Vision-Based Road-Following Using Proportional Navigation

Ryan S. Holt and Randal W. Beard[*†]

Abstract

This paper describes a new approach for autonomous guidance along a road for an unmanned air vehicle (UAV) using a visual sensor. A road is defined as any continuous, extended, curvilinear feature, which can include city streets, highways, and dirt roads, as well as forest-fire perimeters, shorelines, and fenced borders. To achieve autonomous road-following, this paper utilizes Proportional Navigation as the basis for the guidance law, where visual information is directly fed back into the controller. The tracking target for the Proportional Navigation algorithm is chosen as the position on the edge of the camera frame at which the road flows into the image. Therefore, each frame in the video stream only needs to be searched on the edge of the frame, thereby significantly reducing the computational requirements of the computer vision algorithms. The tracking error defined in the camera reference frame shows that the Proportional Navigation guidance law results in a steady-state error caused by bends and turns in the road, which are perceived as road motion. The guidance algorithm is therefore adjusted using Augmented Proportional Navigation Guidance to account for the perceived road accelerations and to force the steady-state error to zero. The effectiveness of the solution is demonstrated through high-fidelity simulations, and with flight tests using a small autonomous UAV.

1 Introduction

The objective of this paper is to develop a guidance law for small UAV using on-board computer vision to follow roads and other visually distinct contours on the ground. There are a variety of applications that could potentially use this technology including border patrol, pipeline surveillance, powerline maintenance, and military convoy support. We are particularly interested in the application to forest fire monitoring [1], where the boundary of the fire is unknown a priori, and will be changing in time. Fire borders also have the property that they may not be contiguous. However, they will be visually distinct when using a thermal camera. Due to the difficulty associated with

[*] R. Holt is with Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, MA, 02451 (email: rholt@ll.mit.edu).

[†] R. Beard is a professor with the Department of Electrical and Computer Engineering, Brigham Young University, Provo, UT, 84602 (email: beard@byu.edu).

flight testing our approach on actual fires, we focus on vision based guidance algorithms for following rural roads.

Vision based road following has been addressed in other publications [2, 3], where it has been noted that road following consists of two challenges: (1) detecting and estimating the position of the road in the image, and (2) devising a guidance strategy to follow the road. The vision system described in [2] uses color segmentation and a Bayesian pixel classifier to distinguish road from non-road pixels. The position and direction of the road is then determined using connected component analysis, Hough transforms, and a robust line fitting scheme. The vision system described in [3] uses intensity profiles, Hough transforms, and vanishing point analysis to determine potential positions and orientations of the road. A RANSAC algorithm [4] is used to robustly identify the road.

Our approach to road detection is similar to [2] in that we use color segmentation and a connected component analysis to distinguish between road and non-road pixels. However, rather than searching the entire image and identifying the road in the image, we simply identify the position on the boundary of the camera plane where the road flows into the image. The associated image processing does not require image rectification or camera calibration and should represent significantly reduced computational requirements relative to the approaches described in [2] and [3]

In [2], three guidance strategies are proposed for following a road. The first strategy is to wrap a PID loop around the lateral cross track error, which, according to [2], did not work well enough in simulation to warrant flight testing. In the other two approaches reported in [2], the MAV is commanded to track a position on the road a specified distance in front of the projected position of the UAV. The algorithms were demonstrated in simulation and limited flight tests. The guidance strategy proposed in [3] constructs a connecting curve between the current position of the UAV and a position on the desired road a specified distance in front of the projected position of the UAV. This method was also successfully demonstrated with flight results. However, as acknowledged in [3], a weakness of this method is that it will be susceptible to wind.

In this paper we present a guidance strategy that is significantly different than those proposed in [2, 3]. Our approach is based on the Proportional Navigation, and Augmented Proportional Navigation schemes that are well known in the missile guidance literature [5–7]. Proportional Navigation is most commonly used to guide missiles into evading targets, both on the ground and in the air. The basic idea is that the missile will be on a collision course with the target if the line of sight vector between the pursuer and the evader is not rotating in the inertial frame. The proportional navigation algorithm is readily implemented by commanding body frame accelerations that are proportional to the line-of-sight rates multiplied by a closing velocity. Typically, LOS rates are available from sensors on a gimbaled seeker and the closing velocity is taken from radar measurements. Though noisy, these inputs to the guidance law are directly measurable. A Kalman filter can be used to mitigate the sensor noise, and the output of this filter is the input to the guidance law.

Proportional navigation has been extensively analyzed in the literature. It has been shown to be optimal under certain conditions [8] and to produce zero miss distances for a constant target acceleration [9]. If rich information regarding the time-to-go is available, augmented proportional navigation [6] improves the performance by adding terms that account for target and pursuer accelerations. A three-dimensional expression of proportional navigation can be found in [10], where the authors compare proportional navigation to a minimum-energy optimal guidance law.

Proportional navigation has many advantages which account for its use on most of the missile systems that have been deployed in the past forty years. The advantages include robustness to ambient winds, and robustness to sensor calibration and alignment errors [5–7]. If the target that is being tracked is accelerating, then the standard proportional navigation algorithm can be augmented to account for that motion.

The application of proportional navigation to road following is not immediately obvious. In contrast to missile guidance problems, our objective is not to collide with the road, but rather to track the road at a fixed altitude. The primary contribution of this paper is to show how the proportional navigation algorithm can be adapted to vision based road following. The basic idea is that we treat the position that the road flows into the image as the target, and pursue this target as if it were at the same altitude as the UAV. A key insight is that when the road bends and curves, the apparent motion in the image plane is identical to that of an accelerating target. Therefore augmented proportional navigation can be used to track roads with significant curvature. In other words, our method is not restricted to locally linear structures like the methods proposed in [2, 3]. Our method also inherits the well-known strengths of proportional navigation including robustness to wind, and robustness to camera calibration and alignment errors.

One of the significant differences between missiles and small UAVs is that most missiles are skid-to-turn platforms, whereas most small UAVs are bank-to-turn platforms. For missiles that are bank-to-turn platforms, the rolling dynamics are relatively fast and can therefore be approximated as skid-to-turn platforms, whereas that may not be the case for small UAVs. The proportional navigation algorithms are most naturally derived for skid-to-turn models. In this paper we will derive a road following algorithm using a skid-to-turn kinematic model and then show how to map the result to a bank-to-turn vehicle with a single axis gimballed camera.

In Section 2, we define both bank-to-turn and skid-to-turn kinematics models, as well as the relevant reference frames. Section 3 contains the main result of the paper, where we derive the vision based proportional navigation road following algorithm for the skid-to-turn model and then map it to a bank-to-turn platform with a two axis gimbal. In Section 4, we introduce the image steady-state error and use augmented proportional navigation to force the error to zero. In Section 5, the experimental simulation and hardware testbeds are described, and simulation and flight results are presented in Section 6. Conclusions are contained in Section 7.

2 Navigation and Camera Models

To reduce cost and complexity, many small UAV platforms are not fully actuated. In particular, many small UAVs do not have rudder control and must therefore bank to turn. However, because it is much simpler to design a guidance law for skid-to-turn systems, we will begin the derivation of the proportional navigation algorithm with a fully actuated (skid-to-turn) model.

For a fully actuated skid-to-turn vehicle, we assume that the actuators consist of ailerons maintaining zero roll angle while a rudder steers the heading of the vehicle. The kinematics can be modeled:

$$\dot{p}_n = V \cos\chi \cos\gamma, \tag{1}$$
$$\dot{p}_e = V \sin\chi \cos\gamma, \tag{2}$$
$$\dot{h} = V \sin\gamma, \tag{3}$$
$$\dot{\chi} = u_1, \tag{4}$$
$$\dot{\gamma} = u_2, \tag{5}$$

where p_n and p_e are the North and East positions of the UAV, h is the altitude, V is the groundspeed (assumed to be constant), χ is the course angle, γ is the flight path angle, and u_1 and u_2 are control inputs. We will also assume that the skid-to-turn platform is equipped with a strap-down camera, where the camera is attached to the belly of the vehicle and directed downward.

