F}^a$ measured with respect to frame $\mathscr{F}^b$ expressed in frame $\mathscr{F}^c$. Let $R^b_a \in SO(3)$ denote the rotation matrix that transforms coordinates in frame $\mathscr{F}^a$ to coordinates in frame $\mathscr{F}^b$. The canonical basis vectors in $\mathbb{R}^3$ are denoted by $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$. The frames $\mathscr{F}^i$, $\mathscr{F}^b$, $\mathscr{F}^g$, $\mathscr{F}^s$ and $\mathscr{F}^t$ denote the inertial, body-fixed, gimbal, stability and target reference frames respectively. Let $\lfloor \mathbf{x} \rfloor$ define the skew-symmetric matrix operator on the generalized vector $\mathbf{x} = (x_1, x_2, x_3)^\top \in \mathbb{R}^3$ such that

$$\lfloor \mathbf{x} \rfloor = \begin{pmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{pmatrix}.$$

4.2.2 Problem Formulation

The equations of motion for a fixed-wing aircraft are given by

$$\dot{\mathbf{p}}^i_{b/i} = \mathbf{v}^i_{b/i} \tag{4.1}$$

$$\dot{\mathbf{v}}^i_{b/i} = g\mathbf{e}_3 + \frac{1}{\mathrm{m}} R^i_s \left((T\cos(\alpha) - D)\mathbf{e}_1 - L\mathbf{e}_3 \right) \tag{4.2}$$

$$\dot{R}^i_b = R^i_b \lfloor \omega^b_{b/i} \rfloor \tag{4.3}$$

$$\omega^b_{b/i} = \omega_c, \tag{4.4}$$

where $\mathbf{p}$ is the UAS position, $\mathbf{v}$ is the UAS velocity, g is the acceleration due to gravity, m is the system mass, ω is the angular velocity of the UAS, ω_c are the commanded angular rates, α is the angle of attack, $D \in \mathbb{R}$ is the drag on the aircraft along the negative $\mathbf{i}_b$ body-fixed axis, $T \in \mathbb{R}$ is the thrust produced by the aircraft along the $\mathbf{i}_b$ body-fixed axis, and $L \in \mathbb{R}$ is the lift force of the aircraft along the negative $\mathbf{k}_b$ body-fixed axis as described in [31]. To simplify the model and enable the derivation of a differential flatness model, we will assume that the rudder deflection, side-slip angle and deviation from the stable angle of attack α_0 are relatively small and that the drag induced by the control surfaces is negligible. Using these assumptions, the drag and lift terms can be defined as

$$D(\mathbf{v}^i_{b/i}, \omega^b_{b/i}) = \frac{1}{2}\rho V_a^2 S (C_{D_0} + C_{D_q} \frac{c}{2V_a} \omega^b_{b/i}{}^{\top} \mathbf{e}_2) \tag{4.5}$$

and

$$L(\mathbf{v}^i_{b/i}, \omega^b_{b/i}) = \frac{1}{2}\rho V_a^2 S (C_{L_0} + C_{L_q} \frac{c}{2V_a} \omega^b_{b/i}{}^{\top} \mathbf{e}_2) \tag{4.6}$$

respectively, where ρ is the air density, $V_a \in \mathbb{R}^+$ is the UAS airspeed, S is the planform area of the wing, c is the mean cord width of the wing, and C_{D_0}, C_{D_q}, C_{L_0}, and C_{L_q} are aerodynamic coefficients as defined in [31]. Assuming zero wind, the UAS airspeed is

$$V_a = \left\| \dot{\mathbf{p}}^i_{b/i} \right\|. \tag{4.7}$$

We assume that the UAS is equipped with a two-axis gimballed camera as defined in [31], where the gimbal azimuth θ_{az} and elevation θ_{el} angles are assumed to be commanded directly using a rapid-response inner control loop. The objective is to image and follow a known ground target defined by

$$\dot{\mathbf{p}}^i_{t/i} = \mathbf{v}^i_{t/i}, \tag{4.8}$$

where $\mathbf{p}^i_{t/i}$ is the position of the target to be tracked and $\mathbf{v}^i_{t/i}$ is the target velocity.

4.3 Differential-flatness Model for Vision-Constrained Fixed-Wing UAS

The general nonlinear system defined by

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}), \tag{4.9}$$

where $\mathbf{x}$ is the state and $\mathbf{u}$ is the control input, with the output mapping

$$\mathbf{z} = h(\mathbf{x}), \tag{4.10}$$

is differentially flat with the flat output $\mathbf{z}$ if the following conditions are met [37]:

1. The components of $\mathbf{z}$ are not differentially related over time,

2. The state $\mathbf{x}$ can be written as a function of the flat output and its derivatives, i.e., there exists a function g_1, and a finite scalar m_1 such that

$$\mathbf{x}(t) = g_1\left(\mathbf{z}(t), \frac{d}{dt}\mathbf{z}(t), \cdots, \frac{d^{m_1}}{dt^{m_1}}\mathbf{z}(t)\right).$$

3. The control inputs $\mathbf{u}$ can be written as a function of the flat output and its derivatives, i.e., there exists a function g_2 and a finite scalar m_2 such that

$$\mathbf{u}(t) = g_2\left(\mathbf{z}(t), \frac{d}{dt}\mathbf{z}(t), \cdots, \frac{d^{m_2}}{dt^{m_2}}\mathbf{z}(t)\right).$$

Therefore, differentially flat systems admit a direct algebraic mapping from the flat outputs to the states and control inputs. This mapping allows trajectory planning to be performed in the lower-dimensional flat output space with a closed-form solution to control inputs along the anticipated path.

The main result of the paper is given in following theorem that states that the UAS/gimbal system is differentially flat with flat output given by the UAS and target positions.

Theorem 2 Assume that

1. there is no wind,
2. the angle-of-attack is constant and equal to α_0,
3. the side-slip angle is zero, and
4. the UAS/target line-of-sight vector is aligned with the camera optical axis.

Then the aircraft and target defined by Eqs. (4.1) to (4.5), (4.7) and (4.8) with states and inputs given by

$$\mathbf{x} = \{\mathbf{p}^i_{b/i}, \mathbf{p}^i_{t/i}, \mathbf{v}^i_{b/i}, R^i_b\}$$
$$\mathbf{u} = \{\omega^b_{b/i}, T, \theta_{\text{az}}, \theta_{\text{el}}\},$$

are differentially flat with flat output

$$\mathbf{z} \triangleq \left(\mathbf{z}_b^\top, \quad \mathbf{z}_t^\top\right)^\top = \left(\mathbf{p}^{i\top}_{b/i}, \quad \mathbf{p}^{i\top}_{t/i}\right)^\top,$$

where all states and inputs are defined by $\mathbf{z}_b$, $\dot{\mathbf{z}}_b$, $\ddot{\mathbf{z}}_b$, $\dddot{\mathbf{z}}_b$ and $\mathbf{z}_t$.

Proof: We first show that the UAS subsystem is differentially flat and then that the UAV/gimbal/target system is differentially flat. For the UAS subsystem, we begin by showing that the system states $\mathbf{x}$ can be described in terms of the flat outputs $\mathbf{z}$, a finite number of its derivatives and the control inputs $\mathbf{u}$. Trivially, $\mathbf{p}^i_{b/i}(\mathbf{z}_b) = \mathbf{z}_b$, and $\mathbf{v}^i_{b/i}(\dot{\mathbf{z}}_b) = \dot{\mathbf{z}}_b$. Let the stability-to-inertial rotation matrix R^i_s be expressed as

$$R^i_s = \begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3\end{bmatrix},$$

where $\mathbf{r}_j$ is the j^{th} stability frame axis of the fixed-wing expressed in the inertial frame. By assumptions (1) and (2), the aircraft's velocity is aligned with the $\mathbf{i}_s$ stability axis, i.e.,

$$\mathbf{r}_1 = \frac{\dot{\mathbf{z}}_b}{\|\dot{\mathbf{z}}_b\|}.$$

Accordingly, Eq. (4.2) becomes

$$\dot{\mathbf{v}}^i_{b/i} - g\mathbf{e}_3 = \frac{(T\cos(\alpha_0) - D)\mathbf{r}_1 - L\mathbf{r}_3}{\mathrm{m}}. \tag{4.11}$$

Taking the dot product with $\mathbf{r}_1$ gives

$$\frac{(T\cos(\alpha_0) - D)}{\mathrm{m}} = \mathbf{r}_1^\top(\dot{\mathbf{v}}^i_{b/i} - g\mathbf{e}_3) \tag{4.12}$$

Rearranging Eq. (4.11), and using Eq. (4.12) gives

$$\frac{L}{\mathrm{m}}\mathbf{r}_3 = (\mathbf{r}_1^\top(\dot{\mathbf{v}}^i_{b/i} - g\mathbf{e}_3))\mathbf{r}_1 - (\dot{\mathbf{v}}^i_{b/i} - g\mathbf{e}_3). \tag{4.13}$$

Since $\|\mathbf{r}_3\| = 1$, we then conclude that

$$\mathbf{r}_3 = \frac{(\mathbf{r}_1^\top(\dot{\mathbf{v}}^i_{b/i} - g\mathbf{e}_3))\mathbf{r}_1 - (\dot{\mathbf{v}}^i_{b/i} - g\mathbf{e}_3)}{\left\|(\mathbf{r}_1^\top(\dot{\mathbf{v}}^i_{b/i} - g\mathbf{e}_3))\mathbf{r}_1 - (\dot{\mathbf{v}}^i_{b/i} - g\mathbf{e}_3)\right\|}.$$

It necessarily follows that

$$\mathbf{r}_2 = \mathbf{r}_3 \times \mathbf{r}_1.$$

The stability-to-inertial rotation matrix can then be used to calculate the body-to-inertial rotation matrix as

$$R^i_b = R^i_s R^s_b,$$

where

$$R^s_b = \begin{bmatrix} \cos(\alpha_0) & 0 & \sin(\alpha_0) \\ 0 & 1 & 0 \\ -\sin(\alpha_0) & 0 & \cos(\alpha_0) \end{bmatrix},$$

as defined in [31]. Therefore, the rotation matrix R_b^i can be written as a function of the flat outputs $R_b^i(\dot{\mathbf{z}}_b, \ddot{\mathbf{z}}_b)$.

From Eq. (4.3), the angular velocity vector is given by

$$\lfloor \omega_{b/i}^b \rfloor = R_b^{i\top} \dot{R}_b^i.$$

Defining the "vee" operator to be

$$\begin{pmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{pmatrix}^\vee = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix},$$

the angular velocity can then be expressed in terms of flat outputs as

$$\omega_{b/i}^b(\dot{\mathbf{z}}_b, \ddot{\mathbf{z}}_b, \mathbf{z}_b^{(3)}) = (R_b^{i\top} \dot{R}_b^i)^\vee. \tag{4.14}$$

Equation (4.12) can be rewritten in terms of T as

$$T = \frac{\mathrm{m}(\mathbf{r}_1^\top(\dot{\mathbf{v}}_{b/i}^i - g\mathbf{e}_3)) + D}{\cos(\alpha_0)}. \tag{4.15}$$

Substituting Eqs. (4.5) and (4.7) into Eq. (4.15), and using the fact that V_a and $\omega_{b/i}^b$ are defined in terms of $\dot{\mathbf{z}}_b$, and $\ddot{\mathbf{z}}_b$ gives

$$T(\dot{\mathbf{z}}_b, \ddot{\mathbf{z}}_b, \mathbf{z}_b^{(3)}) = \frac{\mathrm{m}}{\cos\alpha_0}\mathbf{r}_1^\top(\dot{\mathbf{v}}_{b/i}^i - g\mathbf{e}_3) + \frac{\rho V_a^2 S}{2\cos\alpha_0}(C_{D_0} + C_{D_q}\frac{c}{2V_a}\omega_{b/i}^{b\ \top}\mathbf{e}_2).$$

Therefore, since all UAS states and inputs are defined in terms of flat outputs, the UAS subsystem is then differentially flat.

The normalized line-of-sight vector from the UAS to the target is given by

$$\bar{\mathbf{p}}_{t/b}^b(\dot{\mathbf{z}}_b, \ddot{\mathbf{z}}_b, \mathbf{z}_t) = \frac{R_b^{i\top}(\mathbf{p}_{t/i}^i - \mathbf{p}_{b/i}^i)}{\left\| R_b^{i\top}(\mathbf{p}_{t/i}^i - \mathbf{p}_{b/i}^i) \right\|},$$

Define the camera optical axis as

$$\ell^b_{\text{cam}} = R^b_g \mathbf{e}_1, \tag{4.16}$$

where the body-to-gimbal rotation matrix R^b_g defined as [38]

$$R^b_g(\theta_{\text{az}}, \theta_{\text{el}}) = \begin{pmatrix} c_{\theta_{\text{el}}} c_{\theta_{\text{az}}} & -s_{\theta_{\text{az}}} & -s_{\theta_{\text{el}}} c_{\theta_{\text{az}}} \\ c_{\theta_{\text{el}}} s_{\theta_{\text{az}}} & c_{\theta_{\text{az}}} & -s_{\theta_{\text{el}}} s_{\theta_{\text{az}}} \\ s_{\theta_{\text{el}}} & 0 & c_{\theta_{\text{el}}} \end{pmatrix},$$

where $c_\varphi \triangleq \cos\varphi$ and $s_\varphi \triangleq \sin\varphi$ Therefore Eq. (4.16) gives

$$\ell^b_{\text{cam}} = \begin{pmatrix} \cos\theta_{\text{el}} \cos\theta_{\text{az}} \\ \cos\theta_{\text{el}} \sin\theta_{\text{az}} \\ \sin\theta_{\text{el}} \end{pmatrix}. \tag{4.17}$$

By Assumption (4), $\bar{\mathbf{p}}^b_{t/b} = \ell^b_{\text{cam}}$, implying that

$$\theta_{\text{el}}(\mathbf{z}_b, \dot{\mathbf{z}}_b, \ddot{\mathbf{z}}_b, \mathbf{z}_t) = \arcsin\left(\mathbf{e}_3^\top \bar{\mathbf{p}}^b_{t/b}\right)$$
$$\theta_{\text{az}}(\mathbf{z}_b, \dot{\mathbf{z}}_b, \ddot{\mathbf{z}}_b, \mathbf{z}_t) = \arctan 2\left(\mathbf{e}_2^\top \bar{\mathbf{p}}^b_{t/b}, \mathbf{e}_1^\top \bar{\mathbf{p}}^b_{t/b}\right).$$

Since all states and inputs are defined in terms of flat outputs and their derivatives, the UAS/gimbal/target system is differentially flat. ■

The differential flatness model given in Theorem 2 therefore provides a direct mapping from the UAS and target trajectories to the input space of the UAS and gimbal. Thus, this model can be used to rapidly check if a given trajectory will result in input saturation for dynamic feasibility and target tracking feasibility.

4.4 Error State LQR Control

By Theorem 2, the fixed-wing/camera target tracking system is differentially flat and, therefore, produces a direct feed-forward mapping from the flat outputs to the desired states and inputs. An LQR feedback controller is then used to track the desired trajectory as shown in Fig. 4.1.

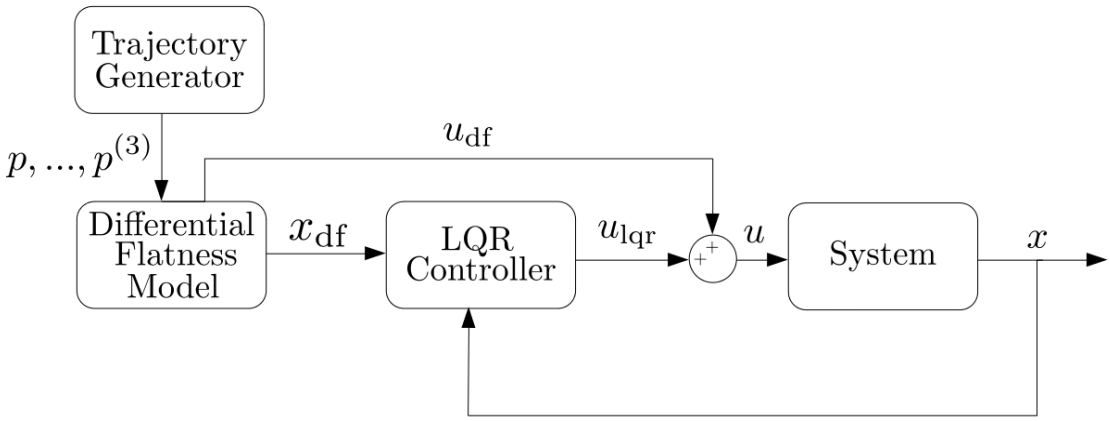


Figure 4.1: Control architecture uses a given inertial position trajectory p up to its third derivative to define the predicted control $u_{(\mathrm{df})}$ and desired aircraft states $x_{(\mathrm{df})}$ using the differential flatness mapping. The error between the desired states and actual states x are passed to an LQR controller whose control output u_{lqr} augments the feed-forward inputs produced by the differential flatness map.

The error state of the LQR controller is given by

$$\mathbf{e} = [\tilde{\mathbf{p}}^\top, \tilde{V}_a, \tilde{\Theta}^\top, \tilde{\omega}^\top, \mathbf{I}^\top]^\top,$$

where

$$\tilde{\mathbf{p}} \triangleq \begin{pmatrix} e_a \\ e_t \\ \tilde{h} \end{pmatrix} = \begin{pmatrix} \cos\chi & \sin\chi & 0 \\ -\sin\chi & \cos\chi & 0 \\ 0 & 0 & -1 \end{pmatrix} \left(\mathbf{p}^i_{b/i(\mathrm{df})} - \mathbf{p}^i_{b/i} \right)$$

are the axial position error, transverse position error, and altitude error, where χ is the course angle of the aircraft,

$$\tilde{V}_a \triangleq V_{a(\mathrm{df})} - V_a$$

is the airspeed error,

$$\tilde{\Theta} \triangleq \mathrm{Euler}(R^i_{b(\mathrm{df})}) - \mathrm{Euler}(R^i_b)$$

is attitude error expressed in local Euler coordinates with Euler$(\cdot)$ being the X-Y-Z Euler angle decomposition of a rotation matrix,

$$\tilde{\omega} \triangleq \omega^b_{b/i(\mathrm{df})} - \omega^b_{b/i}$$

is the angular velocity error, and

$$\mathbf{I} \triangleq \begin{pmatrix} \int_{-\infty}^{t} \tilde{h} d\tau \\ \int_{-\infty}^{t} (V_{a(\mathrm{df})} - V_a) d\tau \\ \int_{-\infty}^{t} e_a d\tau \\ \int_{-\infty}^{t} e_t d\tau \end{pmatrix}$$

are the integral errors on altitude, airspeed, axial position, and transverse position errors.

Linearizing around the aircraft model defined in [31] at level-flight conditions produces the state-space model

$$\dot{\mathbf{e}} = A\mathbf{e} + G\tilde{\mathbf{u}}.$$

The LQR controller is $\tilde{\mathbf{u}} = -K\mathbf{e}$, where the LQR gains K are found by solving the associated algebraic Riccati equation.

4.5 Simulation Results

To assess the effectiveness of the differential-flatness model and LQR controller, we simulate a fixed-wing UAS with a two-axis gimbal camera system observing a ground target, using the ROS/Gazebo simulation environment. The UAS dynamics were implemented following the fixed-wing model in [31] with parameters defined in Table 4.1.

For the UAS, a circular trajectory was defined at a constant altitude of 3000 meters with a radius of 2500 meters, a constant airspeed velocity of 120 meters per second, and a starting north-east-down (NED) inertial position of $[-2000, 100, -3000]$ meters initially facing north. The ground target was given a 1000 meter radius circular trajectory with a starting inertial position at $[-1000, 0, 0]$ meters and a constant velocity of 20 meter per second. Higher-order derivatives of the

Table 4.1: Simulated fixed-wing system parameters

Param	Value	Param	Value	Param	Value
m	1800.0	J_x	700.0	J_y	5000.0
J_z	5500.0	J_{xz}	70.0	S	9.0
b	7.0	c	1.5	C_{L_0}	0.15
C_{L_α}	4.5	C_{L_q}	9.0	$C_{L_{\delta_e}}$	0.3
C_{D_0}	0.02	C_{D_α}	-0.3	C_{D_q}	0.0
C_{l_0}	0.0	C_{l_β}	-0.16	C_{l_p}	-0.4
C_{l_r}	0.1	$C_{l_{\delta_a}}$	0.06	$C_{l_{\delta_r}}$	0.04
C_{m_0}	-0.02	C_{m_α}	-2.4	C_{m_q}	-11.0
$C_{m_{\delta_e}}$	-0.5	C_{n_0}	0.0	C_{n_β}	0.1
C_{n_p}	0.02	C_{n_r}	-0.3	$C_{n_{\delta_a}}$	0.0
$C_{n_{\delta_r}}$	-0.08	C_{Y_0}	0.0	C_{Y_β}	-0.5
C_{Y_p}	-0.1	C_{Y_r}	0.6	$C_{Y_{\delta_r}}$	0.16

UAS trajectory were easily obtained using derivative properties of the sine and cosine components used to create the circular trajectories. The resulting trajectories projected onto the north-east plane are shown in Fig. 4.2.