For an under-actuated bank-to-turn vehicle, we assume that the actuators are elevators and ailerons and that winglets are used to maintain the small UAV in a coordinated turn. The kinematics for this type of platform can be modeled as:

$$\dot{p}_n = V \cos\chi \cos\gamma, \tag{6}$$
$$\dot{p}_e = V \sin\chi \cos\gamma, \tag{7}$$
$$\dot{h} = V \sin\gamma, \tag{8}$$
$$\dot{\chi} = \frac{g}{V} \tan\phi, \tag{9}$$
$$\dot{\phi} = u_3, \tag{10}$$
$$\dot{\gamma} = u_4, \tag{11}$$

where g is the gravitational constant and u_3 and u_4 are control inputs. The bank-to-turn platform is assumed to have a gimbaled camera where we define α_{az} to be the gimbal azimuth angle and α_{el} to be the gimbal elevation angle. These angles are assumed to evolve according to the first order model

$$\dot{\alpha}_{az} = u_5, \tag{12}$$
$$\dot{\alpha}_{el} = u_6, \tag{13}$$

where u_5 and u_6 are gimbal controls.

We will briefly define the coordinate frames of interest, specifically, the inertial, vehicle, body, gimbal, and camera coordinate frames, and how they relate to one another. Table 1 presents the notation for the respective axes in each coordinate frame. The superscript on each axis is used to denote the reference frame with respect to which the vector is defined.

Figures 1 and 2 illustrate each of the coordinate frames and their corresponding relationships. For the inertial frame, we use the NED coordinates $(p_n, p_e, -h)$. The relationship between the inertial frame and the vehicle frame is a simple translation from the origin of the inertial frame to the center of mass (CM) of the UAV. The body

Table 1: Notation of Coordinate Frame Axes.

Coordinate Frame	Axes
Inertial	$\left(X^i, Y^i, Z^i\right)$
Vehicle	(X^v, Y^v, Z^v)
Body	$\left(X^b, Y^b, Z^b\right)$
Gimbal	(X^g, Y^g, Z^g)
Camera	(X^c, Y^c, Z^c)

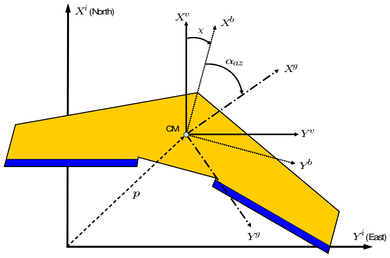


Figure 1: A lateral perspective of the coordinate frames. The inertial and vehicle frames are aligned with the world, the body frame is aligned along the airframe, and the gimbal and camera frames are aligned along the camera.

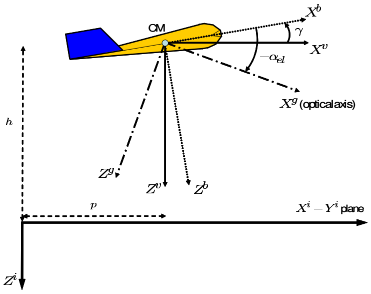


Figure 2: A longitudinal perspective of the coordinate frames.

frame requires a rotation from the vehicle frame so that X^b is directed out the nose, Y^b is directed out the right wing, and Z^b is directed out the belly of the vehicle.

In general, the rotation of a vector from coordinate system j to coordinate system k can be expressed as a rotation matrix R_j^k. Because every rotation matrix is unitary, its inverse is equal to its transpose: $R_k^j = (R_j^k)^{-1} \triangleq (R_j^k)^T$. For the vehicle-to-body rotation, the rotation matrix is

$$R_v^b = \begin{pmatrix} c_\gamma c_\chi & c_\gamma s_\chi & -s_\gamma \\ s_\phi s_\gamma c_\chi - c_\phi s_\chi & s_\phi s_\gamma s_\chi + c_\phi c_\chi & s_\phi c_\gamma \\ c_\phi s_\gamma c_\chi + s_\phi s_\chi & c_\phi s_\gamma s_\chi - s_\phi c_\chi & c_\phi c_\gamma \end{pmatrix}, \tag{14}$$

where $c_* \triangleq \cos(*)$ and $s_* \triangleq \sin(*)$. Assuming that the camera is located at the CM of the UAV, the rotation from the body coordinate system to the gimbal coordinate system is defined as

$$R_b^g = \begin{pmatrix} c_{\alpha_{el}} c_{\alpha_{az}} & c_{\alpha_{el}} s_{\alpha_{az}} & -s_{\alpha_{el}} \\ -s_{\alpha_{az}} & c_{\alpha_{az}} & 0 \\ s_{\alpha_{el}} c_{\alpha_{az}} & s_{\alpha_{el}} s_{\alpha_{az}} & c_{\alpha_{el}} \end{pmatrix}. \tag{15}$$

In the camera frame, X^c is directed from the left side of the image to the right, Y^c is directed from the top of the image to the bottom, and Z^c is directed along the optical axis of the camera. The origin for the camera frame is located in the center of the image. The rotation from the gimbal coordinate system to the camera coordinate system is

$$R_g^c = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}. \tag{16}$$

3 Vision-Based Proportional Navigation

Our objective is to design a vision-based guidance law that commands the UAV to track a road. In order for the guidance law to be vision based, image processing is used to extract the location of the road in the image. While this paper will not focus on the image-processing algorithm, we will briefly describe the necessary information that is to be extracted from the image. Once this information is defined, we will derive the proportional-navigation equations for the skid-to-turn model equipped with a strap-down camera. We will then extend those equations to the bank-to-turn model using a gimbaled camera.

3.1 Image-Processing Algorithm

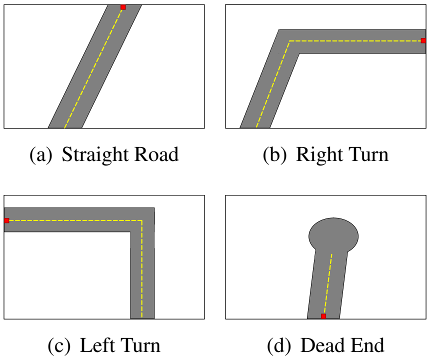
In the missile literature, the proportional navigation (PN) guidance law is based upon the relative position and velocity between the interceptor and the target [5]. Therefore, in order to use PN for road-following, we must define both the position of the road as a single point and its associated velocity in the camera frame. Our approach is unique in that most road-following techniques involve fitting a line to the road rather than a single point [2,11–13]. As will be shown, the amount of computations for representing the road by a point is significantly less than fitting a line to the road. We begin by defining the position and velocity of the road in the camera frame.

In road-following, the desired behavior is for the UAV to travel directly above the road in the inertial frame. This allows the UAV more time to react to changes in the direction of the road. We define the configuration where the UAV is traveling directly above the road to be the desired, steady-state behavior that the UAV should converge to. Once steady state is reached, any deviation of the road or change in the position of the UAV is viewed as a disturbance to the system. The UAV should adjust to disturbances and return to steady state.

In steady state, the road will be centered vertically in the image, extending from the top to the bottom. We will refer to the road as *entering* from the top and *exiting* from the bottom of the image. Note that once the road is in the image, there will always be an entrance as well as an exit location. For example, in the case where there is a 90 degree right-hand turn in the image, the entrance location is the right side of the image and the exit location is the bottom of the image. In the case where the road is a dead-end, the entrance location is the same as the exit location at the bottom of the image.

For a downward-facing camera, the center of the image corresponds to the North/East position of the UAV. To have sufficient time to react to any deviation in the road direction, the UAV must look beyond its current position. Therefore, the UAV must use the information pertaining to the road in the upper half of the image. In order to maximize reaction time to road deviations while in steady-state, we define the position of the road to be the pixel on edge of the image corresponding to the center of the road precisely where the road *enters* the image. Figure 3 illustrates which pixel location is identified as the road by the image-processing algorithm for four different scenarios. The velocity of the road is defined as the rate of change of this pixel in the video sequence.

For this paper, a simple thresholding in the HSV (hue-saturation-value) color space was used to separate road pixels from non-road pixels [14]. This was followed by a connected-components analysis that groups road pixels together into components. The largest component was then selected as the road and the top center pixel was returned to the UAV controller. While this approach was simple and effective, there are a variety of other algorithms that could be used for image processing as long as the delivery time of the needed information met the demands of the controller to ensure stable control.


(a) Straight Road

(b) Right Turn

(c) Left Turn

(d) Dead End

Figure 3: Images from the perspective of the UAV in four different road scenarios, where the pixel corresponding to the road position returned by the image-processing algorithm is highlighted in red.

When fitting a line to the road, the entire image needs to be searched. However, by following the approach described above, we can significantly limit the number of image pixels that are processed. Most of the time the road will enter from the top of the image, which implies that only the top section needs to be searched. If the road is not detected in the top portion of the image, then the sides can be processed. If that fails, then the bottom of the image can be searched. If this fails as well, then the road is not in the image. Therefore, the center portion of the image does not need to be processed, thereby significantly speeding up the algorithm.

The size definition of the top, sides, and bottom sections is dependent upon the quality of the road classification. If the road in the image is hard to extract from its surrounding environment, then the sections should be made larger. To give an example of adequate sizes, the flight-tests described in this paper were performed using a top section of 640×140 pixels, side sections of 100×480 pixels, and a bottom sections of 640×100 pixels, where the entire image was 640×480. In the best case, which occurs most of the time, only the top section is searched limiting the analyzed pixels to 29.17% of the total number of pixels. In the worst case, where all the sections are searched, the analyzed pixels are limited to 57.81% of the total number of pixels. This is a significant improvement in the number of pixel operations needed to extract the road information from the image.

3.2 Skid-to-Turn Model with Strap-Down Camera

This section derives the guidance law for vision-based road-following for a skid-to-turn vehicle model. We will follow the derivation of 3D proportional navigation presented in [15], adjusting the equations and assumptions for road-following.

We begin by defining the inertial position and velocity of the UAV as

$$r^i_{UAV} = (p_n, p_e, -h)^T \tag{17}$$

and

$$v^i_{UAV} = \begin{pmatrix} V\cos(\chi)\cos(\gamma) \\ V\sin(\chi)\cos(\gamma) \\ -V\sin(\gamma) \end{pmatrix}. \tag{18}$$

Assuming a small angle-of-attack, the velocity vector of the UAV can be approximated in body coordinates as

$$v^b_{UAV} = (V, 0, 0)^T. \tag{19}$$

The inertial position and velocity of the road are denoted as r^i_{road} and v^i_{road}. Let $R^i \triangleq r^i_{road} - r^i_{UAV}$ be the relative position vector between the road and the UAV expressed in inertial coordinates.

The 3D equations for pure proportional navigation guidance (PPNG) are given in [10] as

$$a_{UAV} = N\Omega_\perp \times v_{UAV}, \tag{20}$$

where N is the navigation constant and

$$\Omega_\perp = \frac{R \times (v_{road} - v_{UAV})}{\|R\|^2}, \tag{21}$$

is the component of the angular rate of the LOS that is perpendicular to the LOS. By noting that $\dot{R} = \dot{r}_{road} - \dot{r}_{UAV} = v_{road} - v_{UAV}$ and defining

$$\hat{\ell} \triangleq \frac{R}{\|R\|} \tag{22}$$

and

$$\hat{w} \triangleq \frac{\dot{R}}{\|R\|}, \tag{23}$$

we can express $\Omega_\perp$ as

$$\begin{aligned}\Omega_\perp &= \frac{R}{\|R\|} \times \frac{\dot{R}}{\|R\|} \\ &= \hat{\ell} \times \hat{w}.\end{aligned} \tag{24}$$

The control inputs for the models defined in Section 2 are in body coordinates. Therefore, the acceleration command in Equation (20) should also be expressed in body coordinates as

$$a^b_{UAV} = N\Omega^b_\perp \times v^b_{UAV}. \tag{25}$$

Using Equation (19), Equation (25) becomes

$$a^b_{UAV} = N \begin{pmatrix} \Omega^b_{\perp,x} \\ \Omega^b_{\perp,y} \\ \Omega^b_{\perp,z} \end{pmatrix} \times \begin{pmatrix} V \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ NV\Omega^b_{\perp,z} \\ -NV\Omega^b_{\perp,y} \end{pmatrix}. \tag{26}$$

To derive the commanded acceleration, we need to estimate $\Omega^b_\perp$ from the visual information. Our approach will be to estimate $\Omega_\perp$ in camera coordinates and then transform it into body coordinates using

$$\Omega^b_\perp = R^b_g R^g_c \Omega^c_\perp. \tag{27}$$

Let ϵ_x and ϵ_y be the X and Y pixel location corresponding to the position of the road as described in Section 3.1 and let $\dot{\epsilon}_x$ and $\dot{\epsilon}_y$ be the X and Y time derivatives of the pixel location. The focal length f of the camera in units of pixels is defined as

$$f = \frac{M_W}{2\tan(\frac{\eta}{2})}, \tag{28}$$

where η is the horizontal field-of-view and M_W is the width of the image in pixels.

The 3D geometry of the camera model is shown in Figure 4. The displacement of the road from the origin of the camera in the image plane is the vector having a direction of $(P\epsilon_x, P\epsilon_y, Pf)$ and a magnitude of $P\bar{f}$, where

$$\bar{f} = \sqrt{\epsilon_x^2 + \epsilon_y^2 + f^2}, \tag{29}$$

and where P is a conversion factor from pixels to meters. The displacement of the road from the origin of the camera in the world plane is the vector (R^c_x, R^c_y, R^c_z) having a magnitude of $\|R\|$.

By using similar triangles, we can form ratios among these displacements as

$$\hat{\ell}^c_x = \frac{R^c_x}{\|R\|} = \frac{P\epsilon_x}{P\bar{f}} = \frac{\epsilon_x}{\bar{f}}, \tag{30}$$

$$\hat{\ell}^c_y = \frac{R^c_y}{\|R\|} = \frac{P\epsilon_y}{P\bar{f}} = \frac{\epsilon_y}{\bar{f}}, \tag{31}$$

$$\hat{\ell}^c_z = \frac{R^c_z}{\|R\|} = \frac{Pf}{P\bar{f}} = \frac{f}{\bar{f}}. \tag{32}$$

Thus, the first term on the right side of Equation (24) is expressed as

$$\hat{\ell}^c = \frac{1}{\bar{f}} \begin{pmatrix} \epsilon_x \\ \epsilon_y \\ f \end{pmatrix}. \tag{33}$$