Figures 4.3 and 4.4 show the error in the actual states and inputs relative to those required to align the ground target with the optical axis and those predicted by the differential-flatness model. To demonstrate the error in aligning the target with the optical axis, we define the offset angle

$$\eta = \cos^{-1}\left(\left(R_b^i \ell_{\text{cam}}^b\right)^\top \left(\frac{\mathbf{p}_{t/i}^i - \mathbf{p}_{t/b}^i}{\left\|\mathbf{p}_{t/i}^i - \mathbf{p}_{t/b}^i\right\|}\right)\right)$$

between the relative target vector and optical axis.

As can be seen in Figs. 4.3 and 4.4, the flatness-based controller is able to successfully follow the desired trajectory and maintain the target line-of-sight angle to within ± 1.2 degree after the initial transients. The constant 1 degree error in aircraft roll and pitch is a result of estimating a constant angle of attack at steady level flight conditions and the simplifying assumption of no side slip made in the proposed differential flatness model. The proposed control method was successfully able to maintain target visibility as predicted through the duration of the flight trajectory. It

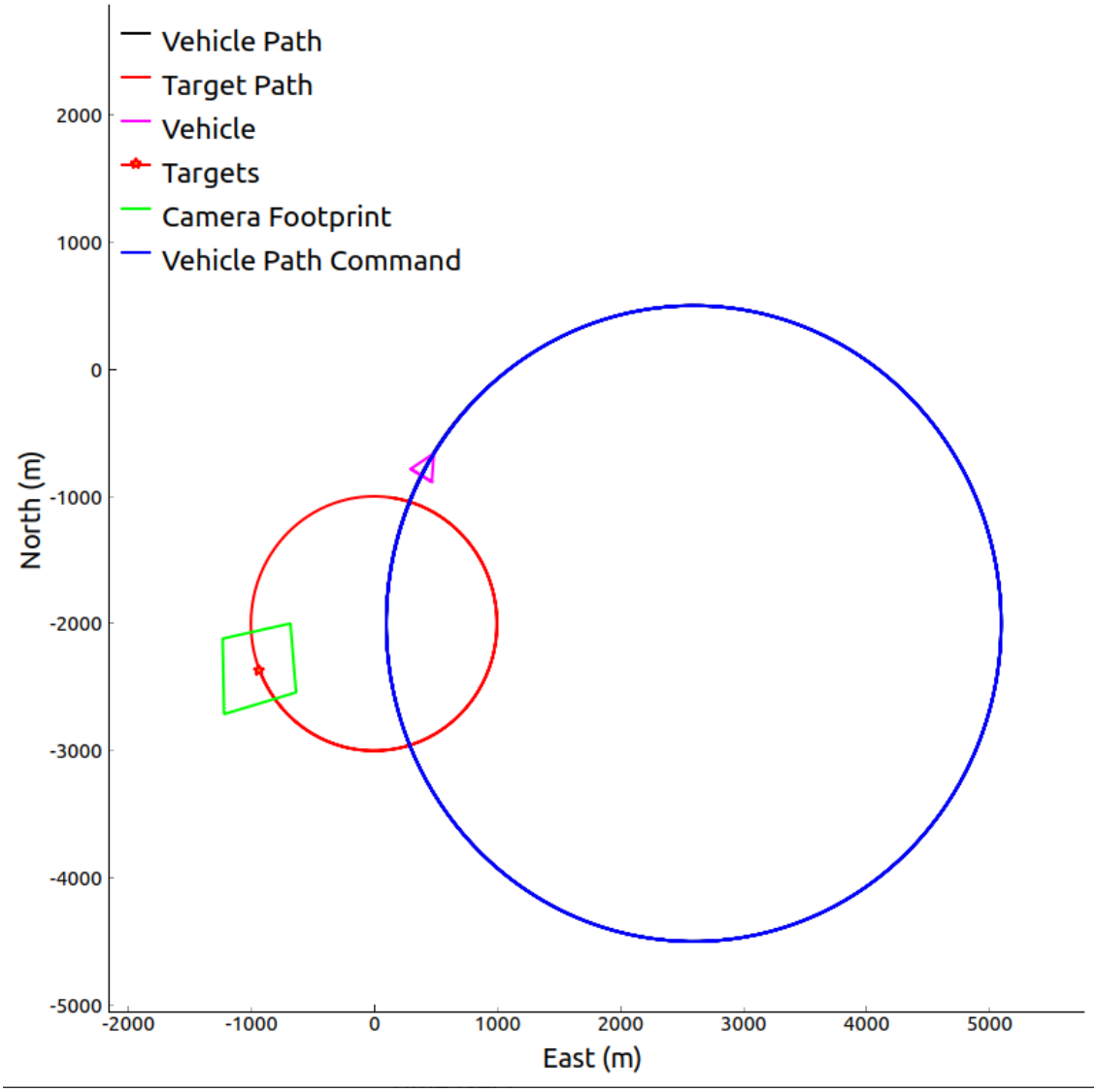


Figure 4.2: Orbit trajectory of the UAS and linear trajectory of the ground target projected onto the inertial north-east plane. Projection of the camera field-of-view on the ground plane containing the target is shown. As can be seen, the proposed control model holds the target relatively close to the center of the camera field-of-view.

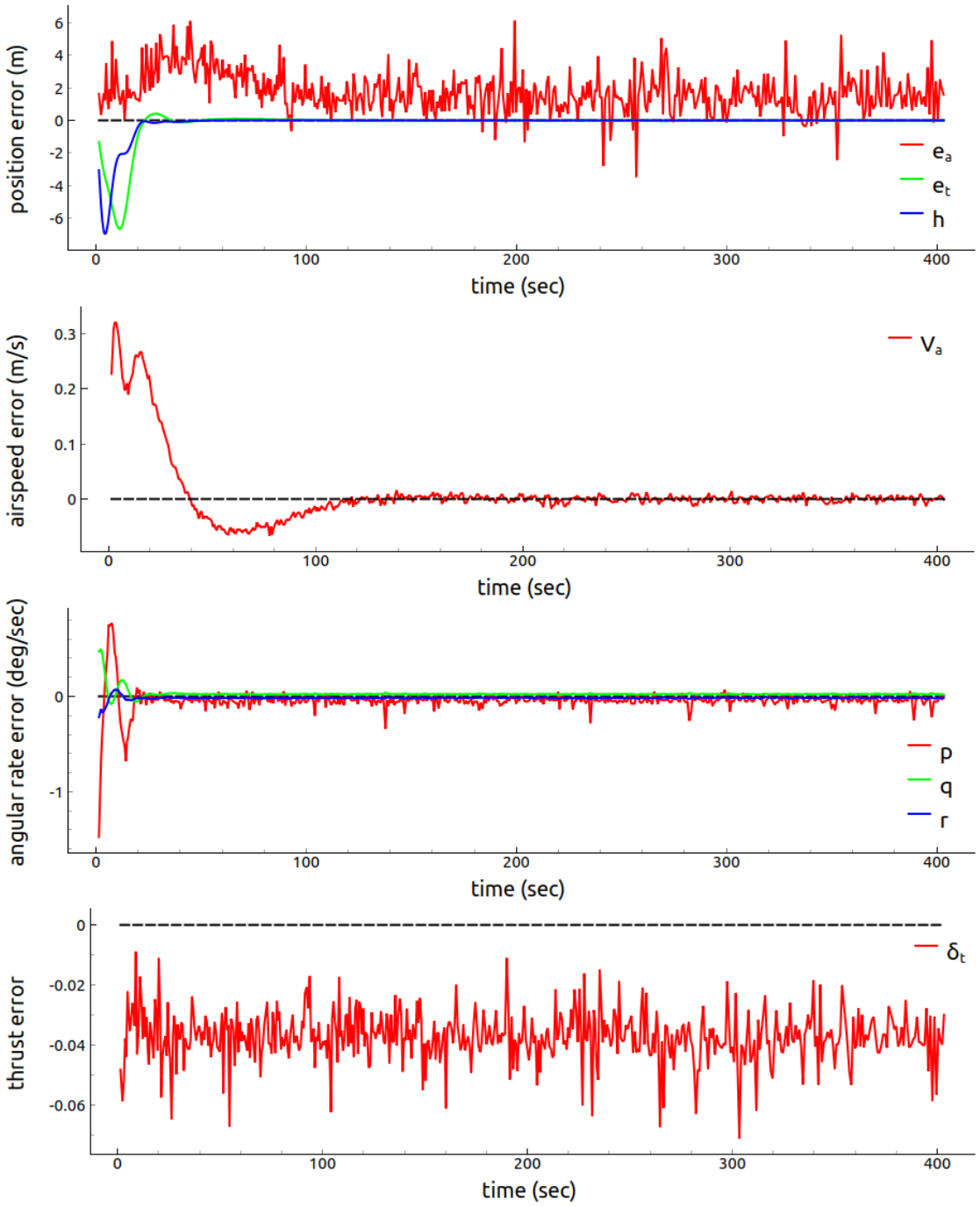


Figure 4.3: Error between UAS position, airspeed and input states predicted by the differential-flatness model and those of the desired trajectory. The aircraft successfully converges to the desired orbit trajectory and predicts inputs relatively well. The noisiness of the axial position error is likely due to the LQR controller being linearized about level flight conditions while being commanded to an orbit.

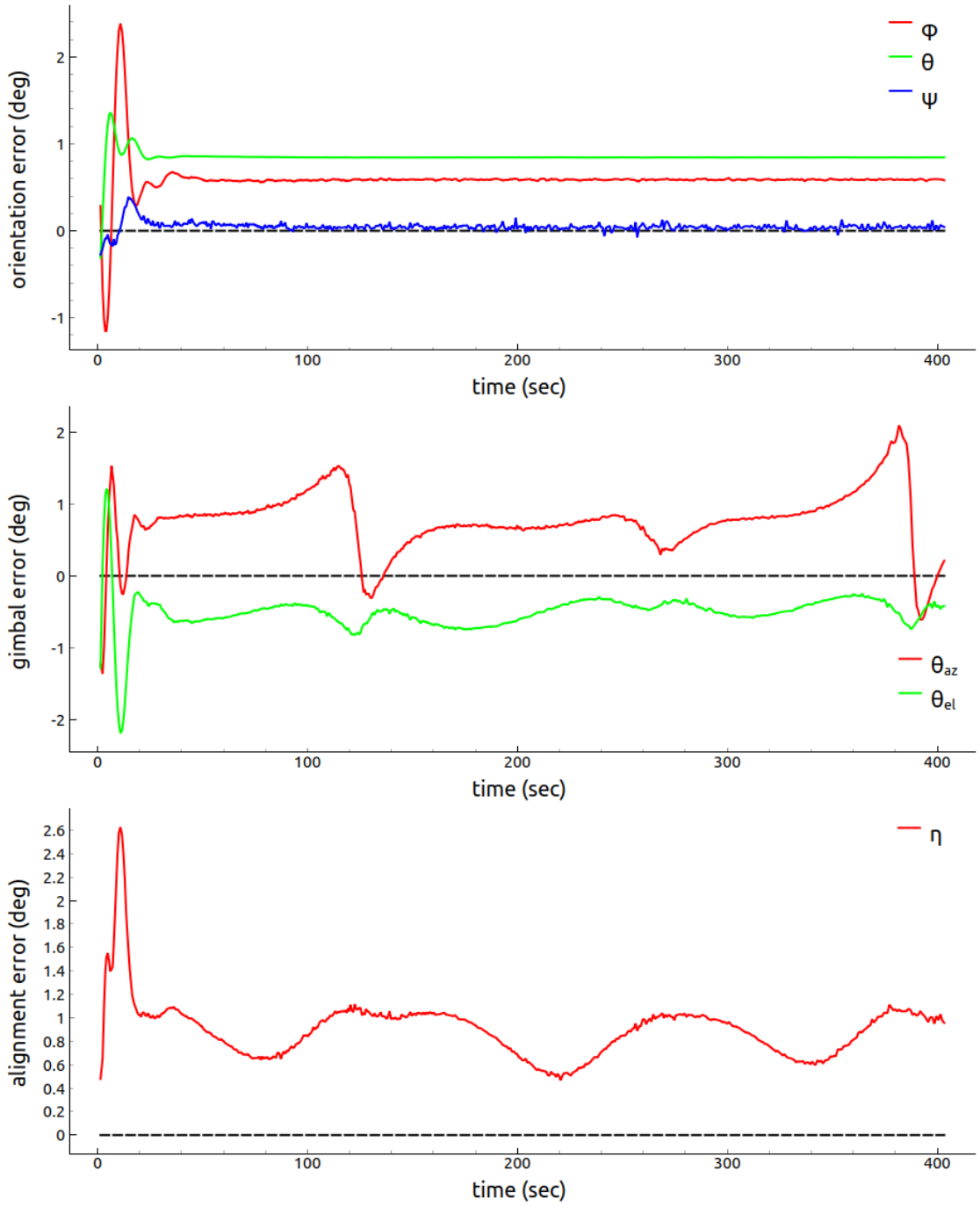


Figure 4.4: Error between UAS and gimbal orientation states predicted by the differential-flatness model and those required to align the ground target with the camera optical axis. Gimbal prediction error does not exceed ± 2.0 degrees after UAS state initialization error is overcome. Alignment η does not exceed 1.2 degree after overcoming initialization error.

should be noted that some misalignment between the target and the optical axis may not impact target visibility depending on the camera field-of-view parameters. For this simulation, a camera with a minimum field-of-view angle of 2 degrees would have maintained target visibility through the duration of the trajectory if predicted gimbal angles were used.

4.6 Conclusion

In this paper, we presented a differential-flatness model for a generalized fixed-wing aircraft with a two-axis gimbal camera tracking a known target. We showed how the model provides a mapping of a UAS trajectory to both the UAS and gimbal control spaces. The predictive accuracy and limitations of this model relative to the output of the same model tracking an inertial trajectory with LQR feedback control were shown in simulation. Future work could include defining uncertainty bounds for simplifications of the flatness model and accounting for the full visual field of the camera in viewing the target instead of constraining the target to align with the optical axis. Additional work in accounting for constant inertial wind and more accurate airspeed influences on angle-of-attack in the differential flatness model could improve the predictive accuracy of the model. Further analysis could be done to assess the impact of eliminating the angle-of-attack and control deflection terms of the drag model. A path planning framework could also be developed using the proposed differential-flatness model for autonomous target tracking with visualization guarantees.

CHAPTER 5. FLATNESS-BASED PATH PLANNING OF A GIMBALLED FIXED-WING UAS[1]

5.1 Introduction

Ground target detection and observation missions have increasingly sought to utilize unmanned aircraft systems (UAS) equipped with a downward-facing, gimballed camera system for both autonomous and semi-autonomous missions. These observation missions have applications in search and rescue, law enforcement, national security, border patrol, reconnaissance, and wildlife management. When a detected target has been selected for further observation, fixed-wing aircraft typically transition into an overhead orbit over the intended target. In many cases, maintaining target visibility during this transitioning trajectory can be critical. Without sufficiently defined vision-based trajectory constraints, tracked targets may escape UAS visibility long enough to evade further detection and localization. The interaction between gimbal constraints and flight dynamics is particularly challenging for fixed-wing UAS as opposed to multirotors due to fewer degrees of freedom in the dynamics.

Many works have developed approaches for controlling a UAS so that vision constraints are satisfied. In [12], a reactive control law successfully guides a fixed-wing aircraft to land on a linear runway while maintaining runway visibility. However, the method in [12] is designed specifically for tracking linear structures rather than generalized targets. Another approach is considered in [9] for maintaining a moving target in the camera field-of-view (FOV) with wind disturbances by controlling and constraining the UAS roll angle toward steady orbits. This approach is effective for transitioning to constant orbits about stationary targets, but was not designed to ensure visibility for gimbal angle limits commonly found on inexpensive off-the-shelf gimbals. The authors in [13] use game theoretic and stochastic optimal control for maintaining a moving, stochastic point target within a constrained FOV. The results of [13] show successful reactive control of a fixed-wing

[1]THIS WORK IS BEING SUBMITTED FOR PUBLICATION IN THE 2021 AIAA JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS (JGCD)

aircraft in following and maintaining line-of-sight of a non-holonomic ground vehicle; however, the control design does not attempt to predict future trajectory points where target visibility may be dynamically unattainable. By proving that a UAS system is differentially flat, the authors of [14]–[20] develop frameworks for successfully planning dynamically feasible trajectories for UAS. However, these works do not account for feasibility of gimbal states or target visibility throughout the trajectory in their flatness models. In [32] a control law is introduced for converging to an orbit over a target under input saturation constraints, however, input saturation constraints on a gimbal for maintaining visibility of a target are not considered. The authors of [33]–[36] develop various additional control laws that account for input saturation constraints, but again gimbal angle constraints are not considered for target tracking.

The main contribution of this work is a path planning framework for a fixed-wing UAS equipped with a gimballed camera system that ensures visibility while transitioning to an overhead orbit. The differential-flatness model for gimballed, fixed-wing aircraft used to drive the path parameterization under visibility constraints is summarized in Section 5.2. A simplified form of the final bi-tangent trajectory model is geometrically derived in Section 5.3. The bi-tangent model is extended with smoothing segments to ensure dynamic feasibility in Section 5.4. An additional path planning framework for dynamically feasible smooth transitions between orbits is shown in Section 5.5. A simulation study is presented in Section 5.6 that demonstrates the effectiveness of the proposed planning frameworks in predicting and maintaining target visibility throughout the entire trajectory. Conclusions are provided in Section 5.7.

5.2 Differential-flatness Model for Gimballed Fixed-Wing UAS

5.2.1 Mathematical notation

Coordinate frames are right-hand and denoted as $\mathscr{F}^a = \{\mathbf{i}_a, \mathbf{j}_a, \mathbf{k}_a\}$ where $\mathbf{i}_a$, $\mathbf{j}_a$, and $\mathbf{k}_a$ are the basis vectors. The expression $\mathbf{x}^c_{a/b}$ denotes a vector $\mathbf{x}$ of frame $\mathscr{F}^a$ measured with respect to frame $\mathscr{F}^b$ expressed in frame $\mathscr{F}^c$. Let $R^b_a \in SO(3)$ denote the rotation matrix that transforms coordinates in frame $\mathscr{F}^a$ to coordinates in frame $\mathscr{F}^b$. The canonical basis vectors in $\mathbb{R}^3$ are denoted by $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$. The frames $\mathscr{F}^i$, $\mathscr{F}^b$, $\mathscr{F}^g$, $\mathscr{F}^s$ and $\mathscr{F}^t$ denote the inertial, body-fixed, gimbal, stability and target reference frames respectively. Let $\lfloor \mathbf{x} \rfloor$ define the skew-symmetric matrix operator

on the generalized vector $\mathbf{x} = (x_1, x_2, x_3)^\top \in \mathbb{R}^3$ such that

$$\lfloor \mathbf{x} \rfloor = \begin{pmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{pmatrix}.$$

5.2.2 Gimballed Fixed-wing Aircraft Model

The simplified equations of motion for a fixed-wing aircraft are given by

$$\dot{\mathbf{p}}^i_{b/i} = \mathbf{v}^i_{b/i} \tag{5.1}$$

$$\dot{\mathbf{v}}^i_{b/i} = g\mathbf{e}_3 + \frac{1}{\mathrm{m}} R^i_s \left((T\cos(\alpha) - D(\mathbf{v}^i_{b/i}, \omega^b_{b/i}))\mathbf{e}_1 - L(\mathbf{v}^i_{b/i}, \omega^b_{b/i})\mathbf{e}_3 \right) \tag{5.2}$$

$$\dot{R}^i_b = R^i_b \lfloor \omega^b_{b/i} \rfloor \tag{5.3}$$

$$\omega^b_{b/i} = \omega_c, \tag{5.4}$$

where $\mathbf{p}^i_{b/i}$ is the UAS position, $\mathbf{v}^i_{b/i}$ is the UAS velocity, g is the acceleration due to gravity, m is the system mass, $\omega^b_{b/i}$ is the angular velocity of the UAS, ω_c are the commanded angular rates, T is the commanded thrust along the body $\mathbf{i}_b$ axis, α is the angle of attack,

$$D(\mathbf{v}^i_{b/i}, \omega^b_{b/i}) = \frac{1}{2}\rho V_a^2 S(C_{D_0} + C_{D_q}\frac{c}{2V_a}{\omega^b_{b/i}}^\top \mathbf{e}_2), \tag{5.5}$$

is the drag on the aircraft along the negative $\mathbf{i}_b$ body-fixed axis,

$$L(\mathbf{v}^i_{b/i}, \omega^b_{b/i}) = \frac{1}{2}\rho V_a^2 S(C_L(\alpha) + C_{L_q}\frac{c}{2V_a}{\omega^b_{b/i}}^\top \mathbf{e}_2), \tag{5.6}$$

is the lift force of the aircraft along the negative $\mathbf{k}_b$ body-fixed axis, where ρ is the air density,

$$V_a = \left\| \mathbf{v}^i_{b/i} \right\| \tag{5.7}$$

is the UAS airspeed, S is the planform area of the wing, c is the mean cord width of the wing, C_{D_0}, C_{D_q}, and C_{L_q} are aerodynamic coefficients, and $C_L(\alpha)$ is the lift curve of the wing, as defined

in [31]. To focus on relevant quantities and thereby simplify the model we have assumed that the rudder deflection, side-slip angle and deviation from the stable angle of attack α_0 are relatively small and that the drag induced by the control surfaces is negligible.

We assume that the UAS is equipped with a two-axis gimballed camera as defined in [38], where the gimbal azimuth θ_{az} and elevation θ_{el} angles are assumed to be commanded directly using a rapid-response inner control loop. The objective is to image and follow a known ground target with kinematics defined by

$$\dot{\mathbf{p}}^i_{t/i} = \mathbf{v}^i_{t/i}, \tag{5.8}$$

where $\mathbf{p}^i_{t/i}$ is the position of the target to be tracked and $\mathbf{v}^i_{t/i}$ is the target velocity.

5.2.3 Differential-flatness Model

In this section, we show that the combined UAS and gimbal system is differentially flat. The following definition is given in [37].

Definition 1 *The general nonlinear system* $\dot{\mathbf{x}} = f(\mathbf{x},\mathbf{u})$*, where* $\mathbf{x}$ *is the state and* $\mathbf{u}$ *is the control input, with the output mapping* $\mathbf{z} = h(\mathbf{x})$*, is differentially flat with the flat output* $\mathbf{z}$ *if the components of* $\mathbf{z}$ *are not differentially related over time, and if the state* $\mathbf{x}$ *and control inputs* $\mathbf{u}$ *can be written as a function of the flat output and its derivatives, i.e., there exists functions* g_1, g_e*, and finite scalars* m_1, m_2 *such that*

$$\mathbf{x}(t) = g_1\left(\mathbf{z}(t), \frac{d}{dt}\mathbf{z}(t), \cdots, \frac{d^{m_1}}{dt^{m_1}}\mathbf{z}(t)\right)$$
$$\mathbf{u}(t) = g_2\left(\mathbf{z}(t), \frac{d}{dt}\mathbf{z}(t), \cdots, \frac{d^{m_2}}{dt^{m_2}}\mathbf{z}(t)\right).$$

Therefore, differentially flat systems admit a direct algebraic mapping from the flat outputs to the states and control inputs. This mapping allows trajectory planning to be performed in the lower-dimensional flat output space, given an open-loop feedforward solution to the states and control inputs along the planned trajectory.

Using the fixed-wing model proposed in Section 5.2.2, [39] shows that the UAS/gimbal system is differentially flat with flat output given by the UAS and target positions by the following theorem.

Theorem 3 ([39]) *If (1) there is no wind, (2) the angle-of-attack is constant and equal to* α_0*, (3) the side-slip angle is zero, and (4) the UAS/target line-of-sight vector is aligned with the camera optical axis, then the aircraft defined by Eqs.* (5.1) *to* (5.5)*,* (5.7) *and* (5.8) *with states and inputs given by*

$$\mathbf{x} = \{\mathbf{p}^i_{b/i}, \mathbf{v}^i_{b/i}, R^i_b, \omega^i_{b/i}\} \in \mathbb{R}^3 \times \mathbb{R}^3 \times SO(3) \times \mathbb{R}^3$$
$$\mathbf{u} = \left(\omega^{b\top}_{b/i}, T, \theta_{az}, \theta_{el}\right)^\top \in \mathbb{R}^6,$$

is differentially flat with flat output

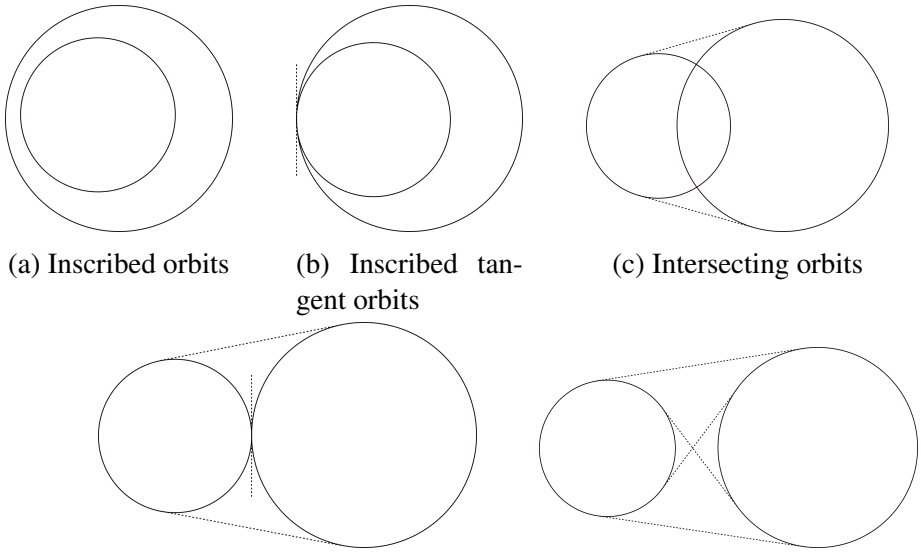
$$\mathbf{z} \triangleq \left(\mathbf{z}_b^\top, \quad \mathbf{z}_t^\top\right)^\top = \left(\mathbf{p}^{i\top}_{b/i}, \quad \mathbf{p}^{i\top}_{t/i}\right)^\top \in \mathbb{R}^6,$$

where g_1 *and* g_2 *are functions of* $\mathbf{z}_b$*,* $\dot{\mathbf{z}}_b$*,* $\ddot{\mathbf{z}}_b$*,* $\dddot{\mathbf{z}}_b$ *and* $\mathbf{z}_t$*.*

By Theorem 3 we then know that a direct algebraic mapping is available for assessing the dynamic and visual feasibility of a fixed-wing trajectory if it can be defined up to its third positional derivative and the inertial position of the target is known. The input space of the trajectory defined by the differential-flatness map can then be explored and augmented to ensure maintaining target visibility is possible over the proposed trajectory.

5.3 Simple Bi-tangent Orbit Transition

In order to plan a visually and dynamically feasible trajectory which converges into a desired orbit, we proceed to develop a generalization of the classical Dubin's path that allows for independently variable arc radii. We will refer to these as simple bi-tangent trajectories. By defining the second arc of this simple bi-tangent path to be the desired orbit, appropriate conditions can be calculated for the initial arc and subsequent linear segment of the path needed to converge to the desired final orbit.



(a) Inscribed orbits (b) Inscribed tangent orbits (c) Intersecting orbits

(d) External tangent orbits (e) External orbits

Figure 5.1: Simple bi-tangent configuration variations. In (a) there are no bi-tangent solutions, in (b) there is only one bi-tangent solution, with (c) there are exactly two, with (d) there are three, and in (e) there are four bi-tangent solutions.

5.3.1 Bi-tangent Line Solutions

Assuming a north-east-down (NED) inertial reference frame, we begin by solving for all possible angles at which a bi-tangent line connects to a given initial and final orbit pair. The number of tangent solutions to both orbits depends on the orbit sizes and relative locations as shown in Fig. 5.1.

For each of the five configurations shown in Fig. 5.1, a different set of bi-tangent line solutions exists. Inscribed orbits illustrated by Fig. 5.1(a) occur when the orbital centers and radii produce the relation

$$\|x_0 - x_1\| < |r_1 - r_0| \,.$$

For inscribed orbits, it is clear that no bi-tangent line solutions can exist between the two orbits. Inscribed tangent orbits illustrated by Fig. 5.1(b) exist when the orbital centers and radii produce the relation

$$\|x_0 - x_1\| = |r_1 - r_0| \,.$$

Inscribed tangent orbits have one unique outer bi-tangent line solution at the point of intersection between the two circles, though the bi-tangent line is infinitesimally small in length. One special case of inscribed tangent orbits occurs with concentric, equiradial orbits. In this case, there exists an infinite number of unique outer bi-tangent line solutions for every angle of the orbit, since the two circles intersect at every point; however, this case is trivial as it implies that an orbit has already been acquired and no transitioning trajectory is needed. Intersecting orbits illustrated by Fig. 5.1(c) exist when the orbital centers and radii produce the relation

$$|r_1 + r_0| > \|x_0 - x_1\| > |r_1 - r_0| \,.$$

Intersecting orbits have two unique outer bi-tangent line solutions. External tangent orbits illustrated by Fig. 5.1(d) exist when the orbital centers and radii produce the relation

$$|r_1 + r_0| = \|x_0 - x_1\| > |r_1 - r_0| \,.$$

External tangent orbits have two unique outer bi-tangent line solutions as well as a single unique inner bi-tangent line solution at the point of intersection between the two orbits. External orbits illustrated by Fig. 5.1(e) exist when the orbital centers and radii produce the relation

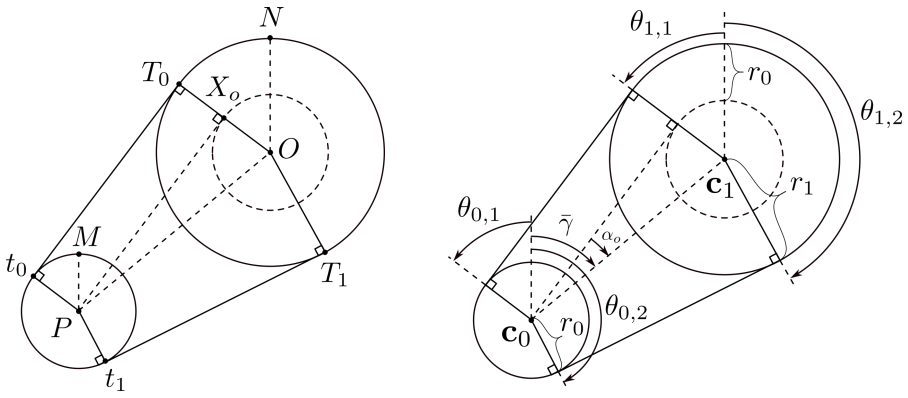
$$\|x_0 - x_1\| > |r_1 + r_0| \,.$$

For external orbits, two unique outer bi-tangent line solutions exist as well as a two unique inner bi-tangent line solutions.

Note that Fig. 5.1(e) has a pair of bi-tangent lines whose intersection lies between the two circles and a pair of bi-tangent lines whose intersection lies outside of the two circles. The next two lemmas provide expressions for these ”inner” and ”outer” bi-tangent solutions.

Lemma 2 *Given two orbits of radius r_0 and r_1 centered at points $\mathbf{c}_0 = (c_{00}, c_{01})^\top$ and $\mathbf{c}_1 = (c_{10}, c_{11})^\top$ respectively, as shown in Fig. 5.2, and let*

$$\bar{\gamma} = \arctan 2 \left(c_{11} - c_{01}, c_{10} - c_{00} \right) . \tag{5.9}$$



(a) Outer tangent geometric view

(b) Outer tangent angle relations

Figure 5.2: Outer bi-tangent geometry.

be the angle of the line connect the orbit centers, and let

$$\alpha_o = \arcsin\left(\frac{r_1 - r_0}{\|\mathbf{c}_1 - \mathbf{c}_0\|}\right) \tag{5.10}$$

be the angle between the line intersecting the orbit centers and the line that passes through $\mathbf{c}_0$ *and is tangent to the circle of radius* $r_1 - r_0$ *centered at* $\mathbf{c}_1$*. Then the angles where the outer tangent lines intersect the orbits, as shown in Fig. 5.2, are given by*

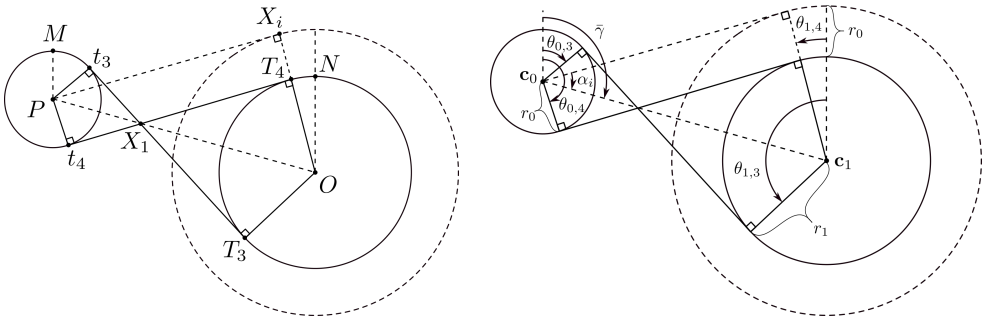
$$\theta_{0,1} = \theta_{1,1} = \bar{\gamma} - \frac{\pi}{2} - \alpha_o \tag{5.11}$$

$$\theta_{0,2} = \theta_{1,2} = \bar{\gamma} + \frac{\pi}{2} + \alpha_o. \tag{5.12}$$

Proof:

To solve for the angle $\angle NOT_0$, defined as $\theta_{1,1}$, we initially solve for the angle $\angle MPO$, defined as $\bar{\gamma}$, with Eq. (5.9). As shown in Fig. 5.2(a), we define a line $\overline{PX_o}$ parallel to $\overline{t_0T_0}$ that intersects the center of the orbit centered at P and lies tangent to a sub-orbit centered at O with radius $r_1 - r_0$. The angle $\angle X_oPO$, defined as α_o, is then described by Eq. (5.10).

For determining the angle $\angle NOT_0$, defined as $\theta_{1,1}$, we begin by noting that $\angle MPO + \angle PON = \pi$. Since $\angle PON = \angle POT_0 + \angle T_0ON_o$, with $\angle POT_0 = \frac{\pi}{2} - \alpha_o$, we then have $\angle T_0ON = \frac{\pi}{2} - \bar{\gamma} + \alpha_o$. We solve for the angle of interest $\angle NOT_0 = -\angle T_0ON$ defining $\theta_{1,1}$ in Eq. (5.11). It is clear that $\angle POT_0 = \angle T_1OP$ and, therefore, $\angle NOT_1 = 2\pi + \angle NOT_0 - 2\angle POT_0$ defining $\theta_{1,2}$



(a) Inner tangent geometric view (b) Inner tangent angle relations

Figure 5.3: Inner bi-tangent geometry

in Eq. (5.12). Since $\angle MPt_0$ and $\angle NOT_0$ are congruent angles of similar triangles, as are $\angle MPt_1$ and $\angle NOT_1$, we define $\theta_{0,1}$ and $\theta_{0,2}$ by Eqs. (5.11) and (5.12).

■

Lemma 3 *Given two orbits of radius r_0 and r_1 centered at points $\mathbf{c}_0 = (c_{00}, c_{01})^\top$ and $\mathbf{c}_1 = (c_{10}, c_{11})^\top$ respectively, as shown in Fig. 5.3, and let $\bar{\gamma}$ defined by Eq.* (5.9) *be the angle of the line connect the orbit centers, and let*

$$\alpha_i = \arcsin\left(\frac{r_1 + r_0}{\|\mathbf{c}_1 - \mathbf{c}_0\|}\right) \tag{5.13}$$

be the angle between the line intersecting the orbit centers and the line that passes through $\mathbf{c}_0$ and is tangent to the circle of radius $r_1 + r_0$ centered at $\mathbf{c}_1$. Then the angles where the inner tangent lines intersect the orbits, as shown in Fig. 5.3, are given by

$$\theta_{0,3} = \bar{\gamma} - \frac{\pi}{2} + \alpha_i \tag{5.14}$$

$$\theta_{0,4} = \bar{\gamma} + \frac{\pi}{2} - \alpha_i \tag{5.15}$$

$$\theta_{1,3} = \bar{\gamma} + \frac{\pi}{2} + \alpha_i \tag{5.16}$$

$$\theta_{1,4} = \bar{\gamma} - \frac{\pi}{2} - \alpha_i. \tag{5.17}$$

Proof: Similar to proof of Lemma 2.

■

Using Lemmas 2 and 3, we can describe bi-tangent line solutions for each of the configurations shown in Fig. 5.1. We proceed to derive an approach for constructing simple bi-tangent trajectories for unmanned aircraft systems (UAS).

5.3.2 Constructing Simple Bi-tangent UAS Trajectories

For UAS trajectory planning, simple bi-tangent paths can be used to quickly define a trajectory transitioning from an initial state to a final orbit over a known ground target. We proceed to define a time-parameterized model for defining the inertial position and positional derivatives of a bi-tangent path.

We begin by defining orbit arc trajectories generally. For a given UAS position $\mathbf{p}$, course angle χ, orbit radius r and orbit direction

$$\lambda \triangleq \begin{cases} 1, & \text{clockwise orbit} \\ -1, & \text{counter-clockwise orbit} \end{cases},$$

it can be shown that a general orbit arc trajectory can be defined as

$$\mathbf{p}_{\text{orb}}(t, t_0, \mathbf{p}_0, r, \lambda, \chi_0, V_g) \triangleq \mathbf{p}_0 + \lambda r \begin{bmatrix} \sin\left(\frac{V_g}{\lambda r}(t-t_0)+\chi_0\right) - \sin(\chi_0) \\ -\cos\left(\frac{V_g}{\lambda r}(t-t_0)+\chi_0\right) + \cos(\chi_0) \\ 0 \end{bmatrix}. \tag{5.18}$$

The center of the orbit tangent to the UAS velocity is also given by

$$\mathbf{c}_o(\mathbf{p}, \chi, r, \lambda) \triangleq \mathbf{p} + r \begin{bmatrix} \cos\left(\chi + \lambda\frac{\pi}{2}\right) \\ \sin\left(\chi + \lambda\frac{\pi}{2}\right) \\ 0 \end{bmatrix}, \tag{5.19}$$

with orbit angle

$$\theta_{\text{orb}}(\chi, \lambda) \triangleq \chi - \lambda\frac{\pi}{2}. \tag{5.20}$$

It can also be shown that general linear path segments can be defined using a start position $\mathbf{p}_l$ and direction χ_l as

$$\mathbf{p}_{\text{lin}}(t,t_0,\mathbf{p}_l,\chi_l,V_g) \triangleq \mathbf{p}_l + V_g(t-t_0)\begin{bmatrix}\cos(\chi_l)\\ \sin(\chi_l)\\ 0\end{bmatrix}. \tag{5.21}$$

.

The differential-flatness model described in Section 5.2 requires up to third order derivatives of flat outputs. It can be shown that the k^{th} derivative of an orbital trajectory defined by Eq. (5.18) is

$$\mathbf{p}^{(k)}_{\text{orb}}(t,t_0,r,\lambda,\chi_0,V_g) \triangleq \frac{V_g^k}{(\lambda r)^{k-1}}\begin{bmatrix}\sin\left(\frac{V_g}{\lambda r}(t-t_0)+\chi_0+\frac{k\pi}{2}\right)\\ -\cos\left(\frac{V_g}{\lambda r}(t-t_0)+\chi_0+\frac{k\pi}{2}\right)\\ 0\end{bmatrix}. \tag{5.22}$$

It can also easily be shown that the k^{th} derivative of a linear trajectory defined by Eq. (5.21) is

$$\mathbf{p}^{(k)}_{\text{lin}}(\chi_l,V_g) \triangleq \begin{cases}\begin{bmatrix}V_g\cos(\chi_l), & V_g\sin(\chi_l), & 0\end{bmatrix}^\top & k=1\\ \begin{bmatrix}0, & 0, & 0\end{bmatrix}^\top & k>1\end{cases} \tag{5.23}$$

To define the full bi-tangent trajectory, we consider a UAS transitioning to a orbit of radius r_1 over a target with an inertial position $\mathbf{p}_t$. The orbit radius r_1 is selected by the operator to be larger than the minimum orbit radius defined using the coordinated turn model in [31] as

$$r_{\min}(V_g,\phi_{\max}) = \frac{V_g^2}{g\tan(\phi_{\max})}, \tag{5.24}$$

where $\phi_{\max}$ is the maximum allowable aircraft roll angle. Suppose a dynamically feasible initial orbit radius r_0 has been selected for the UAS. The UAS has the option of initially turning left into a counter-clockwise orbit or right into a clockwise orbit with orbit center positions $\mathbf{c}_l$ and $\mathbf{c}_r$ respectively as shown in Fig. 5.4. To parameterize the initial orbit arc, we must first select the initial orbit direction. When considering a UAS equipped with a two-axis gimbal camera system, the selection between a clockwise or counter-clockwise orbit ultimately rests upon the constraints of the gimbal azimuth limits. For inexpensive, off-the-shelf gimbal solutions, it is

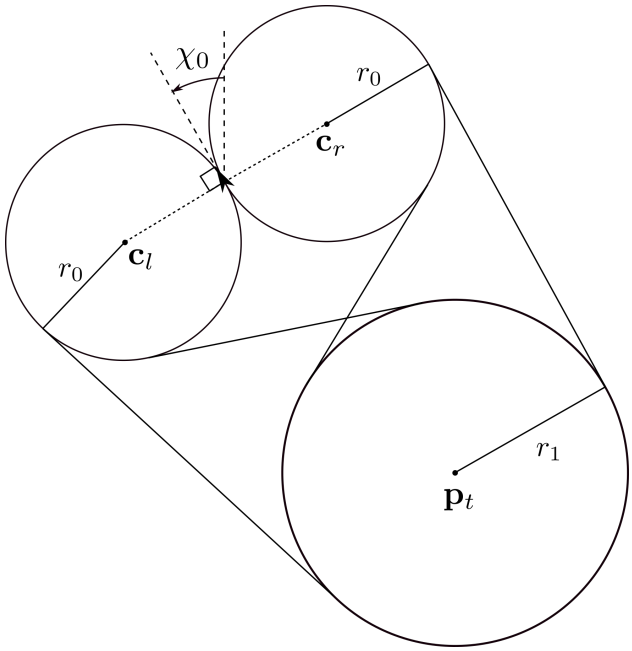


Figure 5.4: Typical simple bi-tangent orbit transition options for UAS transitioning to an overhead target orbit.