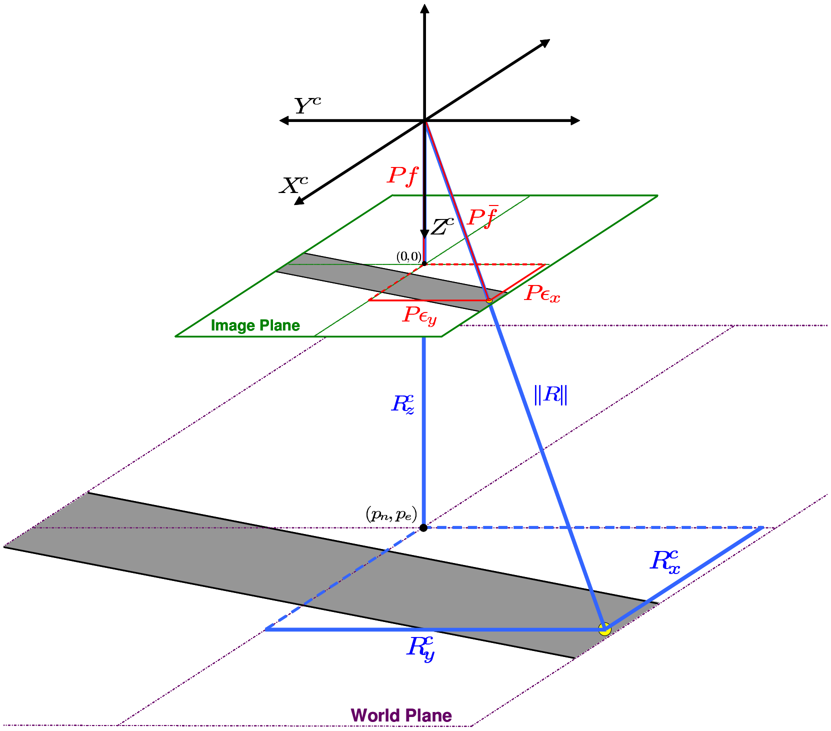


Figure 4: The 3D camera model. The displacements of the road in the image plane are indicated in red while the displacements in the world plane are shown in blue.

From the definition $\hat{\ell} = R / \|R\|$ we have that

$$\begin{aligned}\frac{d\hat{\ell}}{dt} &= \frac{\|R\|\, \dot{R} - R\frac{d\|R\|}{dt}}{\|R\|^2} \\ &= \frac{\dot{R}}{\|R\|} - \frac{R}{\|R\|}\frac{\frac{d\|R\|}{dt}}{\|R\|}.\end{aligned}$$

Therefore, since $\hat{w} = \dot{R} / \|R\|$ we have

$$\hat{w} = \frac{d\hat{\ell}}{dt} + \hat{\ell}\frac{\frac{d\|R\|}{dt}}{\|R\|}. \tag{34}$$

From Eq. (33) we have

$$\begin{aligned}\frac{d\hat{\ell}^c}{dt} &= \frac{\bar{f}\begin{pmatrix}\dot{\epsilon}_x \\ \dot{\epsilon}_y \\ 0\end{pmatrix} - \dot{\bar{f}}\begin{pmatrix}\epsilon_x \\ \epsilon_y \\ f\end{pmatrix}}{\bar{f}^2} \\ &= \frac{1}{\bar{f}}\begin{pmatrix}\dot{\epsilon}_x \\ \dot{\epsilon}_y \\ 0\end{pmatrix} - \frac{\dot{\bar{f}}}{\bar{f}}\hat{\ell}^c.\end{aligned} \tag{35}$$

From the relation $\|R\| = R^c_z \frac{\bar{f}}{f}$ we have

$$\begin{aligned}\frac{d\,\|R\|}{dt} &= \dot{R}^c_z \frac{\bar{f}}{f} + R^c_z \left(\frac{f\dot{\bar{f}} - \bar{f}\dot{f}}{f^2}\right) \\ &= R^c_z \frac{\dot{\bar{f}}}{f} \\ &= \|R\|\,\frac{\dot{\bar{f}}}{\bar{f}}.\end{aligned}$$

Therefore

$$\frac{\frac{d\|R\|}{dt}}{\|R\|} = \frac{\dot{\bar{f}}}{\bar{f}}. \tag{36}$$

Substituting (35) and (36) into (34) gives

$$\hat{w}^c = \frac{1}{\bar{f}}\begin{pmatrix}\dot{\epsilon}_x \\ \dot{\epsilon}_y \\ 0\end{pmatrix}. \tag{37}$$

We have shown that $\hat{\ell}^c$ and $\hat{w}^c$ can be estimated solely upon the visual information.

Therefore, $\Omega_\perp^c$ can be expressed as

$$\begin{aligned}\Omega_\perp^c &= \hat{\ell}^c \times \hat{w}^c \\ &= \frac{1}{\bar{f}^2}\begin{pmatrix} -\dot{\epsilon}_y f \\ \dot{\epsilon}_x f \\ \epsilon_x \dot{\epsilon}_y - \epsilon_y \dot{\epsilon}_x \end{pmatrix}. \end{aligned} \tag{38}$$

We now rotate $\Omega_\perp^c$ into body coordinates using the expression

$$\begin{aligned}\Omega_\perp^b &= R_g^b R_c^g \Omega_\perp^c \\ &= \frac{1}{\bar{f}^2}\begin{pmatrix} -\dot{\epsilon}_x f \\ -\dot{\epsilon}_y f \\ \epsilon_x \dot{\epsilon}_y - \epsilon_y \dot{\epsilon}_x \end{pmatrix}. \end{aligned} \tag{39}$$

By using Equations (26) and (39), the PPNG acceleration in body coordinates is

$$\begin{aligned} a_{UAV}^b &= \begin{pmatrix} 0 \\ NV\Omega_{\perp,z}^b \\ -NV\Omega_{\perp,y}^b \end{pmatrix} \\ &= \frac{1}{\bar{f}^2}\begin{pmatrix} 0 \\ NV\left(\epsilon_x \dot{\epsilon}_y - \epsilon_y \dot{\epsilon}_x\right) \\ NV\left(\dot{\epsilon}_y f\right) \end{pmatrix}. \end{aligned} \tag{40}$$

In target-tracking, $a_{UAV,y}^b$ will direct the heading angle of the UAV toward the inertial position of the target, while $a_{UAV,z}^b$ will direct the pitch angle toward the altitude of the target. However, in road-following, the UAV is not trying to intercept the road but rather maintain constant altitude while tracking the road. Therefore, $a_{UAV,z}^b$ is ignored and the desired altitude is commanded using a proportional controller. The angular acceleration $\dot{\chi}$ is related to the linear acceleration $a_{UAV,y}^b$ by the equation

$$\dot{\chi} = \frac{1}{V} a_{UAV,y}^b. \tag{41}$$

We summarize the derivation for the skid-to-turn model with a strap-down camera by defining the control inputs in Equations (4) and (5) as

$$u_1 = N\left(\frac{\epsilon_x \dot{\epsilon}_y - \epsilon_y \dot{\epsilon}_x}{\epsilon_x^2 + \epsilon_y^2 + f^2}\right) \tag{42}$$

and

$$u_2 = k_h(h^d - h), \tag{43}$$

where k_h is a proportional gain, and h^d is the desired altitude.

3.3 Bank-to-Turn Model with Gimbaled Camera

In this section, we derive the proportional navigation equations for the bank-to-turn model equipped with a gimbaled camera. We begin the derivation by noting that if the camera is directed downward along the Z^v-axis, regardless of the pitch and roll of the UAV, the camera always has the same position and orientation as a strapped-down camera on a skid-to-turn UAV. Therefore, if the gimbal can be commanded to cancel out the pitch and roll effects of the UAV, the results derived in Section 3.2 are directly applicable to the bank-to-turn model.

In Section 3.2, we assumed that the UAV maintains constant altitude. If winglets maintain the vehicle in a coordinated turn, then the pitch angle is essentially zero. Therefore, the gimbal need only cancel out the effects caused by rolling. A roll is defined as a rotation about the X^v-axis that has been directed to coincide with the heading of the UAV. If the gimbal is oriented so that it also rotates about this axis, a rotation angle equal to the roll angle and opposite in direction will cancel out the effects of the roll on the image. To implement this idea, we position the gimbal so its initial orientation is pointed downward along the Z^b-axis, while the UAV is in level flight. The gimbal is then only allowed to move in the Y^b-Z^b plane. If the azimuth angle $\alpha_{az} = \pi/2$, then the negative elevation angle $-\alpha_{el}$ is the displacement angle from the Y^b-axis in the Y^b-Z^b plane. Defining $\tilde{\alpha}_{el} = \alpha_{el} + \pi/2$, then when $\tilde{\alpha}_{el} > 0$, the gimbal will rotate toward the right wing and when $\tilde{\alpha}_{el} < 0$ the gimbal will rotate toward the left wing. Note that this rotation is defined opposite in direction to that of a UAV roll angle.