common for the azimuth axis to be actuated by a servo motor. These systems generally have hard rotational constraints preventing them from rotating beyond approximately ± 150 degrees. It then follows that gimbal azimuth limits are encountered whenever the target of interest crosses behind the UAS outside of the reach of the azimuth actuator. Intuitively, turning toward the direction of the current gimbal azimuth angle results in an orbit that allows the greatest range of free rotation before encountering azimuth limits. Assuming the target begins in the field-of-view, we then define the initial orbit direction using the initial azimuth angle $\theta_{\text{az},0}$ to be

$$\lambda_0(\theta_{\text{az},0}) = \begin{cases} 1, & \text{if } 0 \leq \theta_{\text{az},0} < \pi \\ -1, & \text{if } 0 > \theta_{\text{az},0} > -\pi \end{cases}. \tag{5.25}$$

Once an initial orbit direction has been selected, all possible bi-tangent exit angles for the initial orbit can be determined using Lemmas 2 and 3. It is important, however, to distinguish that some bi-tangent solutions may be considered invalid for a selected initial orbit direction. As derived, Table 5.1 shows all valid exit and entry angles θ_{ex} and θ_{en} respectively given an initial orbit direction. It should be noted that the final orbit direction is determined by the selection of

Table 5.1: Bi-tangent Exit and Entry Angles

Initial Orbit Direction	Outer Bi-tangent $(\theta_{ex}/\theta_{en})$	Inner Bi-tangent $(\theta_{ex}/\theta_{en})$
$\lambda_0 = 1$	$(\theta_{0,1}/\theta_{1,1})$	$(\theta_{0,3}/\theta_{1,3})$
$\lambda_0 = -1$	$(\theta_{0,2}/\theta_{1,2})$	$(\theta_{0,4}/\theta_{1,4})$

either inner or outer bi-tangent paths to be

$$\lambda_1 = \begin{cases} \lambda_0, & \text{outer bi-tangent} \\ -\lambda_0, & \text{inner bi-tangent} \end{cases}. \tag{5.26}$$

To finalize the initial orbit arc trajectory, the duration $t_{f,o}$ of the orbit trajectory defined in Eq. (5.18) must be determined using the arc length of the path. It can be shown for an orbit direction, exiting orbit angle and initial orbit angle θ_{init} that the arc length of the initial orbit path is given by $\theta_{\text{arc}} = \lambda \langle 2\pi + \lambda_0(\theta_{\text{ex}} - \theta_{\text{init}}) \rangle$ where $\theta_{\text{init}}, \theta_{\text{ex}} \in [-\pi, \pi]$ and

$$\langle y \rangle \stackrel{\triangle}{=} y \bmod 2\pi.$$

The duration of the initial orbit arc is then

$$\tau_{f,o} = \frac{r_0}{V_g} \langle 2\pi + \lambda(\theta_{\text{ex}} - \theta_{\text{init}}) \rangle$$

where

$$\theta_{\text{init}} = \theta_{\text{orb}}(\chi_0, \lambda_0).$$

producing the initial orbit completion time

$$t_{f,o} = t_0 + \tau_{f,o}$$

To parameterize the linear segment of the simple bi-tangent path, we use the initial orbit center

$$\mathbf{c}_i = \mathbf{c}_o(\mathbf{p}_0, \chi_0, r_0, \lambda_0)$$

to calculate the starting position of the linear segment as

$$\mathbf{p}_{\text{ls}} = \mathbf{c}_i + r_0 \begin{bmatrix} \cos(\theta_{\text{ex}}) \\ \sin(\theta_{\text{ex}}) \\ 0 \end{bmatrix}. \tag{5.27}$$

Since the linear portion of the simple bi-tangent path is tangent to the initial orbit, its direction is defined as

$$\chi_l = \theta_{\text{ex}} + \lambda_0 \frac{\pi}{2}. \tag{5.28}$$

The end point of the linear path is calculated similarly as

$$\mathbf{p}_{\text{le}} = \bar{\mathbf{p}}_t + r_1 \begin{bmatrix} \cos(\theta_{\text{en}}) \\ \sin(\theta_{\text{en}}) \\ 0 \end{bmatrix}, \tag{5.29}$$

where $\bar{\mathbf{p}}_t$ is the target position projected onto the north-east plane at the UAS altitude given by

$$\bar{\mathbf{p}}_t = \mathbf{p}_t + \mathbf{e}_3(\mathbf{p}_0^\top - \mathbf{p}_t^\top)\mathbf{e}_3.$$

The linear segment duration is then

$$\tau_{f,l} = \frac{\|\mathbf{p}_{\text{le}} - \mathbf{p}_{\text{ls}}\|}{V_g}$$

with a total trajectory time

$$t_f = t_{f,o} + \tau_{f,l}.$$

We then define the full time-parameterized inertial position trajectory as

$$\mathbf{p}(t) = \begin{cases} \mathbf{p}_{\text{orb}}(t, t_0, \mathbf{p}_0, r_0, \lambda_0, \chi_0, V_g), & \text{if } t_0 \le t \le t_{f,o} \\ \mathbf{p}_{\text{lin}}(t, t_{f,o}, \chi_l, V_g), & \text{if } t_{f,o} < t \le t_f \\ \mathbf{p}_{\text{orb}}(t, t_f, \mathbf{p}_{\text{le}}, r_1, \lambda_1, \theta_{\text{en}} - \lambda\frac{\pi}{2}, V_g), & \text{if } t > t_f \end{cases} \tag{5.30}$$

with derivatives

$$\mathbf{p}^{(k)}(t) = \begin{cases} \mathbf{p}_{\text{orb}}^{(k)}(t, t_0, r_0, \lambda_0, \chi_0, V_g), & \text{if } t_0 \leq t \leq t_{f,o} \\ \mathbf{p}_{\text{lin}}^{(k)}(\chi_l, V_g), & \text{if } t_{f,o} < t \leq t_f \\ \mathbf{p}_{\text{orb}}^{(k)}(t, t_f, r_1, \lambda_1, \theta_{\text{en}} - \lambda_1 \frac{\pi}{2}, V_g), & \text{if } t > t_f \end{cases}. \tag{5.31}$$

5.4 Bi-tangent Trajectory Smoothing

The simple bi-tangent paths introduced in Section 5.3 are not dynamically feasible because they assume instantaneous changes in roll angle at the transitions between the orbits and straight-line segment. The purpose of his section is to define a smoothed variation of the simple bi-tangent path defined in Section 5.3. This is done by first defining a model for smooth lateral transition segments in Section 5.4.1. These segments are then inserted at transition points of the simple bi-tangent trajectory to eliminate roll angle discontinuities. The full smoothed bi-tangent trajectory and its derivatives are then defined for use with the differential-flatness model as described in Section 5.4.2. Initial orbit arc length and straight-line length parameters are then optimally selected for the smooth bi-tangent trajectory to ensure the final orbit is centered over the desired target. The initial orbit radius is then selected to ensure gimbal angles required to view the target can be reached throughout the trajectory as described in Section 5.4.3.

To derive the smoothed transition segment trajectory, we use the simplified kinematic model

$$\dot{\mathbf{p}} = \begin{bmatrix} V_g \cos\chi \cos\gamma \\ V_g \sin\chi \cos\gamma \\ -\sin\gamma \end{bmatrix} \tag{5.32}$$

$$\dot{\chi} = \frac{g}{V_g} \tan\phi \tag{5.33}$$

$$\dot{\phi} = u_1 \tag{5.34}$$

$$\dot{\gamma} = u_2 \tag{5.35}$$

where χ, γ, and ϕ represent vehicle course angle, flight path angle, and roll angle respectively with u_1 and u_2 being the desired control inputs. For consistency with the differential-flatness model in Section 5.2, it is assumed that there is no wind.

5.4.1 Smooth Lateral Transition Trajectory Segment

We begin by defining a discretely-sampled trajectory model for the lateral transition trajectory segments used to smooth a bi-tangent path. The dynamic discontinuities exist because we assume that the aircraft can instantaneously transition between the roll angles required for orbits and level-flight. To eliminate these discontinuities, we define a smooth transition trajectory by assuming a constant roll rate between adjacent path segments such that $u_1 = \pm\bar{\omega}_\phi$, where $\pm\bar{\omega}_\phi$ is a constant roll rate during the transition. In this section, we assume that $\dot{\gamma} = \gamma = 0$ to maintain level flight during lateral transition to get

$$\dot{\mathbf{p}}_{\text{lat}}(V_g, \chi) = \begin{pmatrix} V_g \cos\chi \\ V_g \sin\chi \\ 0 \end{pmatrix} \tag{5.36}$$

$$\dot{\chi}(V_g, \phi) = \frac{g}{V_g}\tan\phi \tag{5.37}$$

$$\dot{\phi}(\bar{\omega}_\phi) = \bar{\omega}_\phi \tag{5.38}$$

with derivatives along the trajectory given by

$$\ddot{\mathbf{p}}_{\text{lat}}(\chi, \phi) = \begin{pmatrix} -V_g\dot{\chi}\sin\chi \\ V_g\dot{\chi}\cos\chi \\ 0 \end{pmatrix} = \begin{pmatrix} -g\sin\chi\tan\phi \\ g\cos\chi\tan\phi \\ 0 \end{pmatrix} \tag{5.39}$$

$$\dddot{\mathbf{p}}_{\text{lat}}(V_g, \chi, \phi, \bar{\omega}_\phi) = \begin{pmatrix} -V_g(\dot{\chi}^2\cos\chi + \ddot{\chi}\sin\chi) \\ V_g(-\dot{\chi}^2\sin\chi + \ddot{\chi}\cos\chi) \\ 0 \end{pmatrix} = \begin{pmatrix} -\frac{g^2}{V_g}\cos\chi\tan^2\phi - g\bar{\omega}_\phi\sin\chi\sec^2\phi \\ -\frac{g^2}{V_g}\sin\chi\tan^2\phi + g\bar{\omega}_\phi\cos\chi\sec^2\phi \\ 0 \end{pmatrix} \tag{5.40}$$

where we have used the fact that $\ddot{\chi} = \frac{g}{V_g}\bar{\omega}_\phi \sec^2\phi$. Note that the trajectory and its first three derivatives at time t can be expressed in terms of the states and the input at time t.

The trajectory produced by this set of differential equations resembles that of a clothoid trajectory as shown in Fig. 5.5 which does not have an analytical solution but must be computed numerically. Let the full-state, lateral transition trajectory at discrete time intervals along the seg-

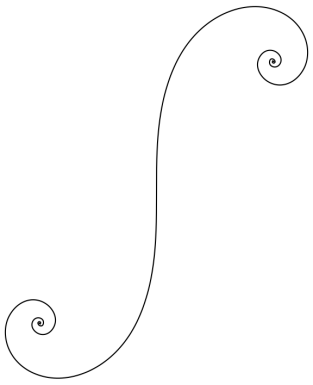


Figure 5.5: Inertial position of UAS obtained from RK45 propagation of the system described by Eqs. (5.36) to (5.38) from an an initial roll angle of -90 degrees to 90 degrees with a relatively small roll rate command. Note its resemblance to that of a classical clothoid curve.

ment be defined as

$$x_{\text{lat}} = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_{m_{\text{lat}}} \\ \chi_1 & \chi_2 & \cdots & \chi_{m_{\text{lat}}} \\ \phi_1 & \phi_2 & \cdots & \phi_{m_{\text{lat}}} \end{bmatrix}$$

where $\mathbf{p}_k$, χ_k, and ϕ_k are the position, course, and roll angle at step k along the segment, and where

$$[\mathbf{p}_k^\top, \chi_k, \phi_k]^\top = f_{\text{RK45}}([\mathbf{p}_{k-1}^\top, \chi_{k-1}, \phi_{k-1}]^\top, V_g, \bar{\omega}_\phi, T_s)$$

where f_{RK45} defines the RK45 propagation [40] of the system defined in Eqs. (5.36) to (5.38) over a small time step T_s with input $\pm\bar{\omega}_\phi$. We define the time t_{lat} needed to complete the transition trajectory as

$$t_{\text{lat}} = \frac{\phi_{m_{\text{lat}}} - \phi_1}{\bar{\omega}_\phi}$$

where $\phi_{m_{\text{lat}}}$ is the desired final roll angle. Using the coordinated turn model in [31], the roll angle for an orbit or radius r is given by

$$\phi_{\text{orb}}(\lambda, V_g, r, \gamma) \triangleq \arctan\left(\lambda \frac{V_g^2 \cos(\gamma)}{gr}\right). \tag{5.41}$$

The final roll angle of the transition path is then

$$\phi_{m_{\text{lat}}} = \phi_{\text{orb}}(\lambda, V_g, r, 0),$$

with the number of discrete propagation steps m_{lat} needed to define the desired transition trajectory calculated as

$$m_{\text{lat}} = \left\lceil \frac{t_{\text{lat}}}{T_s} \right\rceil, \tag{5.42}$$

where the notation $\lceil x \rceil$ is the integer ceiling operation on x. A lateral transition trajectory can then be defined using Algorithm 3, where the notation $x[:, j]$ represents the state of the trajectory at time index j.

Algorithm 3 Lateral Transition Segment Trajectory

procedure LATERAL TRANSITION($\mathbf{p}_0, \chi_0, \phi_0, V_g, \bar{\omega}_\phi, T_s, \phi_f$)
  $x_{\text{lat}}[:, 0] \leftarrow [\mathbf{p}_0^\top, \chi_0, \phi_0]^\top$
  Calculate $m_{\text{lat}} = \left\lceil \frac{(\phi_f - \phi_0)}{\bar{\omega}_\phi T_s} \right\rceil$
  $i \leftarrow 1$
  while $i <= m_{\text{lat}}$ **do**
    $x_{\text{lat}}[:, i] \leftarrow f_{\text{RK45}}(x_{\text{lat}}[:, i-1], V_g, \bar{\omega}_\phi, T_s)$
    $i \leftarrow i + 1$
  end while
  return x_{lat}, m_{lat}
end procedure

The derivatives of the the trajectory can be computed using equations Eqs. (5.36), (5.39) and (5.40).

5.4.2 Smoothed Bi-tangent Construction

A smoothed trajectory is constructed by inserting transition segments defined in the previous section, between the straight-line and circle segments of the simple bi-tangent trajectory. A simple bi-tangent trajectory is composed of (i) an initial circular arc, (ii) a straight line, and (iii) a final circular arc. Accordingly, with reference to Figure 5.6, a *smoothed bi-tangent trajectory* will consist of six segments: (I) a transition to enter the first orbit, (II) a circular arc along the initial orbit, (III) a transition from the first orbit to a straight-line segment, (IV) a straight-line segment, (V) a transition from the straight-line segment to the final orbit, and (VI) a final orbit. In

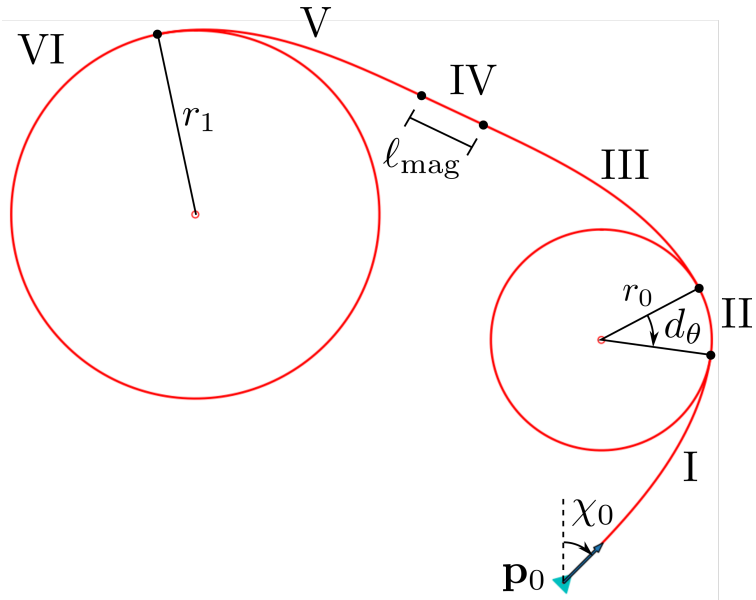


Figure 5.6: Bi-tangent trajectory with transition smoothing elements. Transition segment rates shown were selected to be very slow to exaggerate smoothing trajectory characteristics.

the following we parameterize each of these segments in order to explicitly define the smoothed bi-tangent trajectory. We will assume outer bi-tangent paths are used in the derivation since inner bi-tangents are often undefined at closer orbits, though the derivation can easily be extended to inner bi-tangent solutions as well.

Assuming initial aircraft states $\mathbf{p}_0$, χ_0, and ϕ_0, we calculate the transition to enter the first orbit (i.e. segment I from Fig. 5.6) using Algorithm 3 as

$$[x_{\text{lat,I}}, m_{\text{lat,I}}] \leftarrow \text{Lateral Transition}(\mathbf{p}_0, \chi_0, \phi_0, V_g, \lambda_0 \bar{\omega}_\phi, T_s, \phi_I) \tag{5.43}$$

where $\bar{\omega}_\phi$ is a constant roll rate and $\phi_I = \phi_{\text{orb}}(\lambda_0, V_g, r_0, 0)$ is the roll angle of the initial orbit calculated using Eq. (5.41). The endpoint states of segment I $[\mathbf{p}_{m_{\text{lat,I}}}^\top, \chi_{m_{\text{lat,I}}}, \phi_{m_{\text{lat,I}}}]^\top$ are reached at time $t_1 = t_0 + m_{\text{lat,I}} T_s$.

The arc along the first orbit (i.e. segment II from Fig. 5.6) can simply be parameterized using Eq. (5.18). Assuming a given orbit arc angle d_θ, segment II is completed at time $t_2 = t_1 + \frac{d_\theta r_0}{V_g}$ with endpoint states

$$\mathbf{p}_{\text{II}} = \mathbf{p}_{\text{orb}}(t_2, t_1, \mathbf{p}_{m_{\text{lat,I}}}, r_0, \lambda_0, \chi_{m_{\text{lat,I}}}, V_g)$$

$$\chi_{\text{II}} = \lambda_0 d_\theta + \chi_{m_{\text{lat,I}}}$$

$$\phi_{\text{II}} = \phi_{m_{\text{lat,I}}}$$

We define the transition from the first orbit to a straight-line segment (i.e. segment III from Fig. 5.6) as

$$[x_{\text{lat,III}}, m_{\text{lat,III}}] \leftarrow \text{Lateral Transition}(\mathbf{p}_{\text{II}}, \chi_{\text{II}}, \phi_{\text{II}}, V_g, -\lambda_0 \bar{\omega}_\phi, T_s, 0) \tag{5.44}$$

with endpoint states $[\mathbf{p}_{m_{\text{lat,III}}}^\top, \chi_{m_{\text{lat,III}}}, \phi_{m_{\text{lat,III}}}]^\top$ reached at time $t_3 = t_2 + m_{\text{lat,III}} T_s$.

The straight-line segment (i.e. segment IV from Fig. 5.6) can be parameterized using Eq. (5.21). Given a straight-line path length ℓ_{mag}, segment IV is completed at $t_4 = t_3 + \frac{\ell_{\text{mag}}}{V_g}$ with endpoint states

$$\mathbf{p}_{\text{IV}} = \mathbf{p}_{\text{lin}}(t_4, t_3, \chi_{m_{\text{lat,III}}}, V_g)$$

$$\chi_{\text{IV}} = \chi_{m_{\text{lat,III}}}$$

$$\phi_{\text{IV}} = 0$$

We then define the transition from the straight-line segment to the final orbit (i.e. segment V from Fig. 5.6) as

$$[x_{\text{lat,V}}, m_{\text{lat,V}}] \leftarrow \text{Lateral Transition}(\mathbf{p}_{\text{IV}}, \chi_{\text{IV}}, 0, V_g, \lambda_1 \bar{\omega}_\phi, T_s, \phi_V) \tag{5.45}$$

where $\phi_V = \phi_{\text{orb}}(\lambda_1, V_g, r_1, 0)$ is the roll angle of the final orbit calculated using Eq. (5.41) with endpoint states $[\mathbf{p}_{m_{\text{lat,V}}}^\top, \chi_{m_{\text{lat,V}}}, \phi_{m_{\text{lat,V}}}]^\top$ reached at time $t_5 = t_4 + m_{\text{lat,V}} T_s$.

We define the discrete-time sampling operator

$$\text{Sample}(f(t), t_0, t_f, T_s) = \begin{bmatrix} f(0) & f(T_s) & f(2T_s) & \dots & f(kT_s) \end{bmatrix}, \tag{5.46}$$

where t_0 is the start time, t_f is the end time, $f(t)$ is the continuous-time function to be sampled at a period T_s along t, and

$$k = \left\lfloor \frac{t_f - t_0}{T_s} \right\rfloor$$

is the number of sample points with the notation $\lfloor x \rfloor$ being the integer floor operation on x. Note that if $f(t)$ in Eq. (5.46) is not a function of time, then the result is simply a vector of repeated values or a matrix of repeated column vectors.

We then define the smooth bi-tangent as a discrete inertial position trajectory with course and roll angle endpoint states using Algorithm 4.

The first three derivatives of the trajectory points from Algorithm 4 can be obtained from Eqs. (5.22), (5.23), (5.36), (5.39) and (5.40) for use with the differential-flatness map. The endpoint states given by Algorithm 4 can be used to initialize the final orbit trajectory using Eqs. (5.18) and (5.22). Intermediate trajectory point states and derivatives can be approximated by linear interpolation.

5.4.3 Smoothed Bi-tangent Selection for Target Observation

The smoothed bi-tangent trajectories introduced by Algorithm 4 do not constrain the final orbit to be over a desired target. The parameterization for smoothed bi-tangent trajectories was obtained for an arbitrarily selected initial orbit radius, initial orbit arc length, and straight-line path length. The values of ℓ_{mag} and d_θ can be selected to align the final orbit over the desired target. Because segments I, III, and V are approximated numerically, these path parameters are optimally selected to minimize the north-east distance between the smoothed bi-tangent final orbit center and the target over which to orbit. The final orbit center is calculated using Eq. (5.19) as

$$\mathbf{c}_{\text{s},1} = \mathbf{c}_o(\mathbf{p}_{m_{\text{lat,V}}}, \chi_{m_{\text{lat,V}}}, r_1, \lambda_1), \tag{5.47}$$

where $\mathbf{p}_{m_{\text{lat,V}}}$ and $\chi_{m_{\text{lat,V}}}$ are the outputs from Algorithm 4.

Algorithm 4 Discrete Smooth Bi-tangent Trajectory

procedure SMOOTH BI-TANGENT($\mathbf{p}_0, \chi_0, \phi_0, V_g, \lambda_0, r_0, \lambda_1, r_1, \bar{\omega}_\phi, T_s, t_0, d_\theta, \ell_{\text{mag}}$)

▷ Segment I

$\phi_I \leftarrow \arctan\left(\lambda_0 \frac{V_g^2}{g r_0}\right)$

$\left[\left[\mathbf{p}_{\text{lat,I}}^\top, \chi_{\text{lat,I}}, \phi_{\text{lat,I}}\right]^\top, m_{\text{lat,I}}\right] \leftarrow$ Lateral Transition$(\mathbf{p}_0, \chi_0, \phi_0, V_g, \lambda_0 \bar{\omega}_\phi, T_s, \phi_I)$

▷ Segment II

$t_1 \leftarrow t_0 + m_{\text{lat,I}} T_s$

$t_2 \leftarrow t_1 + \frac{d_\theta r_0}{V_g}$

$\mathbf{p}_{\text{lat,II}} \leftarrow$ Sample $\left(\mathbf{p}_{\text{orb}}(t, t_1, \mathbf{p}_{\text{lat,I}}[:, m_{\text{lat,I}}], r_0, \lambda_0, \chi_{\text{lat,I}}[:, m_{\text{lat,I}}], V_g), t_1, t_2, T_s\right)$

$\mathbf{p}_{\text{II}} \leftarrow \mathbf{p}_{\text{orb}}(t_2, t_1, \mathbf{p}_{\text{lat,I}}[:, m_{\text{lat,I}}], r_0, \lambda_0, \chi_{\text{lat,I}}[:, m_{\text{lat,I}}], V_g)$

$\chi_{\text{II}} \leftarrow \lambda_0 d_\theta + \chi_{\text{lat,I}}[:, m_{\text{lat,I}}]$

$\phi_{\text{II}} \leftarrow \phi_{\text{I}}$

▷ Segment III

$\left[\left[\mathbf{p}_{\text{lat,III}}^\top, \chi_{\text{lat,III}}, \phi_{\text{lat,III}}\right]^\top, m_{\text{lat,III}}\right] \leftarrow$ Lateral Transition$(\mathbf{p}_{\text{II}}, \chi_{\text{II}}, \phi_{\text{II}}, V_g, -\lambda_0 \bar{\omega}_\phi, T_s, 0)$

▷ Segment IV

$t_3 \leftarrow t_2 + m_{\text{lat,III}} T_s$

$t_4 \leftarrow t_3 + \frac{\ell_{\text{mag}}}{V_g}$

$\mathbf{p}_{\text{lat,IV}} \leftarrow$ Sample $\left(\mathbf{p}_{\text{lin}}(t, t_3, \chi_{\text{lat,III}}[:, m_{\text{lat,III}}], V_g), t_3, t_4, T_s\right)$

$\mathbf{p}_{\text{IV}} \leftarrow \mathbf{p}_{\text{lin}}(t_4, t_3, \chi_{\text{lat,III}}[:, m_{\text{lat,III}}], V_g)$

$\chi_{\text{IV}} \leftarrow \chi_{\text{lat,III}}[:, m_{\text{lat,III}}]$

$\phi_V \leftarrow \arctan\left(\lambda_0 \frac{V_g^2}{g r_1}\right)$

▷ Segment V

$\left[\left[\mathbf{p}_{\text{lat,V}}^\top, \chi_{\text{lat,V}}, \phi_{\text{lat,V}}\right]^\top, m_{\text{lat,V}}\right] \leftarrow$ Lateral Transition$(\mathbf{p}_{\text{IV}}, \chi_{\text{IV}}, 0, V_g, \lambda_1 \bar{\omega}_\phi, T_s, \phi_V)$

▷ Combine Segments

$\mathbf{p}_{\text{sbt}} \leftarrow \left[\mathbf{p}_{\text{lat,I}} \mid \mathbf{p}_{\text{lat,II}} \mid \mathbf{p}_{\text{lat,III}} \mid \mathbf{p}_{\text{lat,IV}} \mid \mathbf{p}_{\text{lat,V}}\right]$

return $\mathbf{p}_{\text{sbt}}$, $\chi_{\text{lat,V}}[:, m_{\text{lat,V}}]$, ϕ_V

end procedure

We define the cost function to minimize the lateral distance between the target and final orbit center

$$J_s = \|I_2(\mathbf{p}_t - \mathbf{c}_{\mathrm{s},1})\|,$$

where $I_2 = \mathrm{diag}(1,1,0)$ which can be used to and define the nonlinear optimization problem

$$\begin{aligned} &\min_{\ell_{\mathrm{mag}}, d_\theta} J_s \\ &\text{s.t.} \\ &\ell_{\mathrm{mag}} \geq 0 \\ &0 \leq d_\theta < 2\pi \end{aligned} \tag{5.48}$$

This minimization can be solved using any optimization methodology that doesn't require an analytical Jacobian. In this paper, we used the sequential least-squares quadratic programming (SLSQP) method [41]. Due to the nonlinear nature of this optimization problem, it is important to seed the optimization near a known solution point to ensure convergence. For high roll rate commands, transition trajectories become infinitesimally small until the smoothed bi-tangent solution converges to the simple bi-tangent solution. This makes the initial arc length and straight-line path length parameters of the simple bi-tangent solution good candidates for seeding the optimization. The seed for the simple bi-tangent straight-line path length can be calculated from Eqs. (5.27) and (5.29) as

$$\ell_{\mathrm{mag,seed}} = \|\mathbf{p}_{\mathrm{le}} - \mathbf{p}_{\mathrm{ls}}\| \tag{5.49}$$

with the seed for the arc angle calculated as

$$d_{\theta,\mathrm{seed}} = \lambda_0 \langle 2\pi + \lambda_0(\theta_{\mathrm{ex}} - \theta_{\mathrm{init}}) \rangle . \tag{5.50}$$

An example smoothed bi-tangent trajectory obtained using this seeding approach for a generic set of initial UAS states, target states, and smoothing conditions is shown in Fig. 5.7. It is clear that for rapid roll rates, the smoothed bi-tangent solution is reasonably close to the simple bi-tangent solution. Since the smoothed bi-tangent trajectory was defined using appropriately selected roll rate commands, the smoothed bi-tangent path is dynamically feasible.

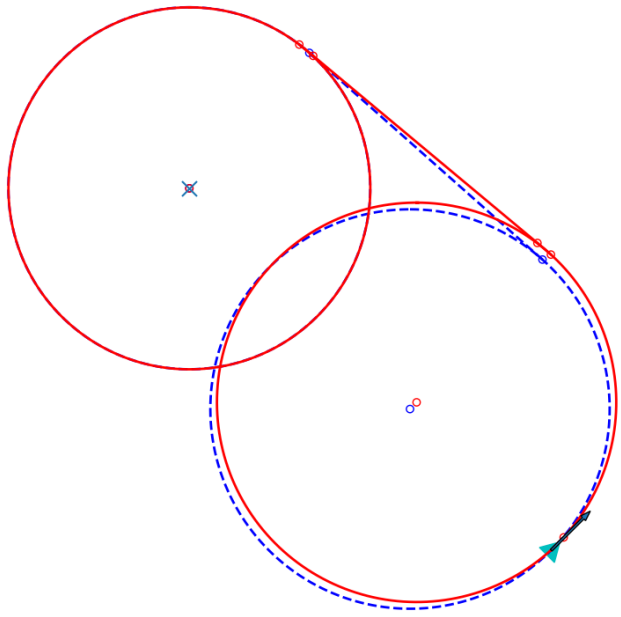


Figure 5.7: Simple bi-tangent trajectory overlaid with the smoothed bi-tangent solution. The simple bi-tangent solution is shown by dashed blue lines with the smoothed solution displayed in red. The cyan triangle marker indicates the UAS initial position with an arrow indicating its initial heading. The blue X marker indicates the known target over which a desired orbit is to be obtained.

Because the smooth transition segments are obtained numerically by discrete model propagation, computation of the segment can become computationally expensive when very small time steps are used to improve model accuracy. This calculation becomes far more expensive when the model is recalculated within an optimization loop as described above. It is possible to compute the transition path offline for significant computational performance gains as shown in Appendix A.

We proceed to consider parameter selection for ensuring visual feasibility of the observed target. The initial orbit radius can be selected to minimize path length while ensuring feasibility of gimbal angles required to align the target with the optical axis. This is done by passing discretely-sampled points along the trajectory through the differential-flatness map and ensuring that no gimbal inputs surpass the reachable bounds of the system. Discretely-sampled points from the smooth bi-tangent trajectory and their derivatives obtained from Algorithm 4 are passed through the differential-flatness map to obtain the flat input set U_{lat}. To ensure visual feasibility, the initial orbit radius is optimally selected to be minimized while producing a smooth bi-tangent solution with flat inputs on the gimbal that never surpass saturation limits. The values of ℓ_{mag} and d_θ are optimally selected at each iteration to align the final orbit over the target. This optimization is

seeded by the linear magnitude and arc angle of the simple bi-tangent solution. The full process for defining a smooth bi-tangent transition trajectory given initial UAS states and the position of a target over which to orbit is given by Algorithm 5.

Algorithm 5 as defined uses a simple line search to identify a small feasible initial orbit radius for the bi-tangent trajectory; however, the minimum initial orbit radius could be found using other optimization techniques such as a bisection search [42]. We proceed to identify conditions that guarantee the existence of a dynamically and visually feasible smooth bi-tangent solution.

Assuming no azimuth gimbal angle can rotate freely, the most significant cause of visual infeasibility for gimballed fixed-wing UAS occurs when aggressive roll angles at large distances from the observed target cause the gimbal elevation angle to saturate at its lower bound as illustrated by Fig. 5.8.

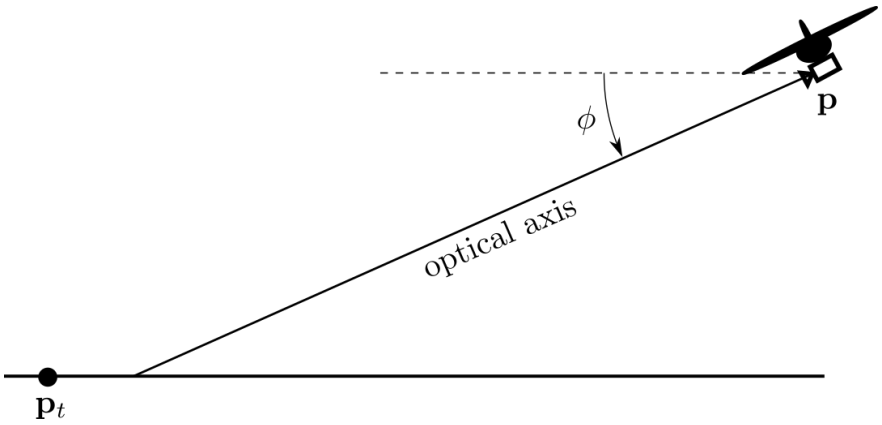


Figure 5.8: Elevation angle saturation resulting in an inability to align a target at $\mathbf{p}_t$ with the optical axis due to an aggressive roll angle at a large distance from the intended target. Assuming a minimum elevation angle of 0 degrees, this results in angle to the optical axis at elevation saturation being equal to the roll angle ϕ of the aircraft. Note that target alignment with the optical axis becomes possible for the case depicted only as the aircraft moves closer to the target or as the roll angle becomes smaller.

The following lemma defines sufficient conditions on the existence of a smooth bi-tangent trajectory with no elevation angle saturation.

Algorithm 5 Smooth Bi-tangent Planner

procedure SMOOTH BI-TANGENT PLANNER($\mathbf{p}_0, \chi_0, \phi_0, V_g, \theta_{\text{az},0}, \mathbf{p}_t$)
  Define the minimum roll rate magnitude $\bar{\omega}_\phi$
  Define the maximum roll angle $\phi_{\max}$
  Calculate the minimum orbit radius $r_{\min}(V_g, \phi_{\max})$ with Eq. (5.24)
  Define initial orbit radius search step r_{step}
  Define gimbal saturation limits $\theta_{\text{el,min}}, \theta_{\text{el,max}}, \theta_{\text{az,min}}, \theta_{\text{az,max}}$
  Calculate $\lambda_0(\theta_{\text{az},0})$ using Eq. (5.25)
  Select λ_1 using Eq. (5.26)
  Select final observation orbit radius $r_1 \geq r_{\min}$
  Select $T_s << 1$ (e.g. 0.01)
  $t_0 \leftarrow 0$
  $r_0 \leftarrow r_{\min}$
  $\Theta_{\text{el}} \leftarrow \{\}$
  $\Theta_{\text{az}} \leftarrow \{\}$
  while $\theta_{\text{el,min}} \leq \Theta_{\text{el}} \leq \theta_{\text{el,max}}$ and $\theta_{\text{az,min}} \leq \Theta_{\text{az}} \leq \theta_{\text{az,max}}$ **do**
    $\ell_{\text{mag}} \leftarrow \ell_{\text{mag,seed}}$ from Eq. (5.49)
    $d_\theta \leftarrow d_{\theta,\text{seed}}$ from Eq. (5.50)
    Solve for optimal $\ell_{\text{mag,opt}}$ and $d_{\theta,\text{opt}}$ to align over target for current r_0 using Eq. (5.48)
    $\mathbf{p}_{\text{traj}}, \chi_f, \phi_f \leftarrow$ Smooth Bi-tangent$(\mathbf{p}_0, \chi_0, \phi_0, V_g, \lambda_0, r_0, \lambda_1, r_1, \bar{\omega}_\phi, T_s, t_0, d_{\theta,\text{opt}}, \ell_{\text{mag,opt}})$
    Calculate $\dot{\mathbf{p}}_{\text{traj}}$, $\ddot{\mathbf{p}}_{\text{traj}}$, and $\mathbf{p}^{(3)}_{\text{traj}}$ using Eqs. (5.22), (5.23), (5.36), (5.39) and (5.40)
    Calculate sampled gimbal inputs $\Theta_{\text{el}}, \Theta_{\text{az}}$ using flatness proof in [39]
    $r_0 \leftarrow r_0 + r_{\text{step}}$
  end while
  return $\mathbf{p}_{\text{traj}}$, $\dot{\mathbf{p}}_{\text{traj}}$, $\ddot{\mathbf{p}}_{\text{traj}}$, $\dddot{\mathbf{p}}_{\text{traj}}$, χ_f, ϕ_f
end procedure

Lemma 4 *Given a gimballed UAS with inertial position* $\mathbf{p}$*, ground speed* V_g *and initial roll angle* ϕ_0 *observing a target at a lateral north-east plane distance of* $\bar{p}_t$ *and minimum allowable elevation angle* $\theta_{el,min}$*. Assuming a maximum roll rate* $\bar{\omega}_\phi$*, a smooth bi-tangent solution without elevation*

angle saturation exists if there is a solution $r_{0,min}$ to

$$\tan^{-1}\left(\frac{-\mathbf{p}^{\top}\mathbf{e}_3}{\sqrt{r_{0,min}^2+(V_g\frac{\phi_f-\phi_0}{\bar{\omega}_\phi}+\bar{p}_t)^2}+r_{0,min}}\right)-\theta_{el,min}-\tan^{-1}\left(\frac{V_g^2}{r_{0,min}g}\right)=0. \tag{5.51}$$

where g is the acceleration due to gravity. The solution $r_{0,min}$ also defines a lower bound on the initial orbit radius for maintaining gimbal elevation feasibility.

Proof: As illustrated by Fig. 5.8, saturation of the elevation angle while attempting to align a target with the optical axis is caused by aircraft roll angles at large lateral distances from the target. We produce a conservative estimate of the most distant roll position by assuming that the aircraft flies directly away from the target for the duration of the initial transition segment and then performs the initial orbit turn to return toward the intended final orbit. It is during the initial orbit turn that the UAS encounters its most distant position from the target where the roll angle is maximized as shown in Fig. 5.9. The initial smoothing segment of the proposed smooth bi-tangent model is

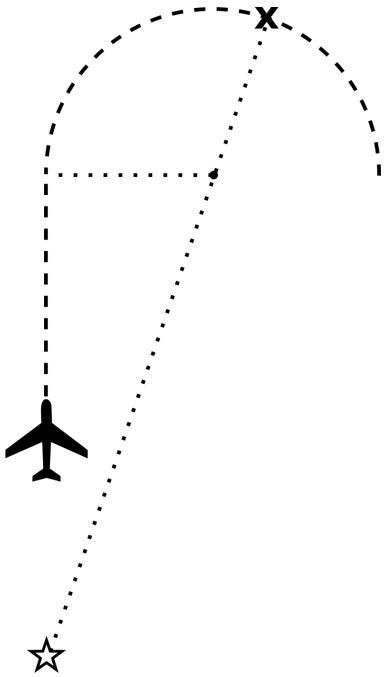


Figure 5.9: Conservative estimate model for farthest possible smooth bi-tangent trajectory position with greatest roll angle corresponding to the most likely location for gimbal elevation angle saturation. The star represents a potential target over which the UAS intends to orbit and the X represents the point on the trajectory where elevation angle saturation is most likely to occur.

approximated as a straight-line away from the intended target at nominal velocity for the duration of the transition. Since the true transition segment has continuously increasing curvature, it will remain at a closer proximity relative to this estimate. The aircraft then enters the initial orbit at the proposed radius and subsequently returns toward the target to enter the final orbit. As illustrated by Fig. 5.9, the most distant position where the roll angle is maximized will occur at the point of the initial orbit segment that intersects the line defined between the intended target and initial orbit center. The largest possible lateral distance between the target and the UAS trajectory is then

$$\ell_{\max} = \sqrt{r_0^2 + (V_g \Delta_{t,\text{trans}} + \bar{p}_t)^2} + r_0$$

where $\bar{p}_t$ is the initial lateral distance from the target and $\Delta_{t,\text{trans}}$ is the time required to execute the initial smooth bi-tangent transition segment calculated as

$$\Delta_{t,\text{trans}} = \frac{\phi_f - \phi_0}{\bar{\omega}_\phi}$$

with $\phi_f = \tan^{-1}\left(\frac{V_g^2}{r_0 g}\right)$ being the roll angle for executing the initial orbit.

For a given initial orbit radius, alignment between the target and optical axis is possible if the roll angle ϕ_f satisfies $phi_f < \phi_{\max}$, where $\phi_{\max}$ is the maximum allowable roll angle resulting in alignment of the target with the camera optical axis achieved by the smallest possible gimbal elevation angle $\theta_{\text{el,min}}$, where

$$\phi_{\max} = \tan^{-1}\left(\frac{h}{\ell_{\max}}\right) - \theta_{\text{el,min}}.$$

A conservative estimate of the minimum allowable initial orbit radius to sufficiently ensure no gimbal elevation saturation occurs is then the solution to $\phi_{\max} = \phi_f$, which is Eq. (5.51). ■

Due to the complicated interactions between visual boundaries produced by gimbal azimuth angle constraints and the lateral transition segments of the smoothed bi-tangent trajectory, sufficient conditions for preventing azimuth angle saturation become difficult to define and are therefore not developed in this work for brevity. Without conditions of sufficiency for azimuth-safe solutions, it is recommended to solve the proposed bi-tangent optimization offline to ensure

the optimization converges to a solution that meets all defined visual feasibility constraints. Especially in search and observation applications with fixed-wing UAS, it is common to execute a search in a constant configuration (e.g. constant velocity level flight with set gimbal angle offsets) making related applications good candidates for offline solution generation.

5.5 Smoothed Altitude Transition Trajectories

Once a final orbit has been acquired over a target, it may be necessary to transition to a different altitude for better target visibility or aircraft safety. The purpose of this section is to define feasible time-parameterized trajectories for performing this altitude transition. We first derive the positional states of a generic spiral trajectory between orbits of differing altitudes. To smooth flight path angle discontinuities at the beginning and end of the spiral model, transition trajectory segments for flight path angle are defined and then injected into the original spiral trajectory to provide a smooth transition path between altitudes. The flight path angle of the spiral segment is then optimally selected to ensure both dynamic and visual feasibility conditions are maintained throughout the trajectory.