With this setup, the equations in Section 3.2 are used to derive the control inputs defined in Equations (10), (11), (12), and (13). In Equation (4), u_1 commands $\dot{\chi}$, which we can combine with Equation (9) to derive a desired roll angle:

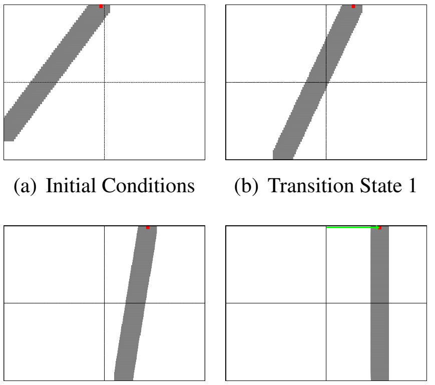
$$\begin{aligned}\phi^d &= \frac{V}{g}\arctan(\dot{\chi}) \\ &= \frac{V}{g}\arctan(u_1). \qquad (44)\end{aligned}$$

Using the desired roll angle, ϕ^d, we can define the control inputs introduced in Equations (10)-(13) for the bank-to-turn model equipped with a gimbaled camera as

$$\begin{aligned}u_3 &= k_\phi\left(\phi^d - \phi\right), & (45)\\ u_4 &= k_h\left(h^d - h\right), & (46)\\ u_5 &= 0, & (47)\\ u_6 &= k_{el}\left(\phi - \frac{pi}{2} + \alpha_{el}\right), & (48)\end{aligned}$$

where k_ϕ, k_h, and k_{el} are positive control gains. Note that for the above assumptions to hold, α_{el} needs to be updated frequently enough to track the roll angle. Also, the response of the gimbal needs to be fast enough to cancel the UAV roll dynamics.

By using u_1 to calculate u_3, the rotation matrix R_g^b is equivalent to the one expressed in the skid-to-turn model derivation and is *not* recomputed with the new camera



(a) Initial Conditions (b) Transition State 1

(c) Transition State 2 (d) Steady-state Conditions

Figure 5: Four screen shots of the road-following algorithm in a Matlab simulation, demonstrating the steady-state error where (a) is the initial state, (b) and (c) are different states that occurred during the transition to steady-state, and (d) is the steady-state condition. The green vector represents the image steady-state error.

elevation angle. By assuming that the camera is directed downward along the Z^i-axis, the controller is unaware that the vehicle is rolling and that the gimbal is rotating.

4 Image Steady-State Error

In this section, we introduce the idea of an image steady-state error and present an adjustment to the guidance laws developed in Sections 3.2 and 3.3 to force this error to zero. Consider the situation for the skid-to-turn model when the road is vertical in the image but not centered. Both $\dot{\epsilon}_x$ and $\dot{\epsilon}_y$ are zero, causing the control input u_1 to be zero, and the UAV no longer maneuvers to center itself over the road. Figure 5 demonstrates this situation by showing snapshots from a numerical simulation: (a) represents initial conditions, (b) and (c) take place during the tracking, and (d) occurs when the UAV has reached a steady state. While the UAV is still tracking the road in this setting, it is non-ideal. If a sharp turn in the road were to occur, the time to react would be severely limited and may cause the road to leave the image before the control can be adjusted. Hence, it is important to have the position of the UAV converge directly over the position of the road. In other words, the road should be centered in the image. We define the image steady-state error as the displacement of the road position from the top center pixel of the image. The green vector in Figure 5(d) represents the image steady-state error.

The image steady-state error results from the manner in which the image processing was designed. Selecting the top pixel corresponding to the road for each video frame produces the effect that the road is traveling at the same velocity as the UAV. Proportional navigation is designed to command the UAV to intercept the target. However, if the UAV and target are traveling at the same velocity, any movement by the UAV that is not in the direction of the target's heading will cause the UAV to move farther away from the target. Therefore, the best maneuver for the UAV is to direct its heading to that of the target so that the two travel in parallel. We can classify the effective road velocity between video frames as a maneuver by the road, since it changes as the vehicle's velocity changes. Any curves or turns in the road can also be viewed as a maneuver.

Equation (20) has been shown to produce zero miss-distance for targets with zero acceleration, but for maneuverable targets the commanded acceleration needs to be adjusted [16]. This is referred to in the missile guidance literature as augmented proportional navigation guidance (APNG) where the adjustment to the control law is based upon the acceleration of the target.

To describe the acceleration of the road, first note that if there is no acceleration the road position should converge to the top center of the image. If the given pixel location is not at the desired location then the road is viewed as having performed a maneuver. Therefore, the image steady-state error can be used to augment the PPNG law.

The image steady-state error in the camera frame is defined as

$$e_{ss}^c = \begin{pmatrix} \frac{\epsilon_x - \epsilon_x^d}{M_W} + \operatorname{sign}(\epsilon_x - \epsilon_x^d)\frac{\left|\epsilon_y - \epsilon_y^d\right|}{M_H} \\ \frac{\epsilon_y - \epsilon_y^d}{M_H} + \operatorname{sign}(\epsilon_y - \epsilon_y^d)\frac{\left|\epsilon_x - \epsilon_x^d\right|}{M_W} \\ 0 \end{pmatrix}, \tag{49}$$

where M_W is the number of pixels across the width of the image, M_H is the number of pixels across the height, and the *sign* function is defined as

$$\operatorname{sign}(x) \triangleq \begin{cases} 1, & \text{if } x \geq 0 \\ -1, & \text{if } x < 0 \end{cases}. \tag{50}$$

The first term in each error component is the main error source in that dimension. The second term is used as an additional push in the correct direction when the road deviation is large. To augment the acceleration command in Equation (40), e_{ss}^c must be rotated into the body frame as

$$\begin{aligned} e_{ss}^b &= R_g^b R_c^g e_{ss}^c \\ &= \begin{pmatrix} -\frac{\epsilon_y - \epsilon_y^d}{M_H} + \operatorname{sign}(\epsilon_y - \epsilon_y^d)\frac{\left|\epsilon_x - \epsilon_x^d\right|}{M_W} \\ \frac{\epsilon_x - \epsilon_x^d}{M_W} + \operatorname{sign}(\epsilon_x - \epsilon_x^d)\frac{\left|\epsilon_y - \epsilon_y^d\right|}{M_H} \\ 0 \end{pmatrix}. \end{aligned} \tag{51}$$

The new command input is therefore

$$\begin{aligned} u_1 = N &\left(\frac{\epsilon_x \dot{\epsilon}_y - \epsilon_y \dot{\epsilon}_x}{\bar{f}^2}\right) \\ &+ L\left(\frac{\epsilon_x - \epsilon_x^d}{M_W} + \operatorname{sign}(\epsilon_x - \epsilon_x^d)\frac{\left|\epsilon_y - \epsilon_y^d\right|}{M_H}\right), \end{aligned} \tag{52}$$

where L is a tunable gain. The first term on the right side of the equation above will be zero only when the road is vertical in the image. The second term on the right side of Equation (52) will be zero only when the top of the road is centered in the image. The combination of these two terms guarantees a single, steady-state condition where the road is vertical and centered in the image. Note that the only adjustment to the control inputs defined in Sections 3.2 and 3.3 is to u_1, since u_2, u_4, u_5, and u_6 are independent of the PPNG law and u_3 will be adjusted by the augmentation to u_1.