5.5.1 Spiral Altitude Transitions

We begin by defining a simple approach for transitioning between two altitudes by adjusting the altitude linearly over time. This is equivalent to maintaining a constant, non-zero flight path angle γ_c. Performing this during a lateral orbit trajectory results in a spiral path. This spiral can be defined by first solving for the course angle trajectory by integrating Eq. (5.33) to produce

$$\chi_{\text{sp}}(t,t_0,\chi_0,\phi_0,V_g) = \chi_0 + \frac{g}{V_g}\tan(\phi_0)(t-t_0).$$

The initial roll angle $\phi_0 = \phi_{\text{orb}}(\lambda_1,V_g,r_1,\gamma_c)$ is found using Eq. (5.41), where γ_c is the desired transitioning flight path angle. The positional trajectory is then defined as

$$\mathbf{p}_{\text{sp}}(t,t_0,\mathbf{p}_0,\phi_0,\chi_0,\gamma_c,V_g) = \mathbf{p}_0 + \begin{bmatrix} \frac{V_g^2}{g}\frac{\cos(\gamma_c)}{\tan(\phi)}\left(\sin(\frac{g}{V_g}\tan(\phi)(t-t_0)+\chi_0)-\sin(\chi_0)\right) \\ -\frac{V_g^2}{g}\frac{\cos(\gamma_c)}{\tan(\phi)}\left(\cos(\frac{g}{V_g}\tan(\phi)(t-t_0)+\chi_0)-\cos(\chi_0)\right) \\ -V_g\sin(\gamma_c)(t-t_0) \end{bmatrix} \tag{5.52}$$

with derivatives

$$\dot{\mathbf{p}}_{\text{sp}}(t,t_0,\mathbf{p}_0,\phi_0,\chi_0,\gamma_c,V_g) = \begin{bmatrix} V_g\cos(\gamma_c)\cos(\chi_0+\frac{g}{V_g}\tan(\phi)(t-t_0)) \\ V_g\cos(\gamma_c)\sin(\chi_0+\frac{g}{V_g}\tan(\phi)(t-t_0)) \\ -V_g\sin(\gamma_c) \end{bmatrix} \tag{5.53}$$

$$\ddot{\mathbf{p}}_{\text{sp}}(t,t_0,\mathbf{p}_0,\phi_0,\chi_0,\gamma_c,V_g) = \begin{bmatrix} -g\cos(\gamma_c)\sin(\chi_0+\frac{g}{V_g}\tan(\phi)(t-t_0))\tan(\phi) \\ g\cos(\gamma_c)\cos(\chi_0+\frac{g}{V_g}\tan(\phi)(t-t_0))\tan(\phi) \\ 0 \end{bmatrix} \tag{5.54}$$

$$\dddot{\mathbf{p}}_{\text{sp}}(t,t_0,\mathbf{p}_0,\phi_0,\chi_0,\gamma_c,V_g) = \begin{bmatrix} -\frac{g^2}{V_g}\cos(\gamma_c)\cos(\chi_0+\frac{g}{V_g}\tan(\phi)(t-t_0))\tan^2(\phi) \\ -\frac{g^2}{V_g}\cos(\gamma_c)\sin(\chi_0+\frac{g}{V_g}\tan(\phi)(t-t_0))\tan^2(\phi) \\ 0 \end{bmatrix}. \tag{5.55}$$

As with the lateral simple bi-tangent trajectory model, this approach produces dynamically infeasible discontinuities in the aircraft flight path angle at the beginning and end of the spiral.

5.5.2 Flight Path Angle Transition Trajectories

To eliminate flight path angle discontinuities at the beginning and end of the simple spiral trajectory, we proceed to define flight path angle transition trajectories which will be used to create a smooth altitude transition trajectory. Using Eqs. (5.32) to (5.35) and assuming a constant flight path angle rate $\dot{\gamma} = \bar{\omega}_\gamma$, we define a longitudinal transition smoothing trajectory by commanding a constant flight path angle rate $\bar{\omega}_\gamma$. Using the coordinated turn definition, we calculate the roll rate needed to maintain a constant orbit radius during an altitude transition segment to be

$$\dot{\phi} = -\tan(\gamma)\tan(\phi)\cos^2(\phi)\dot{\gamma}.$$

Under these assumptions, we have the system

$$\dot{\mathbf{p}}_{\text{lon}}(V_g,\chi,\gamma) = \begin{bmatrix} V_g\cos\chi\cos\gamma \\ V_g\sin\chi\cos\gamma \\ -V_g\sin\gamma \end{bmatrix} \tag{5.56}$$

$$\dot{\chi}(V_g, \phi) = \frac{g}{V_g} \tan \phi \tag{5.57}$$

$$\dot{\phi}(\gamma, \phi) = -\tan(\gamma) \tan(\phi) \cos^2(\phi) \bar{\omega}_\gamma \tag{5.58}$$

$$\dot{\gamma}(\bar{\omega}_\gamma) = \bar{\omega}_\gamma \tag{5.59}$$

with derivatives defined as

$$\ddot{\mathbf{p}}_{\text{lon}}(V_g, \chi, \gamma, \phi, \bar{\omega}_\gamma) = \begin{bmatrix} -V_g(\sin(\chi)\cos(\gamma)\dot{\chi} + \cos(\chi)\sin(\gamma)\bar{\omega}_\gamma) \\ V_g(\cos(\chi)\cos(\gamma)\dot{\chi} - \sin(\chi)\sin(\gamma)\bar{\omega}_\gamma) \\ -V_g \cos(\gamma)\bar{\omega}_\gamma \end{bmatrix} \tag{5.60}$$

$$\dddot{\mathbf{p}}_{\text{lon}}(V_g, \chi, \gamma, \phi, \bar{\omega}_\gamma) = \begin{bmatrix} -V_g(\cos(\chi)\cos(\gamma)(\dot{\chi}^2 + \bar{\omega}_\gamma^2) - 2\sin(\chi)\sin(\gamma)\dot{\chi}\bar{\omega}_\gamma + \sin(\chi)\cos(\gamma)\ddot{\chi}) \\ V_g(-\sin(\chi)\cos(\gamma)(\dot{\chi}^2 + \bar{\omega}_\gamma^2) - 2\cos(\chi)\sin(\gamma)\dot{\chi}\bar{\omega}_\gamma + \cos(\chi)\cos(\gamma)\ddot{\chi}) \\ V_g(\sin(\gamma)\bar{\omega}_\gamma^2) \end{bmatrix} \tag{5.61}$$

where

$$\ddot{\chi}(V_g, \gamma, \phi, \bar{\omega}_\gamma) = \frac{g}{V_g} \sec^2(\phi)\dot{\phi}$$

As with the lateral transition segments, we approximate the longitudinal transition trajectory at discrete time steps. Let x_{lon} represent the full-state longitudinal transition trajectory defined as

$$x_{\text{lon}} = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_{m_{\text{lon}}} \\ \chi_1 & \chi_2 & \cdots & \chi_{m_{\text{lon}}} \\ \phi_1 & \phi_2 & \cdots & \phi_{m_{\text{lon}}} \\ \gamma_1 & \gamma_2 & \cdots & \gamma_{m_{\text{lon}}} \end{bmatrix}$$

where γ_k is the flight path angle at step k along the segment and

$$[\mathbf{p}_k^\top, \chi_k, \phi_k, \gamma_k]^\top = g_{\text{RK45}}([\mathbf{p}_{k-1}^\top, \chi_{k-1}, \phi_{k-1}, \gamma_{k-1}]^\top, V_g, \bar{\omega}_\gamma, T_s),$$

where g_{RK45} defines the RK45 propagation of the system defined in Eqs. (5.56) to (5.59) over a small time step T_s. We define the time t_{lon} needed to complete the transition trajectory as

$$t_{\text{lon}} = \frac{\gamma_{m_{\text{lon}}} - \gamma_1}{\bar{\omega}_\gamma},$$

where $\gamma_{m_{\text{lon}}}$ is the desired final flight path angle. The number of discrete propagation steps m_{lon} needed to define the desired transition trajectory is then

$$m_{\text{lon}} = \left\lceil \frac{t_{\text{lon}}}{T_s} \right\rceil. \tag{5.62}$$

A longitudinal transition trajectory can then be defined using Algorithm 6.

Algorithm 6 Longitudinal Transition Segment Trajectory

procedure LONGITUDINAL TRANSITION($\mathbf{p}_0, \chi_0, \phi_0, \gamma_0, V_g, \bar{\omega}_\gamma, T_s, \gamma_f$)
  $x_{\text{lon}}[:,0] \leftarrow [\mathbf{p}_0^\top, \chi_0, \phi_0, \gamma_0]^\top$
  Calculate $m_{\text{lon}} = \left\lceil \frac{(\gamma_f - \gamma_0)}{\bar{\omega}_\gamma T_s} \right\rceil$
  $i \leftarrow 1$
  while $i <= m_{\text{lon}}$ **do**
    $x_{\text{lon}}[:,i] \leftarrow g_{\text{RK45}}(x_{\text{lon}}[:,i-1], V_g, \bar{\omega}_\gamma, T_s)$
    $i \leftarrow i+1$
  end while
  return x_{lon}, m_{lon}
end procedure

The derivatives of the the trajectory can be computed using Eqs. (5.56), (5.60) and (5.61). With the longitudinal transition trajectory segment defined, we proceed to define a framework for smoothly transitioning between orbit altitudes.

5.5.3 Longitudinal Trajectory Smoothing

A smoothed longitudinal transition trajectory is constructed by inserting transition segments between the orbits and spiral descent segments of the simple spiral transition. Accordingly, with reference to Fig. 5.10, a smoothed altitude transition will consist of four segments: (Ia) a transition to the spiral, (IIa) a spiral, (IIIa) a transition from the spiral to the final orbit, and (IVa) a final orbit. We continue to parameterize each of these segments in order to explicitly define the

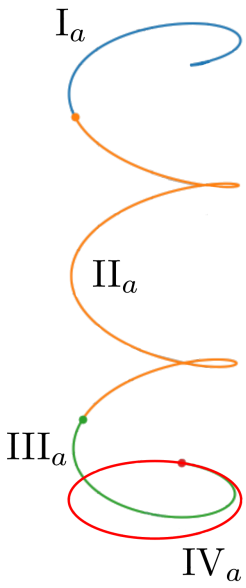


Figure 5.10: Inertial position of generic smoothed altitude transition trajectory. The trajectory was obtained using a relatively slow flight path angle rate with a steep spiral flight path angle to exaggerate path characteristics.

smoothed altitude transition trajectory.

We first define the initial longitudinal transition direction toward the desired spiral flight path angle γ_c from the starting flight path angle γ_0 as

$$\lambda_{\gamma 1} = \text{sign}(\gamma_c - \gamma_0).$$

We define the transition trajectory between the initial orbit altitude and the spiral path segment (i.e. segment I_a from Fig. 5.10) using Algorithm 6 as

$$[x_{\mathrm{lon,Ia}}, m_{\mathrm{lon,I}}] \leftarrow \text{Longitudinal Transition}(\mathbf{p}_0, \chi_0, \phi_0, \gamma_0, \lambda_{\gamma 1} \bar{\omega}_\gamma, T_s, \gamma_c),$$

where $\bar{\omega}_\gamma$ is a constant flight path angle rate and $\phi_0 = \phi_{\mathrm{orb}}(\lambda_1, V_g, r_1, 0)$ is the roll angle of the initial orbit calculated using Eq. (5.41). The endpoint states of segment I_a $[\mathbf{p}_{m_{\mathrm{lon,I}}}^\top, \chi_{m_{\mathrm{lon,I}}}, \phi_{m_{\mathrm{lon,I}}}, \gamma_{m_{\mathrm{lon,I}}}]^\top$ are reached at time $t_{1a} = t_{0a} + m_{\mathrm{lon,I}} T_s$ where t_{0a} is the start time of the altitude transition.

The spiral path segment (i.e. segment II_a from Fig. 5.10) can be parameterized by first determining the duration $\delta_{\mathrm{t,sp}}$ of the simple spiral trajectory segment to identify the initial states of segment III_a needed to level off at the desired final altitude. We calculate the difference in the third NED positional element produced by segment III_a by integrating the third positional component over the time needed to return to level flight from the transitioning flight path angle as

$$\delta_{\mathrm{D,f}} = \int_0^{t_{f,a}} -V_g \sin(\gamma(t))\, dt,$$

where $t_{f,a}$ is the time needed to transition between the flight path angle of segment II_a and level flight given by

$$t_{f,a} = \frac{-\gamma_c}{\bar{\omega}_\gamma}.$$

We then conclude

$$\delta_{\mathrm{D,f}} = \frac{V_g}{\bar{\omega}_\gamma} (\cos(\gamma_c) - 1).$$

Given a desired final inertial down position $p_{\mathrm{D,des}}$, we can then calculate the change in inertial down position $\delta_{\mathrm{D,sp}}$ required by segment II_a as

$$\delta_{\mathrm{D,sp}} = \mathbf{p}_{m_{\mathrm{lon,I}}}^\top \mathbf{e}_3 + \delta_{\mathrm{D,f}} - p_{\mathrm{D,des}} \tag{5.63}$$

The duration of the spiral trajectory can then be found by setting the third element of Eq. (5.52) equal to Eq. (5.63) as

$$\delta_{\mathrm{t,sp}} = -\frac{\delta_{\mathrm{D,sp}}}{V_g \sin(\gamma_c)}.$$

The endpoint states of segment II_a are then

$$\mathbf{p}_{\text{II}_a} = \mathbf{p}_{m_{\text{lon,I}}} + \begin{bmatrix} \frac{V_g^2}{g}\frac{\cos(\gamma_c)}{\tan(\phi_{m_{\text{lon,I}}})}\left(\sin(\frac{g}{V_g}\tan(\phi_{m_{\text{lon,I}}})\delta_{\text{t,sp}}+\chi_0)-\sin(\chi_0)\right) \\ -\frac{V_g^2}{g}\frac{\cos(\gamma_c)}{\tan(\phi_{m_{\text{lon,I}}})}\left(\cos(\frac{g}{V_g}\tan(\phi_{m_{\text{lon,I}}})\delta_{\text{t,sp}}+\chi_0)-\cos(\chi_0)\right) \\ -V_g\sin(\gamma_c)\delta_{\text{t,sp}} \end{bmatrix}$$

$$\chi_{\text{II}_a} = \chi_{m_{\text{lon,I}}} + \frac{g}{V_g}\tan(\phi_{m_{\text{lon,I}}})\delta_{\text{t,sp}}$$

$$\phi_{\text{II}_a} = \phi_{m_{\text{lon,I}}}$$

$$\gamma_{\text{II}_a} = \gamma_{m_{\text{lon,I}}},$$

which are reached at time $t_{2a} = t_{1a} + \delta_{\text{t,sp}}$.

The transition from the spiral path to the final orbit (i.e. segment III_a from Fig. 5.10) is parameterized by first defining the final transitioning flight path angle direction to be

$$\lambda_{\gamma_2} = \text{sign}(-\gamma_c).$$

Similar to segment I_a, we then define segment III_a to be

$$[x_{\text{lon,IIIa}}, m_{\text{lon,IIIa}}] \leftarrow \text{Longitudinal Transition}(\mathbf{p}_{\text{II}_a}, \chi_{\text{II}_a}, \phi_{\text{II}_a}, \gamma_{\text{II}_a}, \lambda_{\gamma_2}\bar{\omega}_\gamma, T_s, 0)$$

with endpoint states $[\mathbf{p}^\top_{m_{\text{lon,IIIa}}}, \chi_{m_{\text{lon,IIIa}}}, \phi_{m_{\text{lon,IIIa}}}, \gamma_{m_{\text{lon,IIIa}}}]^\top$ reached at time $t_{3a} = t_{2a} + m_{\text{lon},2}T_s$. The final orbit (i.e. segment IV_a from Fig. 5.10) is parameterized using Eq. (5.18) with these endpoint states.

We then define the smooth altitude transition as a discrete inertial position trajectory with course and roll angle endpoint states using Algorithm 7.

The first three derivatives of the trajectory points from Algorithm 7 can be obtained from Eqs. (5.53) to (5.56), (5.60) and (5.61) for use with the differential-flatness map. The endpoint states given by Algorithm 7 can be used to initialize the final orbit trajectory using Eqs. (5.18) and (5.22). Intermediate trajectory point states and derivatives can again be approximated by linear interpolation.

Algorithm 7 Discrete Smooth Altitude Transition Trajectory

procedure SMOOTH ALTITUDE TRANSITION($\mathbf{p}_0, \chi_0, \phi_0, \gamma_0, V_g, \gamma_c, \bar{\omega}_\gamma, T_s, t_{0a}, p_{\mathrm{D,des}}$)

▷ Segment I

$\lambda_{\gamma 1} \leftarrow \mathrm{sign}(\gamma_c - \gamma_0)$

$\left[\left[\mathbf{p}_{\mathrm{lon,I}}^\top, \chi_{\mathrm{lon,I}}, \phi_{\mathrm{lon,I}}, \gamma_{\mathrm{lon,I}}\right]^\top, m_{\mathrm{lon,I}}\right] \leftarrow$ Longitudinal Transition$(\mathbf{p}_0, \chi_0, \phi_0, \gamma_0, \lambda_{\gamma 1}\bar{\omega}_\gamma, T_s, \gamma_c)$

▷ Segment II

$t_{1a} \leftarrow t_{0a} + m_{\mathrm{lon,I}} T_s$

$\delta_{\mathrm{D,f}} \leftarrow \frac{V_g}{\bar{\omega}_\gamma}(\cos(\gamma_c) - 1)$

$\delta_{\mathrm{D,sp}} \leftarrow \mathbf{p}_{m_{\mathrm{lon,I}}}^\top \mathbf{e}_3 + \delta_{\mathrm{D,f}} - p_{\mathrm{D,des}}$

$\delta_{\mathrm{t,sp}} \leftarrow -\frac{\delta_{\mathrm{D,sp}}}{V_g \sin(\gamma_c)}$

$t_{2a} \leftarrow t_{1a} + \delta_{\mathrm{t,sp}}$

$\mathbf{p}_{\mathrm{lon,II}} \leftarrow$ Sample $\left(\mathbf{p}_{\mathrm{sp}}(t, t_{1a}, \mathbf{p}_{\mathrm{lon,I}}[:, m_{\mathrm{lon,I}}], \phi_{\mathrm{lon,I}}[:, m_{\mathrm{lon,I}}], \chi_{\mathrm{lon,I}}[:, m_{\mathrm{lon,I}}], \gamma_c, V_g), t_{1a}, t_{2a}, T_s\right)$

$\mathbf{p}_{\mathrm{II}_a} \leftarrow \mathbf{p}_{\mathrm{sp}}(t_{2a}, t_{1a}, \mathbf{p}_{\mathrm{lon,I}}[:, m_{\mathrm{lon,I}}], \phi_{\mathrm{lon,I}}[:, m_{\mathrm{lon,I}}], \chi_{\mathrm{lon,I}}[:, m_{\mathrm{lon,I}}], \gamma_c, V_g)$

$\chi_{\mathrm{II}_a} \leftarrow \chi_{\mathrm{lon,I}}[:, m_{\mathrm{lon,I}}] + \frac{g}{V_g}\tan(\phi_{\mathrm{lon,I}}[:, m_{\mathrm{lon,I}}])\delta_{\mathrm{t,sp}}$

$\phi_{\mathrm{II}_a} \leftarrow \phi_{\mathrm{I}_a}$

▷ Segment III

$\lambda_{\gamma 2} \leftarrow \mathrm{sign}(-\gamma_c)$

$\left[\left[\mathbf{p}_{\mathrm{lon,III}}^\top, \chi_{\mathrm{lon,III}}, \phi_{\mathrm{lon,III}}, \gamma_{\mathrm{lon,III}}\right]^\top, m_{\mathrm{lon,III}}\right]$
$\leftarrow$ Longitudinal Transition$(\mathbf{p}_{\mathrm{II}_a}, \chi_{\mathrm{II}_a}, \phi_{\mathrm{II}_a}, \gamma_c, \lambda_{\gamma 2}\bar{\omega}_\gamma, T_s, 0)$

▷ Combine Segments

$\mathbf{p}_{\mathrm{sat}} \leftarrow \left[\mathbf{p}_{\mathrm{lon,I}} \mid \mathbf{p}_{\mathrm{lon,II}} \mid \mathbf{p}_{\mathrm{lon,III}}\right]$

return $\mathbf{p}_{\mathrm{sat}}$, $\chi_{\mathrm{lon,III}}[:, m_{\mathrm{lon,III}}]$, $\phi_{\mathrm{lon,III}}[:, m_{\mathrm{lon,III}}]$

end procedure

The parameterization for smoothed altitude transition trajectories was obtained for an arbitrarily selected spiral flight path angle. We optimally select γ_c such that the altitude transition rate is maximized while maintaining both dynamic and visual feasibility constraints. Dynamical path feasibility is enforced by selecting a feasible flight path angle rate and enforcing that the flight path angle magnitude be less than the maximum allowable flight path angle magnitude $\gamma_{\mathrm{c,max}}$. We en-

force visual feasibility by again checking discretely-sampled trajectory points using the differential flatness map.

The full process for defining a smooth altitude transition trajectory given initial UAS states is given by Algorithm 8.