5 Experimental Testbed

This section describes the experimental platform used to test and validate the road following algorithm developed in the previous section. The testbed is comprised of both hardware and software. The hardware consists of a UAV equipped with an autopilot and various sensors. The software includes a simulation program that is used to test the autopilot code, and a program that is used for interfacing with both the simulator and the actual autopilot. Figure 6 is a flowchart describing the interaction of the hardware and software.

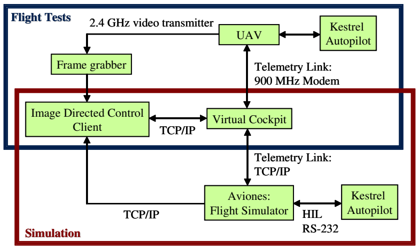
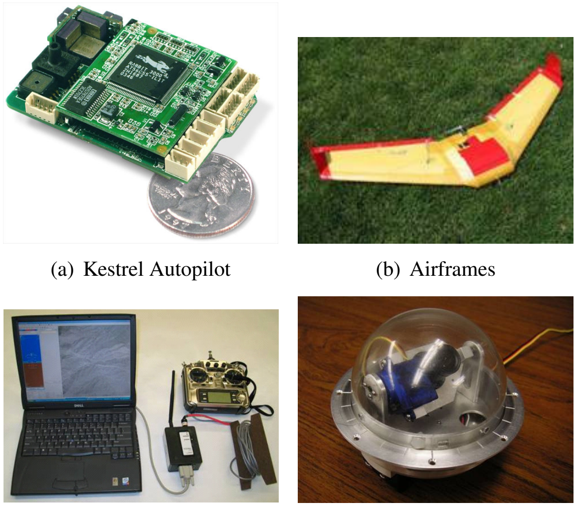


Figure 6: A flowchart depicting the layout of the basic hardware and software components used both in simulation and actual flight-tests.

5.1 Hardware

Figure 7 displays the main hardware components of the experimental testbed. Figure 7(a) shows the Kestrel autopilot designed at Brigham Young University (BYU) and manufactured by Procerus Technologies. It is equipped with a Rabbit 3000 29-MHz microprocessor, rate gyroscopes, accelerometers, absolute and differential pressure sensors. The autopilot measures $2.0 \times 1.37 \times 0.47$ inches and weighs 16.65 grams, making it ideal for small aircraft. Figure 7(b) shows the airframe used in the flight tests. It is a custom designed flying wing constructed of expanded polypropylene (EPP) foam and coated with Kevlar, with a wingspan of 48 inches. This airframe is easily hand-deployable and resilient to crashes, making it a good research and test vehicle. Embedded within the foam are the autopilot, batteries, a 1000-mW/900-MHz radio modem, a GPS receiver, a video transmitter, and a small analog camera mounted on a pan-tilt gimbal. The fully loaded weight is just under two pounds. With this setup the UAV can fly for approximately 40 minutes at a cruise speed of 13 m/s. The gimbal, which was



(a) Kestrel Autopilot

(b) Airframes

(c) Ground Station

(d) Gimbal

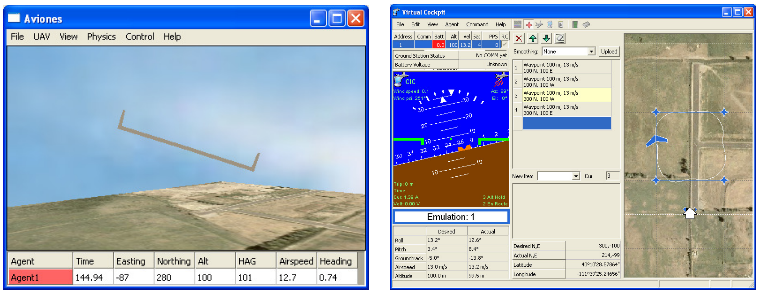
Figure 7: The hardware components used in flight-testing the algorithms presented in this paper. In (a), the Kestrel Autopilot is shown, in (b), the airframe is displayed, in (c), the components of the ground station are laid out, and (d) shows the gimbal.

designed at BYU is shown in Figure 7(d). The azimuth axis of the gimbal is actuated by a Hitec HS-77BB servo with 61.1 oz-in of torque. The elevation axis is actuated by a Hitec HS-50 servo with 8.22 oz-in of torque. The camera is a Panasonic KX-141 camera with 480 lines of resolution. Figure 7(c) shows the components that comprise the ground station. A laptop runs the Virtual Cockpit ground-control software, allowing the user to interface with the UAV via a communication box. A remote control (RC) transmitter also is attached to the communication box, acting as a standby fail-safe mechanism to facilitate safe operation.

5.2 Software

There are two main software applications that are used in simulating and testing the developed algorithms. The first, Aviones, is a flight simulator developed at BYU, which emulates both the physics of the airframe as well as the communication between the ground station and the UAV. The motion of each virtual UAV is calculated from full nonlinear, six-degree-of-freedom equations of motion [17]. Aviones is adaptable to many styles of aircraft and various terrain conditions. The most powerful aspect of Aviones is that the autopilot code tested in simulation is identical to the code on the actual autopilot, allowing for quick transfer from software development to flight-testing. Figure 8(a) shows a screen capture of the Aviones software during a simulated flight-test.

The second piece of software that is utilized in both simulation and flight-testing is Virtual Cockpit. Virtual Cockpit connects to Aviones through a TCP/IP connection



Agent	Time	Easting	Northing	Alt	HAG	Airspeed	Heading
Agent1	144.94	-87	280	100	101	12.7	0.74

Address	Comm	Batt	Alt	Vel	Sat	PPS	RC
1		0.0	100	13.2	4	0	✓

Ground Station Status	No COMM yet
Battery Voltage	Unknown

Wind speed: 0.1
Wind psi: 251°
Az: 89°
El: 0°
Trip: 0 m
Time:
Cur: 1.39 A
Volt: 0.00 V
3 Alt Hold
2 En Route

	Desired	Actual
Roll	13.2°	12.6°
Pitch	3.4°	8.4°
Groundtrack	-5.0°	-13.8°
Airspeed	13.0 m/s	13.2 m/s
Altitude	100.0 m	99.5 m

Smoothing: None

1	Waypoint 100 m, 13 m/s 100 N, 100 E
2	Waypoint 100 m, 13 m/s 100 N, 100 W
3	Waypoint 100 m, 13 m/s 300 N, 100 W
4	Waypoint 100 m, 13 m/s 300 N, 100 E

New Item
Cur 3

Desired N,E	300,-100
Actual N,E	214,-99
Latitude	40°10'28.57864"
Longitude	-111°39'25.24656"

(a) Screen Capture of Aviones

(b) Screen Capture of Virtual Cockpit

Figure 8: (a) Screen capture of Aviones with a simulated aircraft in flight. (b) Screen capture of Virtual Cockpit with the navigation screen displayed on the right-hand side. The UAV tracks the path defined by the four blue stars. The artificial horizon along with the desired and actual attitude values are located on the left-hand side.

and allows the user to interface with the simulated UAV in the same manner as during a real-world flight-test. In both simulation and testing, Virtual Cockpit allows the user to view the current attitude of the UAV, tune the control loops in real-time, adjust various algorithmic variables, and command a list of waypoints for the UAV to follow. Additionally, a frame processor may be connected to Virtual Cockpit through a TCP/IP connection, allowing for video images from the UAV to be converted into control commands and sent to the UAV. Figure 8(b) shows a screen capture of the Virtual Cockpit software.