Algorithm 8 Smooth Altitude Planner

procedure SMOOTH ALTITUDE PLANNER($\mathbf{p}_0, \chi_0, \phi_0, \gamma_0, V_g, p_{\text{D,des}}$)
  Define the minimum flight path angle rate magnitude $\bar{\omega}_\gamma$
  Define the maximum flight path angle $\gamma_{\max}$
  Define initial flight path angle search step γ_{step}
  Define gimbal saturation limits $\theta_{\text{el,min}}, \theta_{\text{el,max}}, \theta_{\text{az,min}}, \theta_{\text{az,max}}$
  Select $T_s << 1$ (e.g. 0.01)
  $t_{0a} \leftarrow 0$
  $\gamma_c \leftarrow \gamma_{\max}$
  $\Theta_{\text{el}} \leftarrow \{\}$
  $\Theta_{\text{az}} \leftarrow \{\}$
  while $\theta_{\text{el,min}} \leq \Theta_{\text{el}} \leq \theta_{\text{el,max}}$ and $\theta_{\text{az,min}} \leq \Theta_{\text{az}} \leq \theta_{\text{az,max}}$ **do**
    $\mathbf{p}_{\text{traj}}, \chi_f, \phi_f \leftarrow$ Smooth Altitude Transition$(\mathbf{p}_0, \chi_0, \phi_0, \gamma_0, V_g, \gamma_c, \bar{\omega}_\gamma, T_s, t_{0a}, p_{\text{D,des}})$
    Calculate $\dot{\mathbf{p}}_{\text{traj}}$, $\ddot{\mathbf{p}}_{\text{traj}}$, and $\mathbf{p}^{(3)}_{\text{traj}}$ using Eqs. (5.53) to (5.56), (5.60) and (5.61)
    Calculate sampled gimbal inputs $\Theta_{\text{el}}, \Theta_{\text{az}}$ using flatness proof in [39]
    $\gamma_c \leftarrow \gamma_c - \gamma_{\text{step}}$
  end while
  return $\mathbf{p}_{\text{traj}}$, $\dot{\mathbf{p}}_{\text{traj}}$, $\ddot{\mathbf{p}}_{\text{traj}}$, $\dddot{\mathbf{p}}_{\text{traj}}$, χ_f, ϕ_f
end procedure

As with Algorithm 5, Algorithm 8 employs a simple line search to identify a large feasible flight path angle for the altitude transition; however, the maximum feasible flight path angle could be found using other optimization techniques such as a bisection search [42].

Table 5.2: Fixed-wing system parameters

Param	Value	Param	Value	Param	Value
m	1800.0	J_x	700.0	J_y	5000.0
J_z	5500.0	J_{xz}	70.0	S	9.0
b	7.0	c	1.5	C_{L_0}	0.15
C_{L_α}	4.5	C_{L_q}	9.0	$C_{L_{\delta_e}}$	0.3
C_{D_0}	0.02	C_{D_α}	-0.3	C_{D_q}	0.0
C_{l_0}	0.0	C_{l_β}	-0.16	C_{l_p}	-0.4
C_{l_r}	0.1	$C_{l_{\delta_a}}$	0.06	$C_{l_{\delta_r}}$	0.04
C_{m_0}	-0.02	C_{m_α}	-2.4	C_{m_q}	-11.0
$C_{m_{\delta_e}}$	-0.5	C_{n_0}	0.0	C_{n_β}	0.1
C_{n_p}	0.02	C_{n_r}	-0.3	$C_{n_{\delta_a}}$	0.0
$C_{n_{\delta_r}}$	-0.08	C_{Y_0}	0.0	C_{Y_β}	-0.5
C_{Y_p}	-0.1	C_{Y_r}	0.6	$C_{Y_{\delta_r}}$	0.16

5.6 Simulation Results

To demonstrate the ability of the proposed smoothed bi-tangent framework from Section 5.4 and the altitude transition framework from Section 5.5 in maintaining dynamical and visual feasibility of a detected target while transitioning to an overhead orbit, a UAS with an inertially-mounted, two-axis gimbal camera system observing a ground target was simulated using the ROS Gazebo simulation environment. The UAS dynamics were implemented following the fixed-wing model in [31] with parameters defined in Table 5.2.

Model simplifications and system disturbances were accounted for by pairing the differential-flatness control inputs with an linear quadratic regulator (LQR) feedback controller [39] as shown in Fig. 5.11.

The UAS was given a starting NED inertial position of $[-400, 100, -3000]$ meters initially facing north with an airspeed velocity of 120 meters per second and initial gimbal azimuth and elevation angles of -90 degrees and 88 degrees respectively. The ground target was given a static inertial position at the origin. The UAS executed a level flight trajectory until the target of interest was detected. A smoothed bi-tangent trajectory was then calculated and tracked to transition to an orbit of 2500 meter radius over the target of interest. The UAS then executed a smooth altitude transition trajectory to an orbit altitude of 2000 meters. Commanded roll rates and flight path angle rates were selected as 8 degrees per second and 0.1 degrees per second respectively. Gimbal was

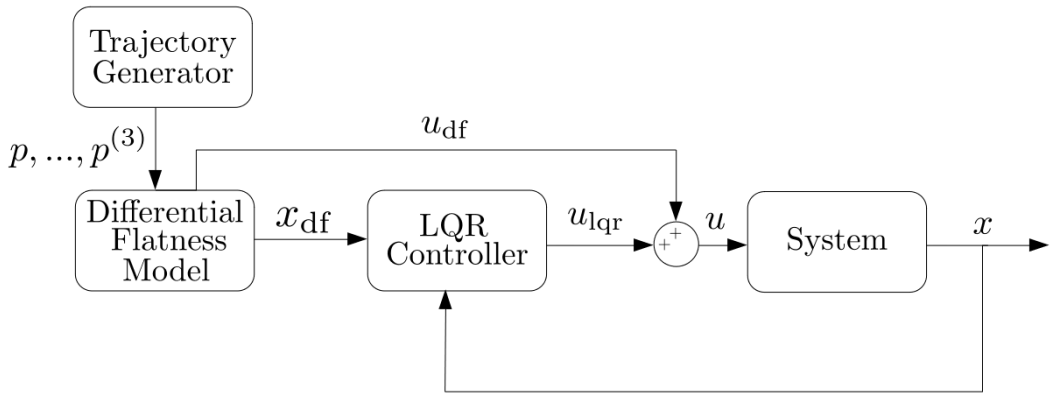


Figure 5.11: Control architecture defined in [39] uses a given inertial position trajectory p up to its third derivative to define the predicted control $u_{(\mathrm{df})}$ and desired aircraft states $x_{(\mathrm{df})}$ using the differential-flatness mapping. The error between the desired states and actual states x are passed to an LQR controller whose control output u_{lqr} augments the feed-forward inputs produced by the differential-flatness map. The positional trajectory and its derivatives are provided by the smooth bi-tangent and altitude transition frameworks.

given elevation saturation limits at 0 and 90 degrees along with azimuth limits of ± 150 degrees. The horizontal field-of-view angle of the camera was 8.08 degrees with an aspect ratio of 4:3. The resulting trajectory projected onto the north-east plane is shown in Fig. 5.12 with the altitude trajectory shown in Fig. 5.13.

The dynamic feasibility of the calculated path is demonstrated by the minimal tracking error of the UAS and path shown by Fig. 5.14. Visual feasibility is shown by the boundedness of the gimbal angles within their respective saturation limits shown in Fig. 5.15 and by reasonably bounded alignment error between the target and the optical axis produced by the gimbal inputs obtained from the differential-flatness mapping illustrated by Fig. 5.16.

As can be seen in Figs. 5.14 to 5.16, the smoothed bi-tangent and altitude transition models are successfully able to produce both dynamically and visually feasible trajectories for converging to an orbit over a known target and transitioning to a second orbit altitude. Dynamical feasibility of the path was demonstrated by the minimal tracking error of the UAS and visual feasibility between the UAS and target was demonstrated by gimbal angles never passing saturation bounds. Throughout the trajectory, target alignment with the optical axis never surpassed 2.5 degrees. For the defined camera parameters, the smallest possible alignment error resulting in loss of target visibility was 3 degrees, therefore the target was continuously held in the camera field-of-view.

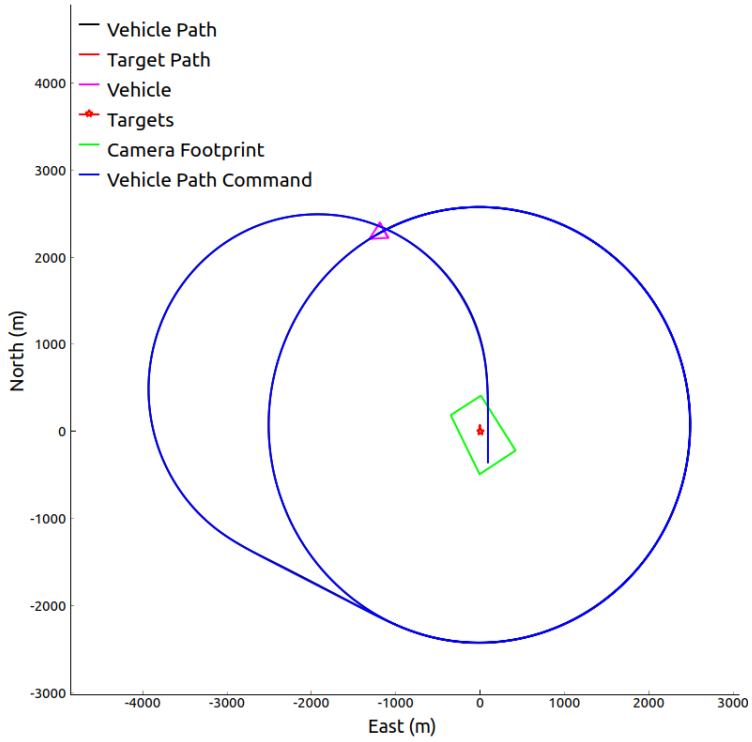


Figure 5.12: UAS and target positional states over time projected onto the inertial north-east plane. Projection of the camera field-of-view on the ground plane containing the target is shown. The UAS is show to accurately transition to an overhead orbit of the intended target.

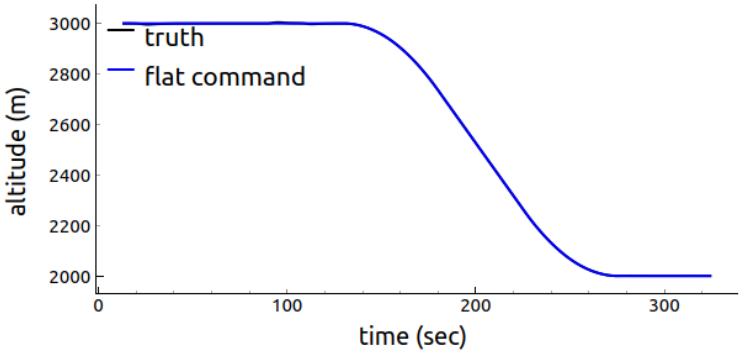


Figure 5.13: UAS altitude trajectory. The smoothed altitude transition is executed at the end of the smooth bi-tangent transition at approximately 130 seconds with a selected flight path angle rate of 0.1 degrees per second.

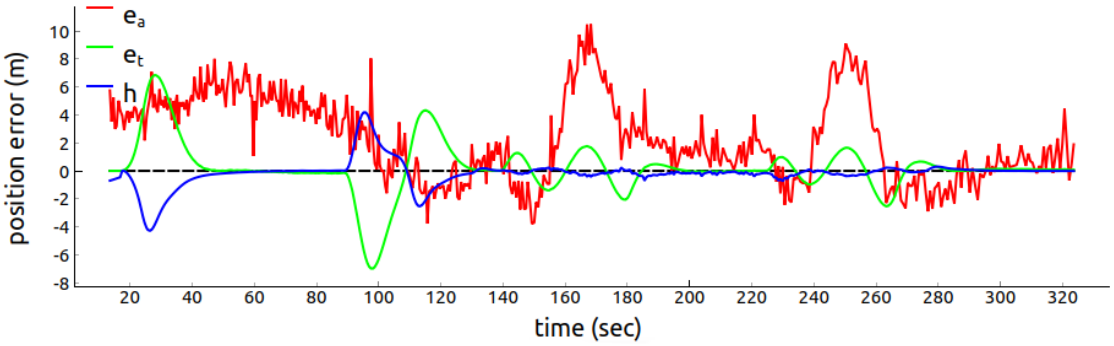


Figure 5.14: Error between actual UAS positional states and the desired positional outputs of the proposed path planner. Dynamical feasibility of the path is demonstrated by the minimal tracking error of the UAS. Positional errors are transformed to axial (e_a), transverse (e_t), and altitude (h) error for interpretation convenience.

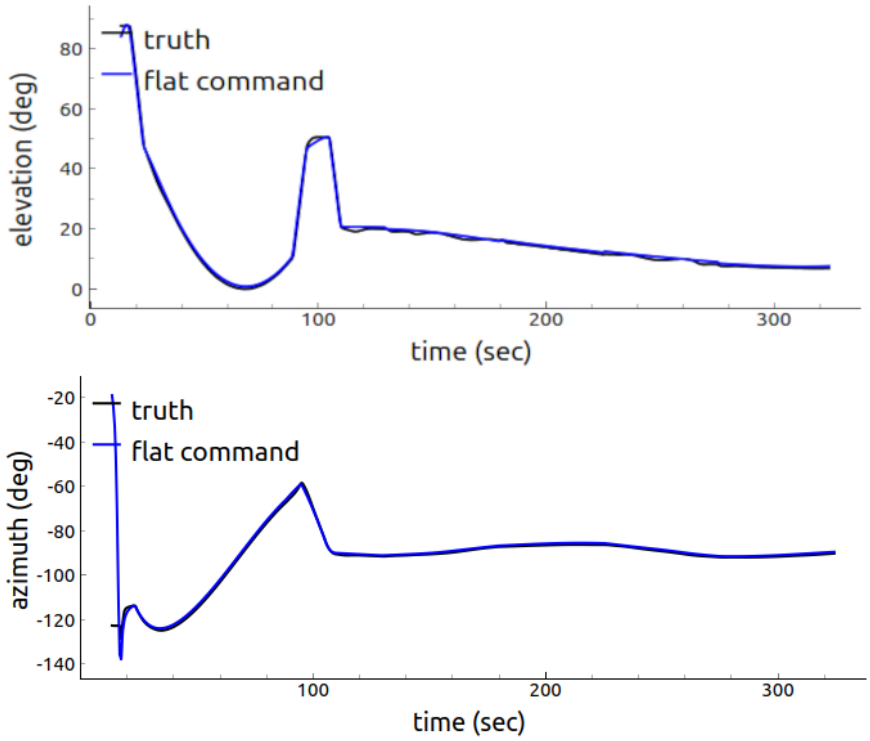


Figure 5.15: Gimbal azimuth and elevation angle trajectories resulting from following the proposed path planner. Visual feasibility between the UAS and target is demonstrated by elevation and azimuth angles never passing their saturation bounds at (0,90) and (-150,150) degrees respectively.

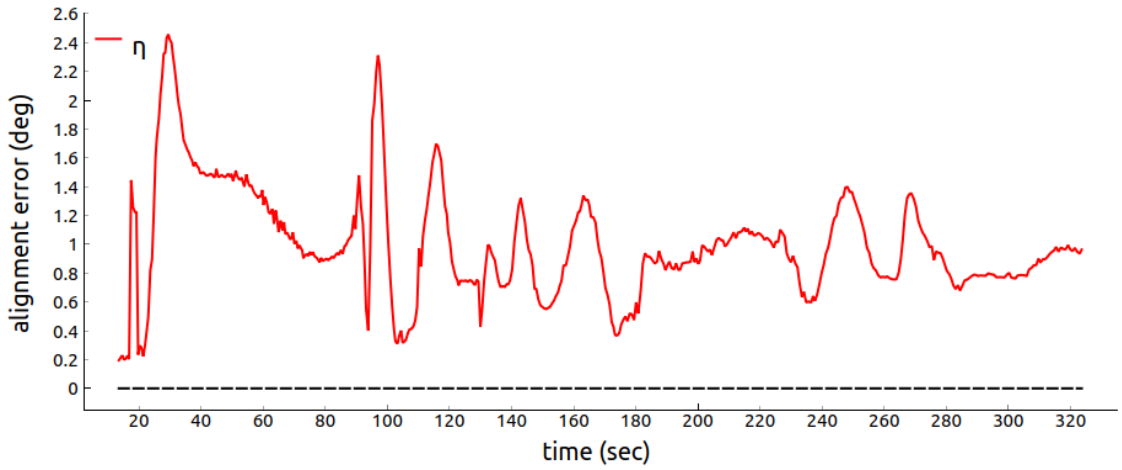


Figure 5.16: Angle alignment error between the optical axis and the vector defining the line-of-sight between the camera and the target of interest. Alignment error never surpasses 2.5 degrees. The target is therefore maintained within the field-of-view of the camera throughout the entire trajectory.

5.7 Conclusion

In this paper, we presented a path planning framework for a generalized fixed-wing aircraft with a two-axis gimbal camera transitioning to an orbit over a known target and descending to a new altitude without surpassing gimbal limits. The proposed framework was able to successfully converge to an orbit over a ground target in simulation while maintaining continuous target visibility. Future work includes defining sufficient conditions for the existence of smooth bi-tangent trajectories without azimuth angle saturation. Feasibility conditions could also be extended to include the full field-of-view of the camera instead of strict alignment with the optical axis. Additional work could also be done in adjusting the planning framework for wind conditions and high-velocity target motion.

CHAPTER 6. CONCLUSIONS AND FUTURE WORK

In this thesis we explored the challenges of tracking very small, slow-moving targets from a gimballed UAS camera. We provided guidelines for tuning the parameters of the Visual MTT algorithm to enhance its performance in detecting and tracking these types of targets and demonstrated its effectiveness by successfully tracking walking human targets on 2048×1500 resolution test data obtained from an aircraft equipped with a camera flying at 1400 meters.

To account for loss of target visibility the rotational constraints of a gimballed UAS were considered and an algorithmic approach was derived for determining the rotational limits of the system along its available rotational degrees of freedom. A method for calculating the most critical rotational bound resulting in the fastest loss or acquisition of target visibility for a given aircraft and target configuration was also derived. The accuracy of these calculated bounds was validated in simulation.

To ensure a known target can continuously be observed from a gimballed fixed-wing UAS under both dynamic and visual constraints, a differential flatness model was derived for gimballed fixed-wing UAS which can be used both as a feedforward component in trajectory tracking as well as a predictive input map for a given trajectory. This algebraic input map was used to develop a path planning framework for producing both dynamically and visually feasible trajectories for transitioning a gimballed fixed-wing aircraft to an orbit over a known target along with feasible trajectories for transitioning to orbits or varying altitudes. The proposed framework was able to maintain target visibility throughout both the smoothed bi-tangent and altitude orbit transition trajectories while maintaining continuous target visibility with a maximum optical axis alignment error of approximately 2.5 degrees.

6.1 Summary of Contributions

The contributions of this thesis are listed below.

- This thesis introduced guidelines and tuning parameters for tracking very small, slow-moving targets in high-resolution images with the Visual MTT algorithm. The effectiveness of these guidelines was demonstrated in tracking walking human targets described $10-15$ pixels from a high altitude on real flight test data.

- This work derived an algorithm for determining the rotational limits of gimballed UAS tracking a known target. This result provides insight into the sensitivity of target visibility due to rotations about the controllable axes of the UAS.

- An algorithm was defined for determining the most critical UAS or gimbal rotation resulting in the fastest loss or acquisition of target visibility. This algorithm could easily be extended to work with controllers for assessing visual sensitivity for a given aircraft-target configuration.

- A novel differential flatness model and control framework was introduced for gimballed fixed-wing aircraft allowing trajectory tracking under vision constraints. This control framework demonstrated successful target tracking on simple trajectories and motivated the development of a path planning framework with guarantees on dynamic feasibility and target visibility.

- A path planning framework was presented for feasibly transitioning to an overhead orbit over a target while maintaining constant target visibility. The framework includes lateral transitions to overhead orbits as well as transitions between orbits of different altitudes. The effectiveness of this framework is demonstrated on a UAS with slow dynamics in simulation by continuously maintaining target visibility throughout both proposed transition trajectories.

6.2 Future Work

The various algorithms and theories presented in this thesis can be improved in many ways. Generically, the differential flatness control and path planning frameworks function under the assumption that there are no significant wind disturbances in the environment. It is also assumed generally that the angle of attack of the aircraft is held constant, which is especially not the case for more aggressive or acrobatic maneuvers. Extending these models to appropriately account

for the effects of relative airspeed on the aircraft would greatly add to their utility in real-world applications.

6.2.1 Small Target Tracking

The Visual MTT algorithm is currently being refactored to function with a new Recursive RANSAC back end that is able to generate models described by lie groups. As a result, the tuning parameters of Visual MTT are in the process of being adjusted as well. Additional work could therefore be done to extend the proposed parameter tuning guidelines for the new parameters.

6.2.2 Visual Rotational Constraints

One significant limitation of the VRC and VRCC algorithms discussed in Chapter 3 is that they exclusively account for rotational limitations of the system but do not in any way define limits on translational motion of the camera. Especially in circumstances where targets are observed in close proximity, translational motion of the camera can have significant impacts on the camera's ability to observe a desired target.

Another significant limitation of these algorithms is that the calculated bounds only account for motion along a single axis at a time. As shown in the simulation results of Chapter 3, rotation along on axis transforms the rotational bounds of other axes. Extending the algorithm to detect boundary conditions for hybrid rotations about multiple rotational axes simultaneously would allow for more realistic boundary analysis, especially in UAS applications where hybrid rotations are commonly encountered.

This work could potentially be utilized in a visual servoing framework by using distances from calculated critical rotational bounds as control variables to drive a known target to the center of a camera's field-of-view using its controllable degrees of freedom. The proposed algorithms could also be extended to account for occlusion points created by features of the UAS such as wings, landing gear, payloads, etc.

6.2.3 Gimballed Fixed-Wing Flatness Control

The differential flatness control model could be improved by continuously re-linearizing the LQR controller gains in the control framework about the given trajectory. For simplicity, the current approach uses a state space model obtained by linearizing the dynamic model once about level flight conditions. This re-linearization could greatly improve the tracking capabilities of the aircraft relative to the desired trajectory resulting in even more accurate input predictions for visual feasibility analysis.

Another significant improvement would be to account for the full camera field-of-view in assessing visual feasibility. Currently the proposed model constrains the optical axis of the camera to be exactly aligned with the target. If gimbal angles cannot produce this alignment, the configuration is considered not visually feasible. Intuitively, however, the field-of-view of the camera occupies an expanded region around the optical axis. Thus, some misalignment may be allowable for maintaining target visibility. Additional work could also be done to analyze the confidence of the model under model and sensor uncertainties.

6.2.4 Gimballed Fixed-Wing Flatness Path Planning

Perhaps the greatest area of improvement for the path planning framework is defining sufficient conditions for the existence of smooth bi-tangent trajectories without azimuth angle saturation. Sufficient conditions were provided for the existence of a smooth bi-tangent trajectory without elevation saturation, however, without a sufficiency condition for azimuth the framework cannot guarantee the existence of a fully visually feasible trajectory solution. This limits confidence in using the algorithm online where rapid calculation of a transition trajectory is critical.

Another improvement to the path planner could be to account for transitioning to an orbit trajectory over a rapidly moving target. The current framework assumes that the target is moving slow enough to be considered static resulting in a transition trajectory to a final static orbit. For fast-moving targets, the framework could be extended to transition to a translating orbit trajectory over the target instead.

REFERENCES

[1] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, X. Zhao, and T.-K. Kim, "Multiple object tracking: A literature review," *Artificial Intelligence*, vol. 293, 05 2017. 1, 2

[2] Y. Xu, X. Zhou, S. Chen, and F. Li, "Deep learning for multiple object tracking: a survey," *IET Computer Vision*, vol. 13, no. 4, pp. 355–368, 2019. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-cvi.2018.5598 1, 2

[3] P. C. Niedfeldt and R. W. Beard, "Multiple target tracking using recursive RANSAC," in *2014 American Control Conference*, June 2014, pp. 3393–3398. 1, 5

[4] P. C. Niedfeldt, K. Ingersoll, and R. W. Beard, "Comparison and analysis of recursive-RANSAC for multiple target tracking," *IEEE Transaction on Aerospace and Electronic Systems*, vol. 53, no. 1, pp. 461–476, feb 2017. 1, 5

[5] P. C. Lusk and R. W. Beard, "Visual multiple target tracking from a descending aerial platform," in *2018 Annual American Control Conference (ACC)*, 2018, pp. 5088–5093. 1

[6] W. Li, J. Mu, and G. Liu, "Multiple object tracking with motion and appearance cues," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 161–169. 2

[7] A. A. Butt and R. T. Collins, "Multiple target tracking using frame triplets," in *Computer Vision – ACCV 2012*, K. M. Lee, Y. Matsushita, J. M. Rehg, and Z. Hu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 163–176. 2

[8] D. Shi, S. Zhang, J. Wang, and Y. Gong, "Detection and association based multi-target tracking in surveillance video," in *2015 IEEE International Conference on Multimedia Big Data*, 2015, pp. 377–382. 2

[9] J. Saunders and R. W. Beard, "Visual tracking in wind with field of view constraints," *International Journal of Micro Air Vehicles*, vol. 3, no. 3, 2011. 2, 11, 12, 39, 54

[10] J. Egbert and R. W. Beard, "Low-altitude road following using strap-down cameras on miniature air vehicles," *Mechatronics*, vol. 21, no. 5, pp. 831 – 843, 2011, special Issue on Development of Autonomous Unmanned Aerial Vehicles. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957415810001881 2, 11, 12

[11] R. Sharma and D. Pack, "Cooperative sensor resource management to aid multi target geolocalization using a team of small fixed-wing unmanned aerial vehicles," in *AIAA Guidance, Navigation, and Control (GNC) Conference*, 2013. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2013-4706 2, 12

[12] L. Burlion and H. de Plinval, "Toward vision based landing of a fixed-wing UAV on an unknown runway under some FOV constraints," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2017, pp. 1824–1832. 2, 39, 54

[13] S. A. Quintero and J. P. Hespanha, "Vision-based target tracking with a small UAV: Optimization-based control strategies," *Control Engineering Practice*, vol. 32, pp. 28 – 42, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0967066114001774 3, 39, 54

[14] M. Gros and W. Fichter, "G3-continuous trajectory design for fixed-wing aircraft based on 6-DoF kinematics," in *AIAA Guidance, Navigation, and Control Conference*, 01 2016. 3, 40, 55

[15] A. Bry, C. Richter, A. Bachrach, and N. Roy, "Aggressive flight of fixed-wing and quadrotor aircraft in dense indoor environments," *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 969–1002, 2015. [Online]. Available: https://doi.org/10.1177/0278364914558129 3, 40, 55

[16] H. Alturbeh and J. Whidborne, "Real-time obstacle collision avoidance for fixed wing aircraft using B-splines," in *2014 UKACC International Conference on Control, CONTROL 2014 - Proceedings*, 07 2014. 3, 40, 55

[17] O. Ogunbodede, S. Nandi, and T. Singh, "Periodic Control of Unmanned Aerial Vehicles Based on Differential Flatness," *Journal of Dynamic Systems, Measurement, and Control*, vol. 141, no. 7, 03 2019, 071003. [Online]. Available: https://doi.org/10.1115/1.4043114 3, 40, 55

[18] P. Martin, "Aircraft control using flatness," in *IMACS/IEEE-SMC Multiconference CESA'96 - Symposium on Control, Optimization and Supervision*, Lille, France, 1996, pp. 194–199. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00935110 3, 40, 55

[19] N. Faiz, S. K. Agrawal, and R. M. Murray, "Trajectory planning of differentially flat systems with dynamics and inequalities," *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 2, pp. 219–227, 2001. [Online]. Available: https://doi.org/10.2514/2.4732 3, 40, 55

[20] Chaojie Zhang, Nan Wang, and Jing Chen, "Trajectory generation for aircraft based on differential flatness and spline theory," in *2010 International Conference on Information, Networking and Automation (ICINA)*, vol. 1, 2010, pp. 110–114. 3, 40, 55

[21] R. Rysdyk, "Unmanned aerial vehicle path following for target observation in wind," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 5, pp. 1092–1100, 2006. [Online]. Available: https://doi.org/10.2514/1.19101 11

[22] V. Stepanyan and N. Hovakimyan, "Visual tracking of a maneuvering target," *Journal of Guidance Control and Dynamics*, vol. 31, pp. 66–80, 01 2008. 11

[23] P. Theodorakopoulos and S. Lacroix, "A strategy for tracking a ground target with a UAV," *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1254–1259, 2008. 11

[24] N. R. Gans, J. Shen, and J. W. Curtis, "Selection of a UAV orbit to keep multiple targets in the camera field of view," in *2010 IEEE International Symposium on Intelligent Control*, Sep. 2010, pp. 807–812. 11

[25] L. Burlion and H. de Plinval, "Keeping a ground point in the camera field of view of a landing UAV," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 5763–5768. 11

[26] H. Oh and S. Kim, "Persistent standoff tracking guidance using constrained particle filter for multiple UAVs," *Aerospace Science and Technology*, vol. 84, pp. 257 – 264, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1270963817323246 11

[27] Z. Li, N. Hovakimyan, V. Dobrokhodov, and I. Kaminer, "Vision-based target tracking and motion estimation using a small UAV," in *49th IEEE Conference on Decision and Control (CDC)*, 2010, pp. 2505–2510. 11

[28] S. Heshmati-alamdari, C. P. Bechlioulis, M. V. Liarokapis, and K. J. Kyriakopoulos, "Prescribed performance image based visual servoing under field of view constraints," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2014, pp. 2721–2726. 11

[29] G. López-Nicolás, N. R. Gans, S. Bhattacharya, C. Sagüés, J. J. Guerrero, and S. Hutchinson, "Homography-based control scheme for mobile robots with nonholonomic and field-of-view constraints," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 40, no. 4, pp. 1115–1127, Aug 2010. 11

[30] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. 18

[31] R. Beard and T. McLain, *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press, 2012. 36, 41, 42, 45, 48, 57, 64, 71, 91

[32] Y. Yu, X. Wang, and L. Shen, "Optimal uav circumnavigation control with input saturation based on information geometry," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 2471–2476, 2020, 21th IFAC World Congress. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896320304614 40, 55

[33] Z. Yu, Z. Liu, Y. Zhang, Y. Qu, and C.-Y. Su, "Distributed finite-time fault-tolerant containment control for multiple unmanned aerial vehicles," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 6, pp. 2077–2091, 2020. 40, 55

[34] Z. Yu, Y. Zhang, B. Jiang, C.-Y. Su, J. Fu, Y. Jin, and T. Chai, "Nussbaum-based finite-time fractional-order backstepping fault-tolerant flight control of fixed-wing uav against input saturation with hardware-in-the-loop validation," *Mechanical Systems and Signal Processing*, vol. 153, p. 107406, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327020307925 40, 55

[35] J. Patrikar, V. R. Makkapati, A. Pattanaik, H. Parwana, and M. Kothari, "Nested Saturation Based Guidance Law for Unmanned Aerial Vehicles1," *Journal of Dynamic Systems,*

Measurement, and Control, vol. 141, no. 7, 04 2019, 071008. [Online]. Available: https://doi.org/10.1115/1.4043107 40, 55

[36] R. W. Beard, J. Ferrin, and J. Humpherys, "Fixed wing uav path following in wind with input constraints," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 6, pp. 2103–2117, 2014. 40, 55

[37] I. Cowling, O. Yakimenko, J. Whidborne, and A. Cooke, "A prototype of an autonomous controller for a quadrotor UAV," in *2007 European Control Conference, ECC 2007*, 07 2007, pp. 4001–4008. 42, 57

[38] R. Beard and T. McLain, *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press, 2012, ch. 13. 46, 57

[39] H. Morgan and R. Beard, "Flatness-based control of a gimballed fixed-wing UAS," in *2021 International Conference on Unmanned Aircraft Systems, ICUAS 2021*, 06 2021. 58, 79, 90, 91, 92

[40] E. Fehlberg, *Low-order classical Runge-Kutta formulas with stepsize control and their application to some heat transfer problems*. National aeronautics and space administration, 1969, vol. 315. 70

[41] D. Kraft, *A software package for sequential quadratic programming*, ser. Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt Köln: Forschungsbericht. Wiss. Berichtswesen d. DFVLR, 1988. [Online]. Available: https://books.google.com/books?id=4rKaGwAACAAJ 76

[42] R. Burden and D. Faires, *Numerical Analysis*, 3rd ed. Prindle, Weber & Schmidt, 1985, ch. 2. 78, 90

APPENDIX A. PRE-LOADED LATERAL TRANSITION TRAJECTORY

For optimization routines calculating discrete lateral transition trajectories, reconstruction will likely be performed a significant number of times. Numerically integrating the curve at small time intervals over a large trajectory can make this process computationally infeasible for many applications. By using a pre-defined constant roll rate input, these transition paths can be computed offline and then appropriately segmented and transformed to represent initial and endpoint conditions of the curves.

We begin by defining the discrete RK45 approximation of the vehicle states assuming initial conditions $\mathbf{p}_0 = 0$, $\chi_0 = 0$, and $\phi_0 = 0$ up to a maximum desired roll condition $\phi_{\max}$ using Algorithm 3 as

$$[x_{\text{tr}+}, m_z] \leftarrow \text{Lateral Transition}(p_0, \chi_0, \phi_0, \dot{\phi}_{\text{des}}, T_s, \phi_{\max}).$$

This segment accounts for all possible states of the lateral transition segment with a positive roll angle. States associated with a negative roll angle are obtained by propagating backward in time, or more simply by reflecting the discrete states across the origin and negating the position and roll angle states using Algorithm 9

Algorithm 9 Pre-load Lateral Transition Trajectory

procedure PRE-LOAD LATERAL TRANSITION($\bar{\omega}_\phi, T_s, \phi_{\max}$)
  Calculate $m_z = \left\lceil \frac{(\phi_{\max} - \phi_0)}{\bar{\omega}_\phi T_s} \right\rceil$ from Eq. (5.42)
  Set $x_{\text{pre}}[:, m_z] = [0,0,0,0,0]^\top$
  $i \leftarrow 1$
  while $i <= m_z$ **do**
    $x_{\text{pre}}[:, m_z + i] = f_{\text{RK45}}(x_{\text{pre}}[m_z + i - 1])$
    $x_{\text{pre}}[:, m_z - i] = I_{\text{ref}} x_{\text{pre}}[m_z + i]$
    $i \leftarrow i + 1$
  end while
  return x_{pre}, m_z
end procedure

where x_{pre} is the preloaded transition state matrix, m_z is the index to the zero state at the center of the state matrix, and I_{ref} is a reflection matrix defined as

$$I_{\text{ref}} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix}.$$

With the full transition trajectory state vector defined, we can then segment and appropriately transform it to arbitrary transition trajectory endpoint constraints.

We first note that Algorithm 9 pre-loads a transition trajectory under the assumption that the roll rate command is a positive constant value. However, smoothed bi-tangent trajectories will require both positive and negative roll rate commands to smooth the various transition segments.

For notational simplicity, we define a roll direction flag λ_ϕ to be

$$\lambda_\phi = \begin{cases} 1, & \text{if } \phi_f \geq \phi_0 \\ -1, & \text{otherwise} \end{cases} \tag{A.1}$$

Given a preloaded transition, it can be shown that the transition curve for a negative roll rate command is given by

$$x_{\text{sgn}} = I_{\text{inv}} x_{\text{pre}} + T_{\text{inv}} \tag{A.2}$$

where

$$I_{\text{inv}} = \begin{bmatrix} \lambda_\phi & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \lambda_\phi & 0 \\ 0 & 0 & 0 & 0 & \lambda_\phi \end{bmatrix}$$

and

$$T_{\text{inv}} = \begin{bmatrix} 0 & 0 & 0 & \frac{\lambda_\phi - 1}{2}\pi & 0 \end{bmatrix}^\top$$

such that I_{inv} and T_{inv} reflect the preloaded trajectory across the x-axis when roll rate is to be negated.

To properly segment the full transition trajectory to appropriately satisfy initial conditions, we begin by defining the index at which the initial aircraft roll matches the preloaded trajectory as

$$m_{\text{s}} = m_z + \lambda_\phi \left\lceil \frac{\phi_0}{\dot{\phi} T_s} \right\rceil \tag{A.3}$$

with the index to the end of the transition segment calculated as

$$m_{\text{e}} = m_z + \lambda_\phi \left\lceil \frac{\phi_f}{\dot{\phi} T_s} \right\rceil. \tag{A.4}$$

Note that since x_{sgn} inverts the roll angle trajectory of x_{pre}, the start and finish indices would be negated for a constant negative roll rate command. The desired trajectory is then all discrete states between the starting and finishing indices of the preloaded or negated transition trajectory estimate.

The transition segment can then be appropriately transformed to meet the initial transition states by rotating the shifting the segment. To rotate the segment, we simply calculate the pre-load offset course angle as

$$\chi_{\text{off}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \end{bmatrix} x_{\text{sgn}}[:, m_{\text{s}}] \tag{A.5}$$

with the full course angle correction being

$$\chi_{\text{corr}} = \chi_0 - \chi_{\text{off}}. \tag{A.6}$$

Positional rotation offsets due to the course angle correction can be rectified using the rotation matrix

$$R_{\text{corr}} = \begin{bmatrix} \cos(\chi_{\text{corr}}) & -\sin(\chi_{\text{corr}}) & 0 \\ \sin(\chi_{\text{corr}}) & \cos(\chi_{\text{corr}}) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{A.7}$$

The pre-load positional offset would be found similarly as

$$p_{\text{off}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} x_{\text{sgn}}[m_{\text{s}}]. \tag{A.8}$$

With the correct transformations defined, we then define the transition trajectory to be

$$p_{tr} = p_0 + R_{\text{corr}} \left(\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} x_{\text{sgn}}[m_{\text{s}} : m_{\text{e}}] - p_{\text{off}} \right), \tag{A.9}$$

$$\chi_{tr} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \end{bmatrix} x_{\text{sgn}}[m_{\text{s}} : m_{\text{e}}] - \chi_{\text{corr}}, \tag{A.10}$$

$$\phi_{tr} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \end{bmatrix} x_{\text{sgn}}[m_{\text{s}} : m_{\text{e}}] \tag{A.11}$$

and

$$x_{tr} = \begin{bmatrix} p_{\text{tr}}^{\top} & \chi_{\text{tr}} & \phi_{\text{tr}} \end{bmatrix}^{\top}, \tag{A.12}$$

where the vector index notation $[a : b]$ indicates the vector subset created by all vector column elements between indices a and b. The terminating index m_{pre} can then be calculated by simply subtracting the starting and ending indices as

$$m_{\text{pre}} = m_{\text{e}} - m_{\text{s}} \tag{A.13}$$

The full discrete transition segment can be calculated once offline for a selected roll rate and then segmented and transformed for all online calculations. The solution rates for Eq. (5.48) using preloaded transition trajectories compared to transition trajectories calculated numerically online at each iteration are shown in Fig. A.1.

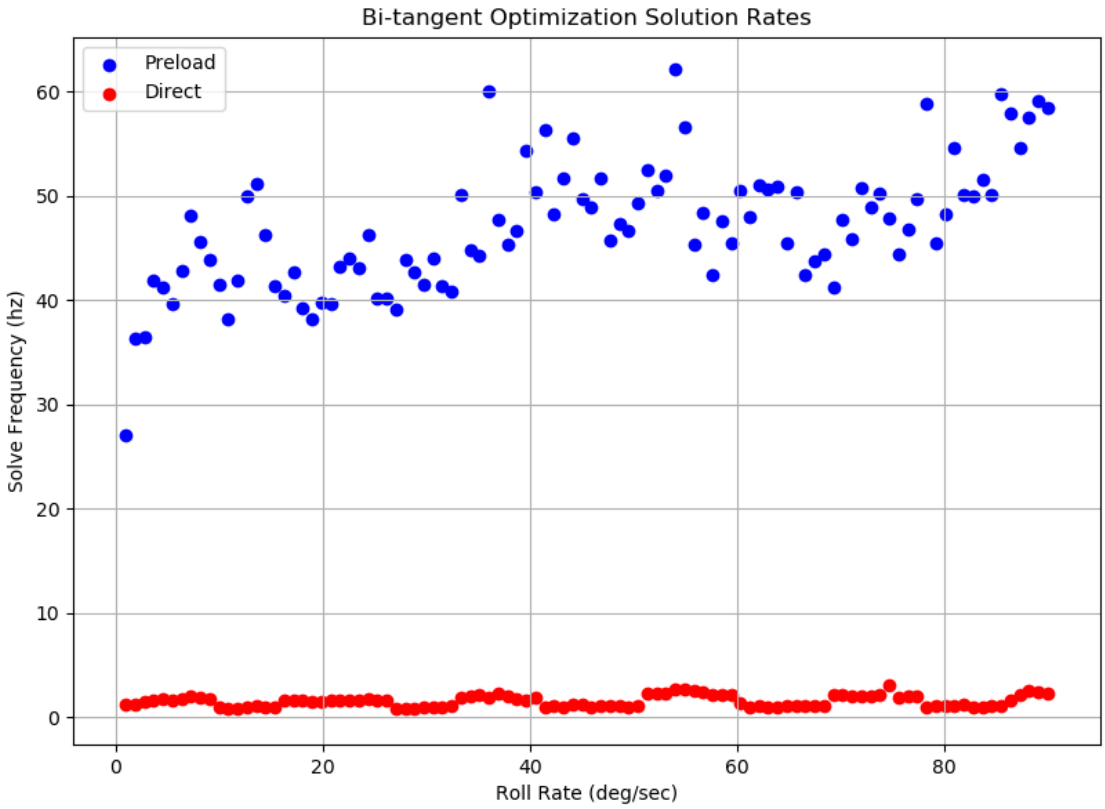


Figure A.1: Solution rate comparison of smoothed bi-tangent solutions using preloaded vs directly calculated transition path segments relative to roll rate magnitude. The preloaded solutions significantly outperform the direct solutions by over a factor of 50 in many cases with equivalent solution accuracy. By sacrificing flexibility in selecting different roll rate commands online, the smoothed bi-tangent solution rate from preloaded transition paths becomes more feasible for real-time performance.