6 Results

6.1 Simulation Results

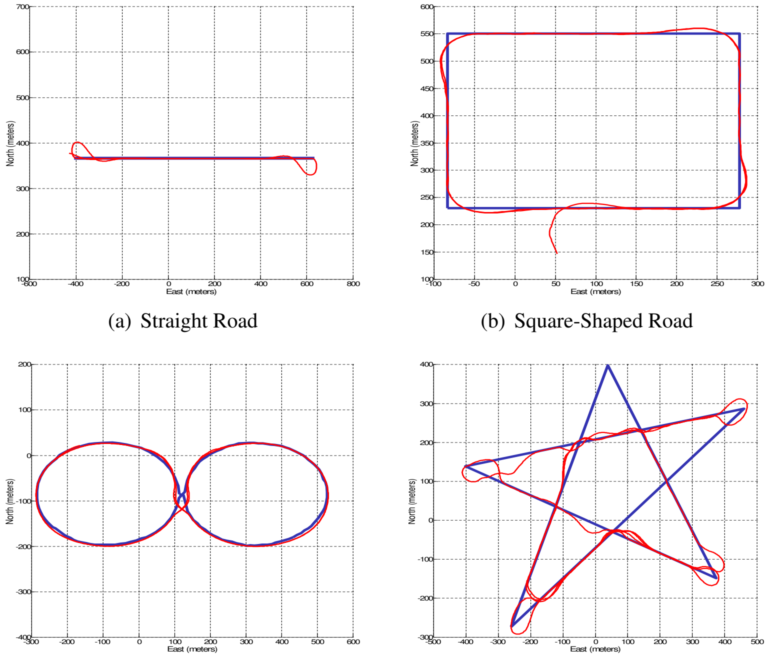
To verify the feasibility of the developed road-following guidance law, medium fidelity simulations were conducted using the software described in Section 5.2. The UAV model used in the simulations was the bank-to-turn platform with a gimbaled camera, where the following constraints were imposed upon the air vehicles:

$$
\begin{aligned}
V &= 13\ m/s, && (53)\\
-45\text{ deg} \le \phi &\le 45\text{ deg}, && (54)\\
-60\text{ deg} \le \alpha_{el} &\le 60\text{ deg}, && (55)\\
-40\text{ deg}/s \le u_1 &\le 40\text{ deg}/s, && (56)\\
-60\text{ deg}/s \le u_2, u_4 &\le 60\text{ deg}/s, && (57)
\end{aligned}
$$

and

$$-100\text{ deg}/s \le u_3 \le 100\text{ deg}/s. \tag{58}$$

The simulations involved four different scenarios where the road configuration was a straight line, a square, a figure eight, and a star. Each scenario consisted of binary video footage, where road pixels were white and non-road pixels were black. Figure 9 shows the inertial position of the road (blue) for each scenario overlaid by the position of the UAV (red).



(a) Straight Road

(b) Square-Shaped Road

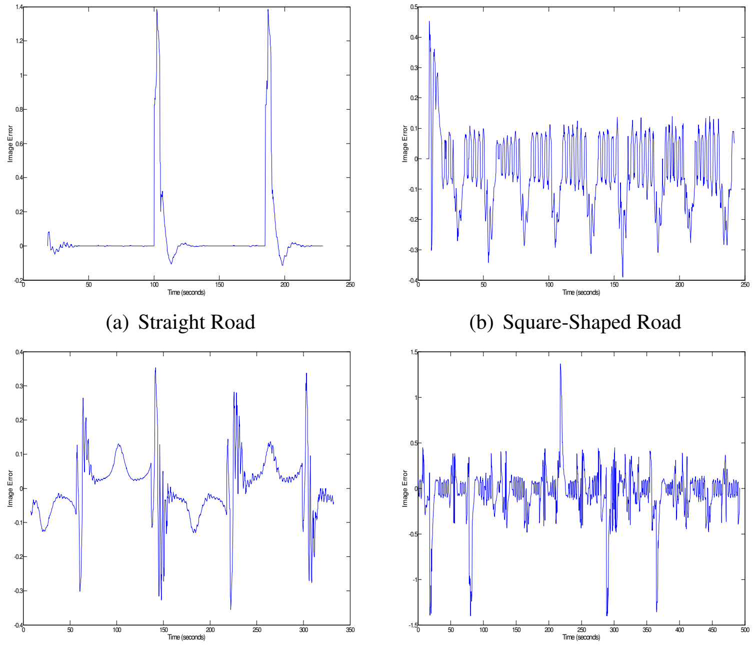
(c) Figure-Eight-Shaped Road

(d) Star-Shaped Road

Figure 9: A plot of the road position used in simulation overlaid by the GPS position of the UAV for four different scenarios, where (a) is a straight road, (b) is a road in the shape of a box, (c) is a road in the shape of a figure eight, and (d) is a road in the shape of a star.

In the first three scenarios, the guidance law performed as expected, with the position of the UAV converging over the center of the road. When a change in the road direction occurred, the UAV responded and tracked the change. The image steady-state error plots are displayed in Figure 10. The large spikes in these plots correspond to sharp changes in the road position and are quickly driven to zero.

In the star-shaped scenario, the image steady-state error also goes to zero after transitions in the road. However, the UAV did not track the entire shape like it did in the other scenarios. This is a result of how the image-processing algorithm functions when multiple roads are in the image at the same instance. Sometimes the algorithm switched between which road it selected, causing the UAV to change road segments. However, despite this issue, the guidance algorithm still maneuvered the UAV to whichever road was returned by the image-processing algorithm. In order to allow the desired road to be selected every time, even if the image contains multiple roads, decision-making capabilities must be added to the image-processing algorithm.



(a) Straight Road

(b) Square-Shaped Road

(c) Figure-Eight-Shaped Road

(d) Star-Shaped Road

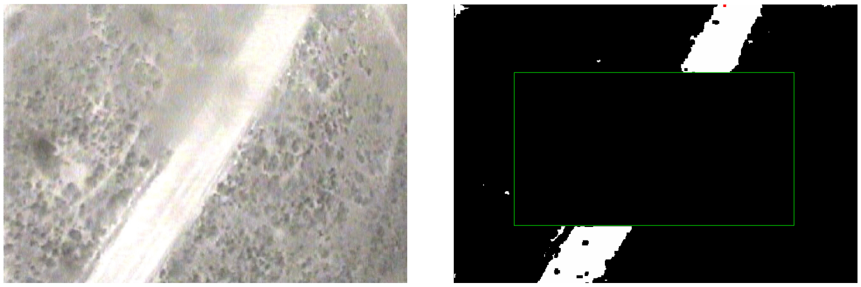
Figure 10: The image steady-state error for each road scenario. The large spikes correspond to sudden changes in the road, such as a turn or a dead-end.

6.2 Flight Test Results

To verify the practicality of using the road-following guidance law, flight-tests were performed. The UAV platform used was the fixed-wing design described in Section 5.1, equipped with a pan-and-tilt gimbaled camera oriented as described in Section 3.3. Figure 11 shows the results of the image-processing algorithm, where (a) is the original video frame and (b) is the classified image. Each edge section has been processed in order to illustrate the sections sizes. Note how the middle section has not been classified. The position of the road is indicated by the red square at the top of the image.

In the experiment, we initialized the algorithm when the UAV was not directly over the road in order to demonstrate how the control law will cause the UAV to change its heading. Figure 12 shows an aerial view of the road that was used for the experiment, where the actual GPS location of the UAV is overlaid on top. The UAV started from the red circle and continued in a south-easterly direction over the road until reaching the red triangle. As can be seen, the control did very well.

The image steady-state error is displayed in Figure 13. Note that the error converges to zero only for a short time. This deviation can be explained by two disturbances: the lag in the movement of the gimbal and high wind conditions. The lag allows the camera to leave the downward position, causing the road in the image to leave the



(a) Original Video Frame

(b) Classified Image

Figure 11: A sample from the results of the image-processing algorithm used during the experimental test flights. In (a) is the original image and in (b) is the classified image. The road position determined by the algorithm for this image is indicated by the red square at the top of the image. The green lines indicate the boundaries of the sections that were process.

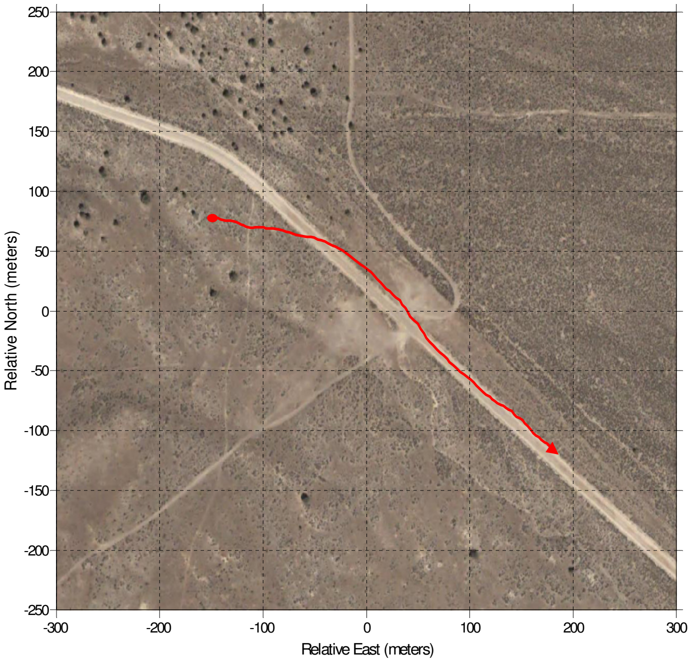


Figure 12: An aerial view of the road used in the flight test with the actual GPS path of the UAV depicted in red, starting at the circle and ending at the triangle. The image was obtained from Google Earth.

desired location. During experiments the wind speeds were approximately 50-60$\%$ of the vehicle's airspeed, requiring the UAV to compensate by significant rolling motions to counteract the wind. Despite these disturbances, the UAV was still able to track the road.

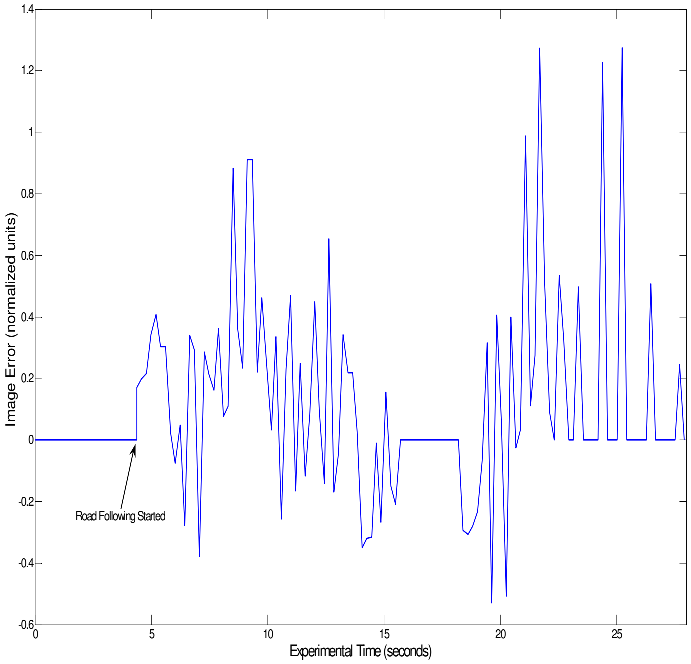


Figure 13: The image steady-state error recorded during the road-following flight-test.

7 Conclusions

In this paper, we have applied proportional navigation to road-following for a UAV using only visual information. We derived the necessary equations and developed a control law for both a skid-to-turn model using a strap-down camera and a bank-to-turn model using a gimbaled camera. After performing some analysis, a modification was made to the algorithm to account for road accelerations caused by bends and turns in the road. We showed simulation results that support the feasibility and the effectiveness of the algorithm, as well as the limitations of the image-processing algorithm used. Finally, flight-tests were performed and validate the suitability of the algorithm.

Acknowledgment

This research was supported by NASA under STTR contract No. NNA04AA19C to Scientific Systems Company, Inc. and Brigham Young University and by NSF award no. CCF-0428004. Portions of this work were performed while the second author was

a National Research Council Fellow at the Air Force Research Laboratory, Munitions Directorate, Eglin, AFB.

References

[1] D. W. Casbeer, D. B. Kingston, R. W. Beard, T. W. McLain, S.-M. Li, and R. Mehra, “Cooperative forest fire surveillance using a team of small unmanned air vehicles,” *International Journal of System Sciences*, vol. 37, no. 6, pp. 351–360, May 2006. [Online]. Available: https://dspace.byu.edu/handle/1877/55

[2] E. Frew, T. McGee, Z. Kim, X. Xiao, S. Jackson, M. Morimoto, S. Rathinam, J. Padial, and R. Sengupta, “Vision-based road-following using a small autonomous aircraft,” in *Proceedings of the IEEE Aerospace Conference*, vol. 5, March 2004, pp. 3006–3015.

[3] S. Rathinam, Z. Kim, A. Soghikian, and R. Sengupta, “Vision based following of locally linear structure using an unmanned aerial vehicle,” in *Proceedings of the 44th IEEE Conference on Decision and Control and the European Control Conference*, Seville, Spain, December 2005, pp. 6085–6090.

[4] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry, *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer-Verlag, 2003.

[5] P. Zarchan, *Tactical and Strategic Missile Guidance*, 4th ed. Reston, Virginia: American Institute of Aeronautics and Astronautics, Inc., 2002, vol. 199 of Progress in Astronautics and Aeronautics.

[6] C. F. Lin, *Modern Navigation, Guidance, and Control Processing*. Englewood Cliffs, New Jersey: Prentice Hall, 1991.

[7] G. M. Siouris, *Missile Guidance and Control Systems*. New York: Springer-Verlag, 2004.

[8] A. E. Bryson and Y. C. Ho, *Applied Optimal Control*. Waltham, Massachusetts: Blaisdell Publishing Company, 1969.

[9] M. Geulman, “Proportional navigation with a maneuvering target,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 8, pp. 364–371, May 1972.

[10] M. Geulman, M. Idan, and O. M. Golan, “Three-dimensional minimum energy guidance,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 31, pp. 835–841, April 1995.

[11] J. W. Egbert and R. W. Beard, “Low altitude road following constraints using strap-down EO cameras on miniature air vehicles,” in *Proceedings of the American Control Conference*, 2007, (to appear).

[12] G. F. Silveira, J. R. H. Carvalho, M. K. Madrid, P. Rives, and S. S. Bueno, "A fast vision-based road following strategy applied to the control of aerial robots," in *Proceedings of the IEEE XIV Brazilian Symposium on Computer Graphics and Image Processing*, October 2001, pp. 226–231.

[13] A. Broggi, "An image reorganization procedure for automotive road following systems," in *Proceedings of the IEEE International Conference on Image Processing*, vol. 3, October 1995, pp. 532–535.

[14] A. K. Jain, *Fundamentals of Digital Image Processing*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1989, ch. Image Analysis and Computer Vision, pp. 407–414.

[15] R. W. Beard, J. W. Curtis, M. Eilders, J. Evers, and J. R. Cloutier, "Vision aided proportional navigation for micro air vehicles," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*. Hilton Head, North Carolina: American Institute of Aeronautics and Astronautics, August 2007, paper number AIAA-2007-6609.

[16] M. Mehrandezh, N. M. Sela, R. G. Fenton, and B. Benhabib, "Three-dimensional minimum energy guidance," *IEEE Transactions on Systems, Man, and Cybernetics–Part A: Systems and Humans*, vol. 30, no. 3, pp. 238–250, May 2000.

[17] B. L. Stevens and F. L. Lewis, *Aircraft Control and Simulation, 2nd Edition*. John Wiley & Sons, Inc., 2003.